Machine Learning for HEP

Lecture III — Generative models for the LHC

UNIVERSITÀ
DEGLI STUDI
DI MILANO

BND Graduate School — Blankenberge 2024

Ramon Winterhalder

Plan of attack

Wednesday

1. Introduction to Machine Learning
    - Basic concepts of machine learning

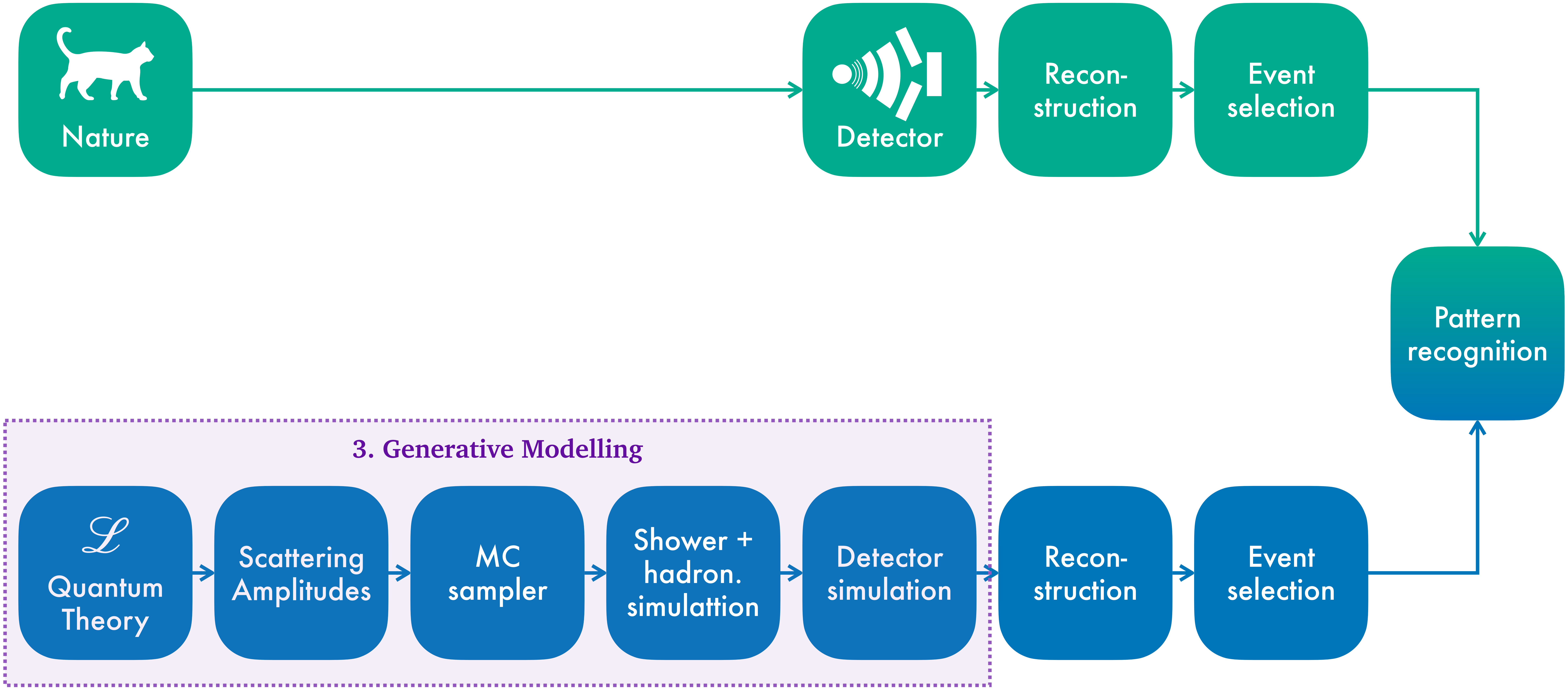
2. Bayesian Neural Networks
    - Uncertainty estimation

Thursday

3. Generative Models for the LHC
    - GANs, normalizing flows & diffusion models

4. Anomaly Detection
    - Autoencoders, CWoLa,…

LHC analysis + ML



Lecture III

Generative Models for the LHC

Current Hype: Text-to-Image Models

Text prompt

Realistic photography of Max Planck sitting at his laptop being super happy while coding

Text analysis
(Typically **Transformer**)

Image generation
(Typically **Diffusion**)

Generated with Midjourney

Generative Models

GAN

GAN Art (2018)
→ sold for $432,500

Diffusion Models

State-of-the-art
image generation

Transformer

ChatGPT

State-of-the-art
text generation

What is a Generative Model?

We have: $\boxed{p_{\text{truth}} \equiv p_{\text{data}}(x)}$ → We want to generate new samples $\boxed{x \sim p_\omega(x) \simeq p_{\text{data}}(x)}$

The distribution p_{truth} is usually given as:

- **explicit** as function (e.g. $\mathrm{d}\sigma \propto$ differential cross-section)
- **implicit** via a set of training data $\boxed{\{x\} \sim p_{\text{data}}(x)}$

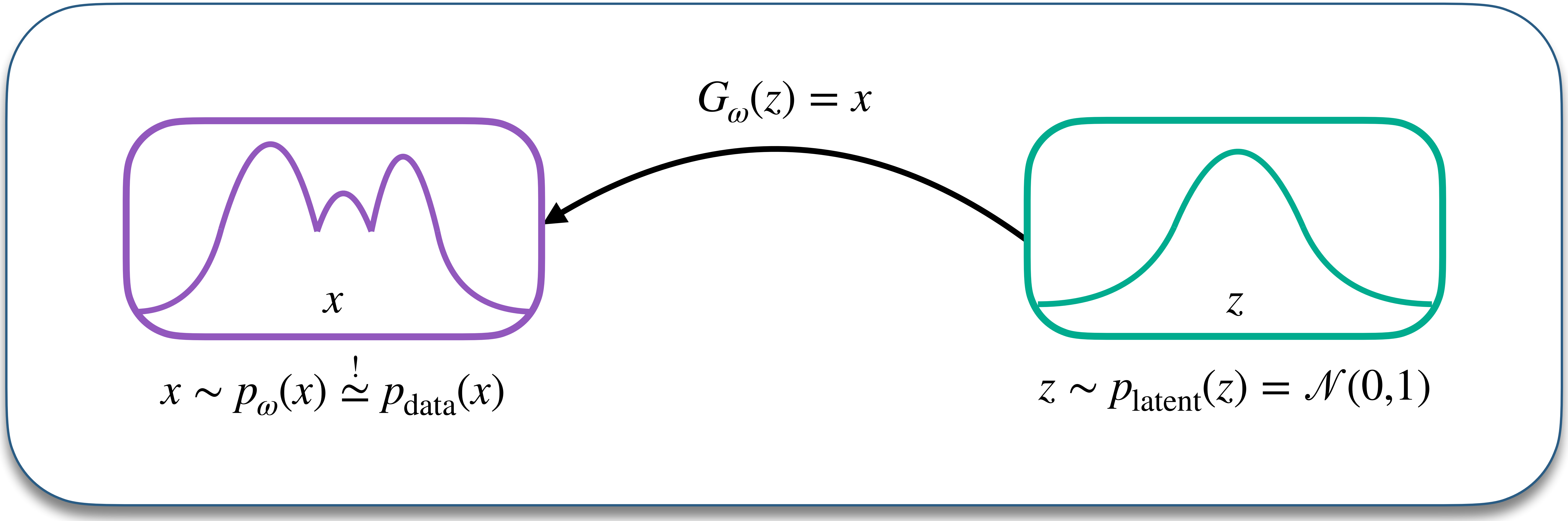
In **particle physics:**

- Event generation
- Calorimeter simulation
- Unfolding
- MEM (transfer function)

→ this is a stochastic (random) process (RNG)

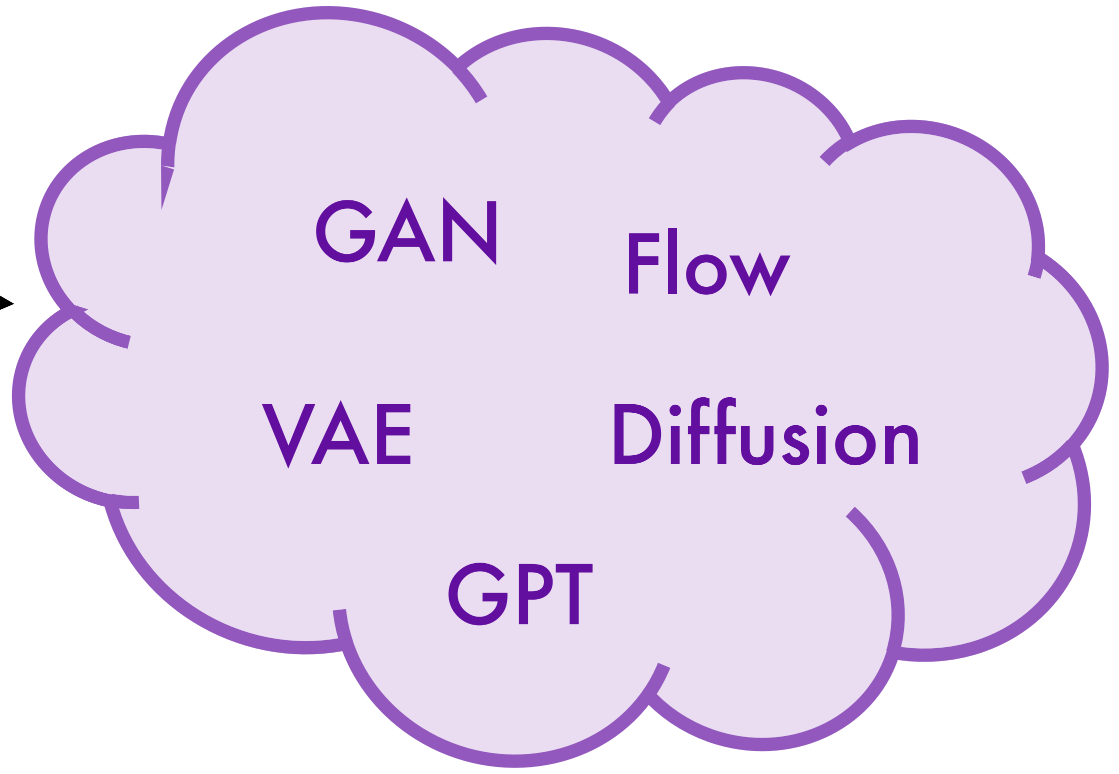
→ needs “random” input

What is a Generative Model?



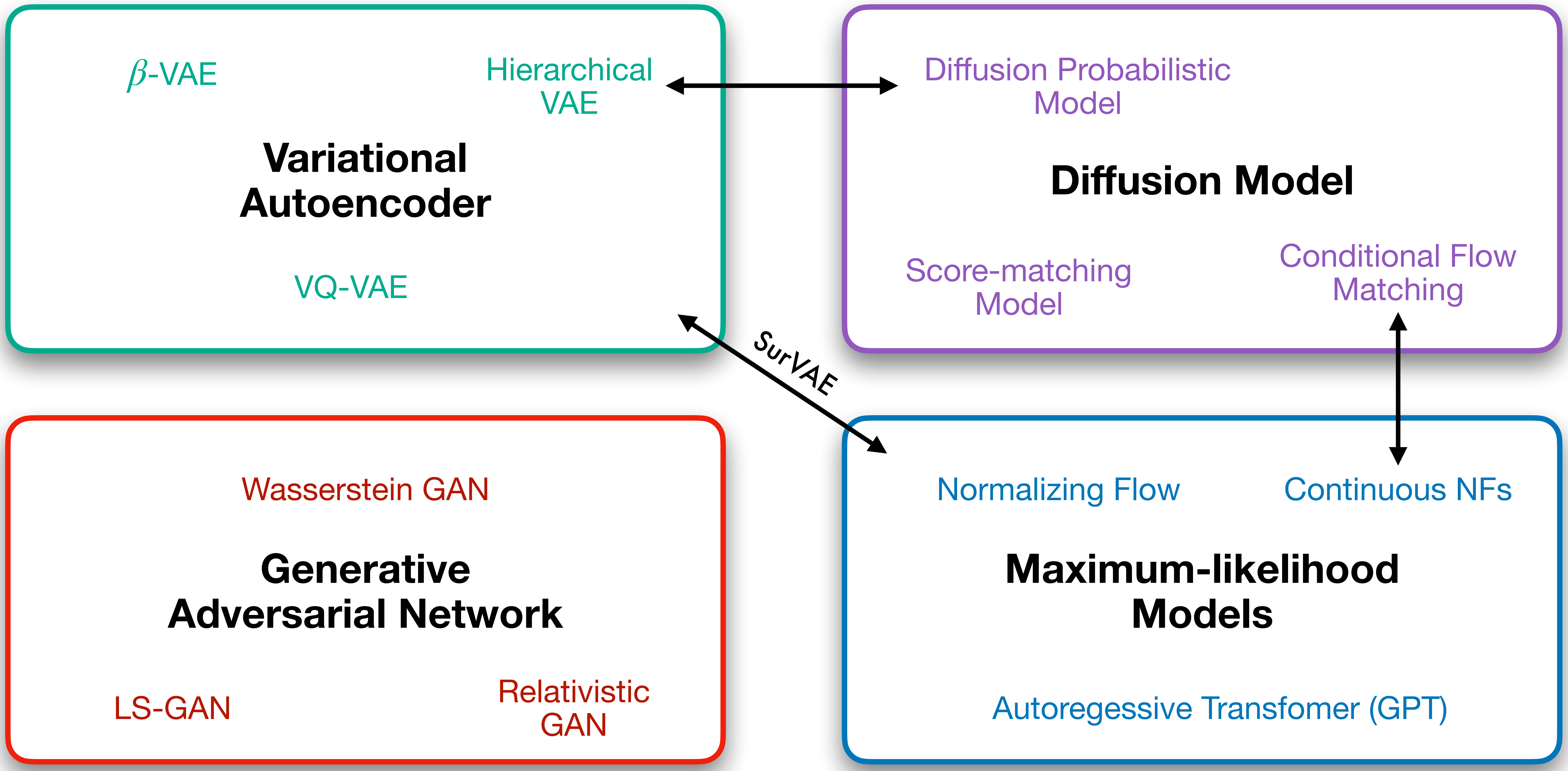
→ How to **construct** and **train** $G_\omega(z)$?

→ **Multiple types** of generative models

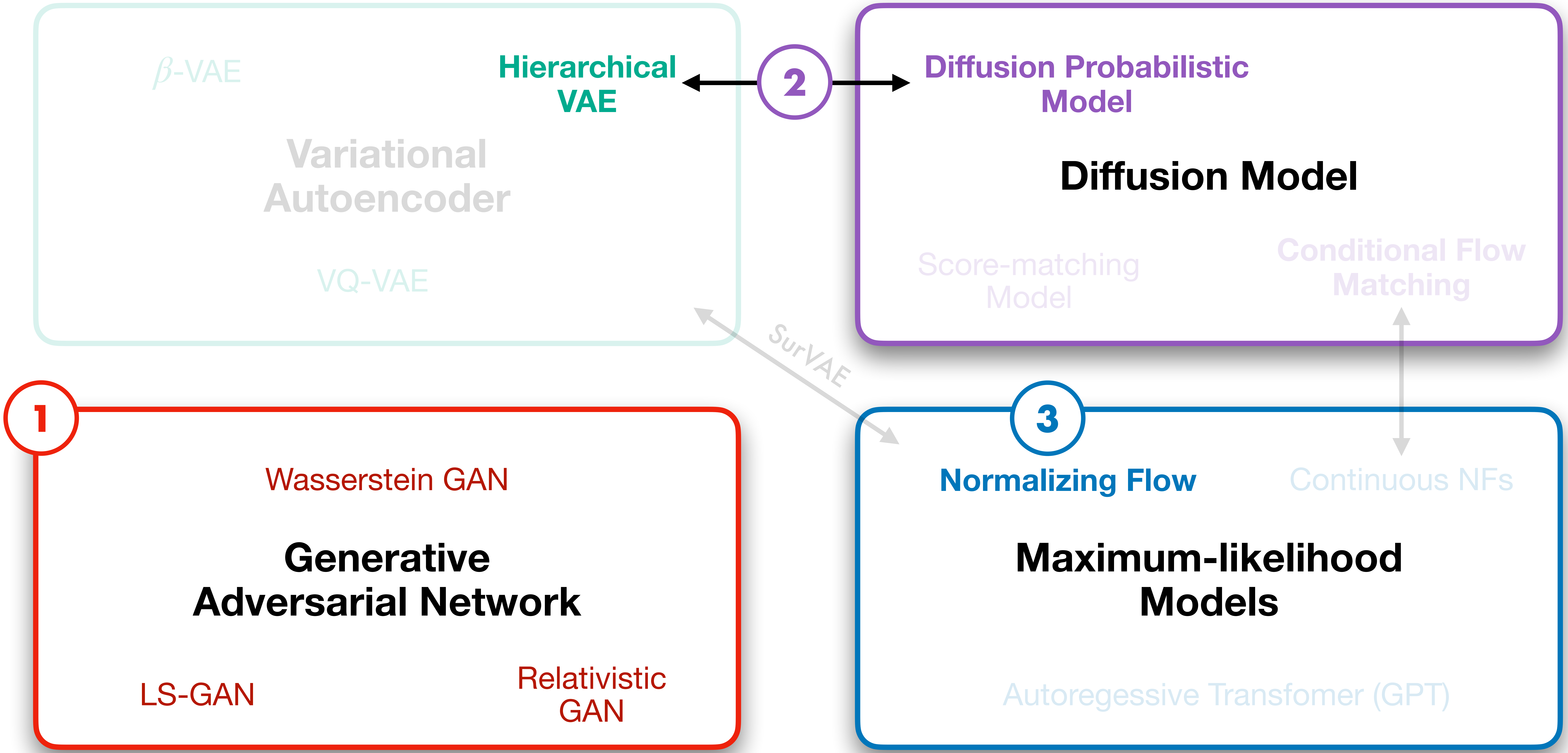


Types of deep generative models

Deep generative models



Deep generative models

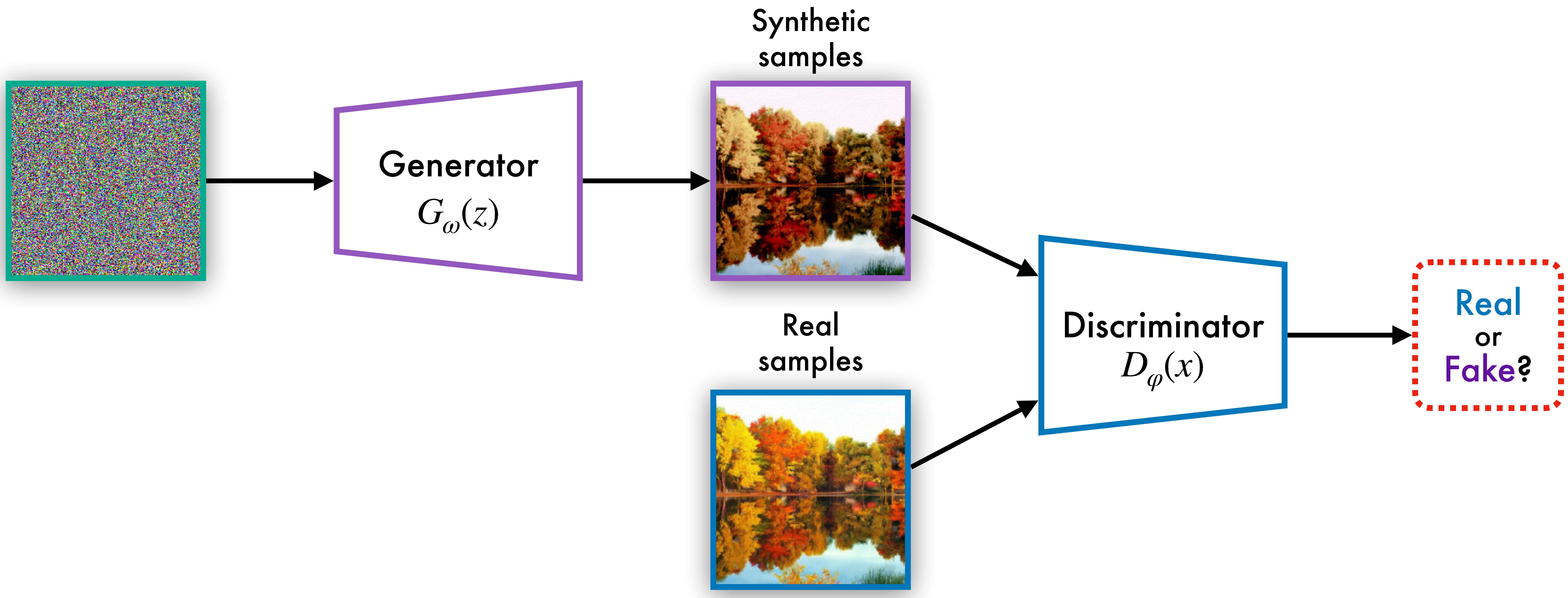


Part I

Generative Adversarial Networks

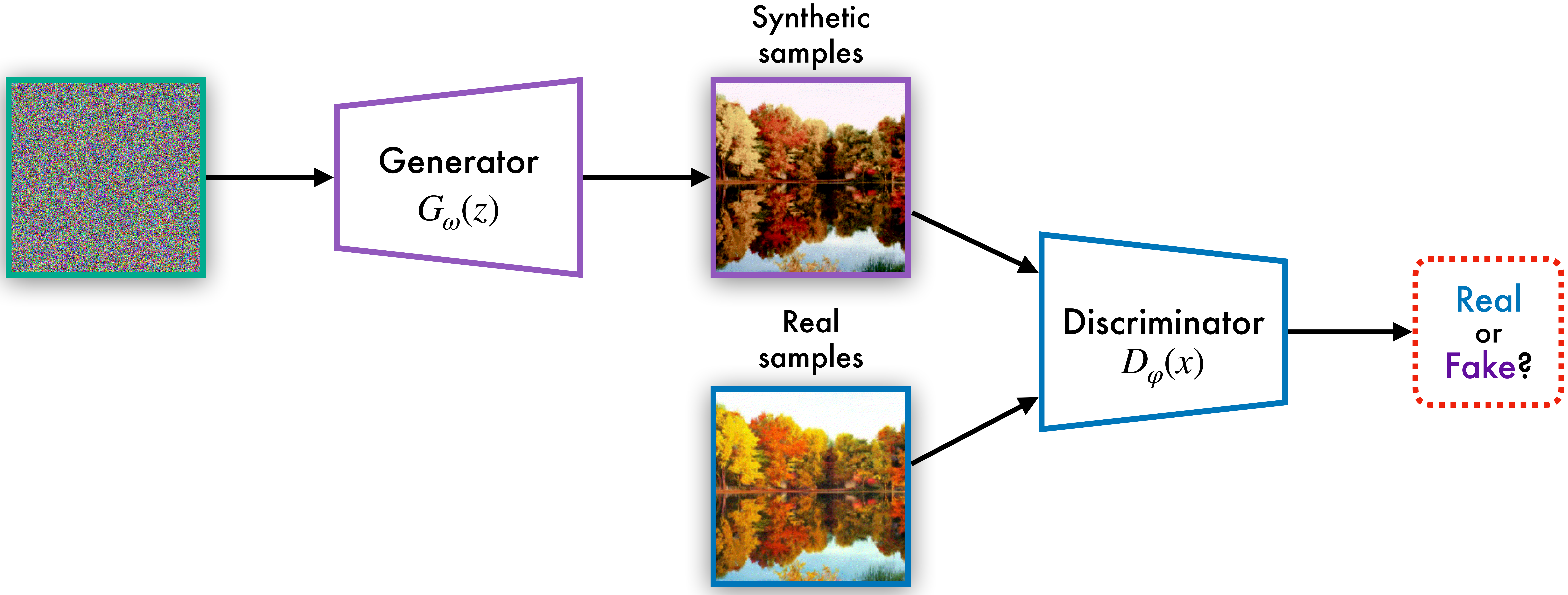
[1406.2661]

Generative adversarial network



Goodfellow [1406.2661]

Generative adversarial network



$$\mathcal{L}_D = -\left\langle \log D_\varphi(x) \right\rangle_{x\sim p_{\text{data}}} - \left\langle \log(1 - D_\varphi(x)) \right\rangle_{x\sim p_{\hat{\omega}}}$$
$$= -\left\langle \log D_\varphi(x) \right\rangle_{x\sim p_{\text{data}}} - \left\langle \log(1 - D_\varphi(G_{\hat{\omega}}(z))) \right\rangle_{z\sim p_{\text{latent}}}$$

Iterative Training

$$\mathcal{L}_G = \left\langle \log(1 - D_{\hat{\varphi}}(x)) \right\rangle_{x\sim p_\omega}$$
$$= \left\langle \log(1 - D_{\hat{\varphi}}(G_\omega(z))) \right\rangle_{z\sim p_{\text{latent}}}$$

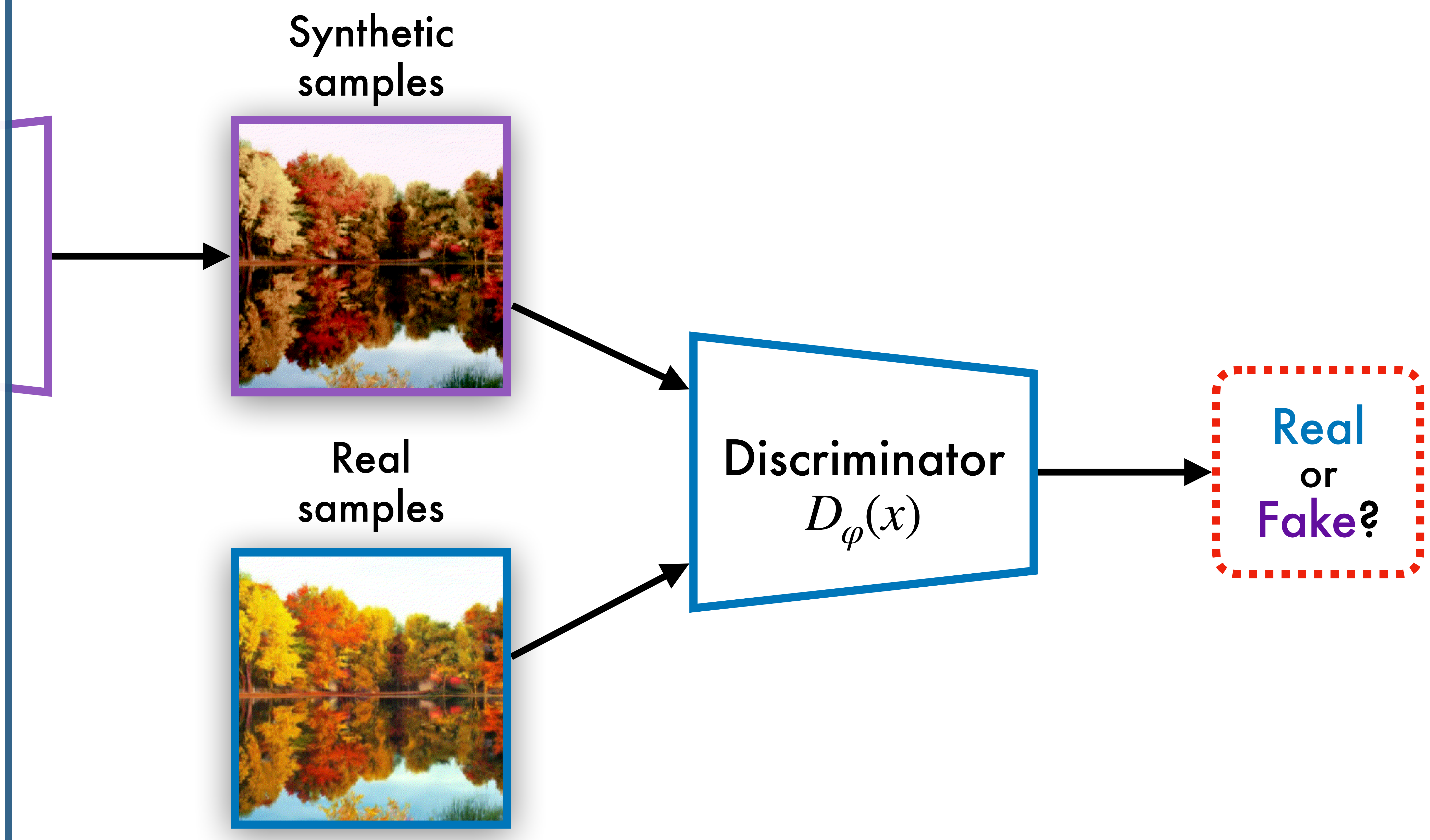
Generative adversarial network

Problems with GANs

- ⊖ Min-max training unstable
  → vanishing gradients
- ⊖ Metric for sucess?
  → loss only shows competition!
- ⊖ Mode collapse

Ways to improve

- ⊕ Wasserstein GAN [1701.07875, 1704.00028]
- ⊕⊕ Gradient penalty [1705.09367, 1801.04406]
- ⊕⊕ Spectral normalization [1802.05957]
- ⊕ Other losses
  - Least square [1611.04076]
  - Relativistic GAN [1807.00734]
  - …



Iterative Training

$$\mathcal{L}_G = \left\langle \log(1 - D_{\hat\varphi}(x)) \right\rangle_{x \sim p_\omega} = \left\langle \log(1 - D_{\hat\varphi}(G_\omega(z))) \right\rangle_{z \sim p_{\text{latent}}}$$

Part II

Diffusion Models

[2006.11239, 2210.02747]

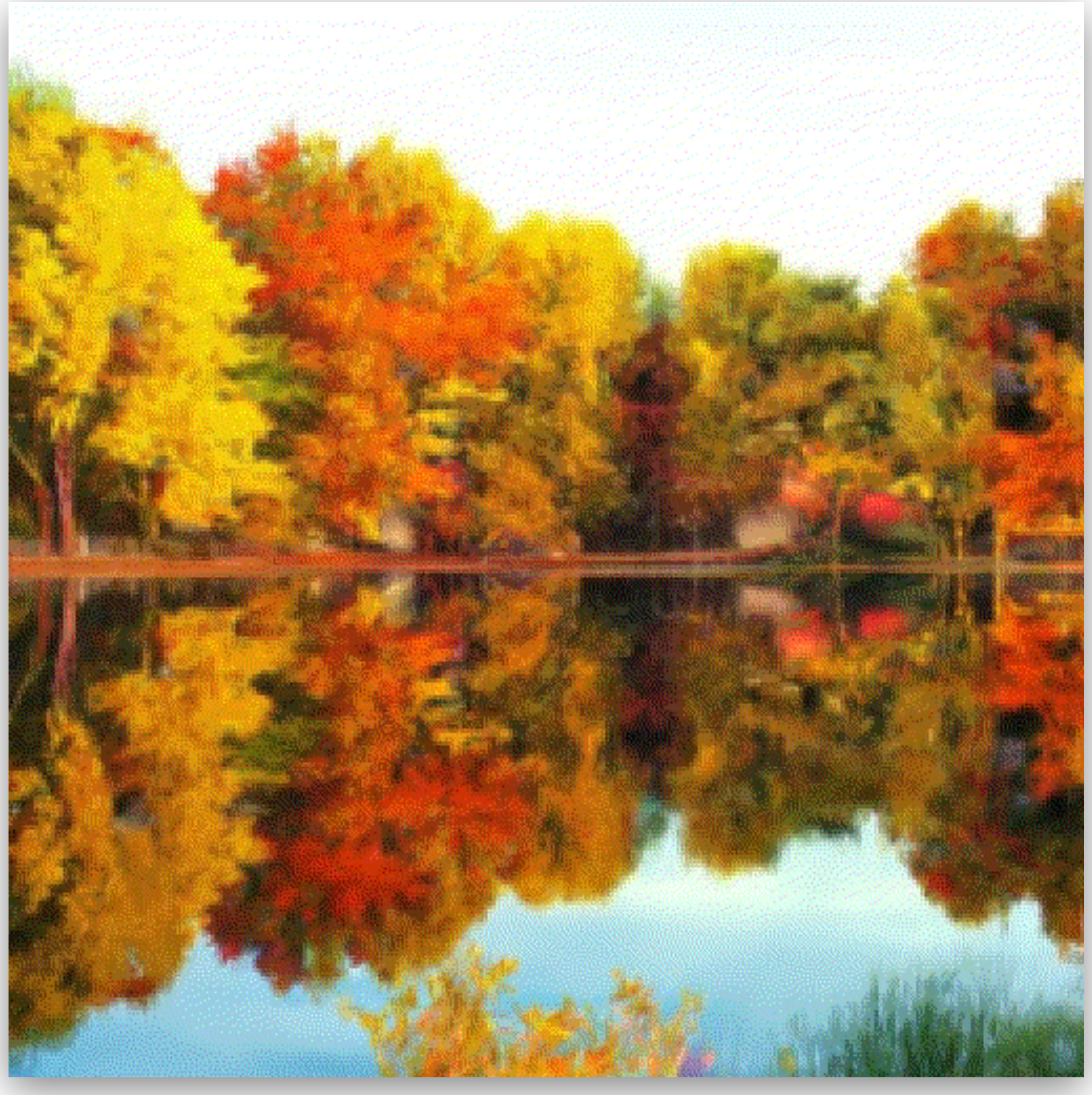
Midjourney

Dall-E

Recently…

Stable Diffusion

Diffusion Models

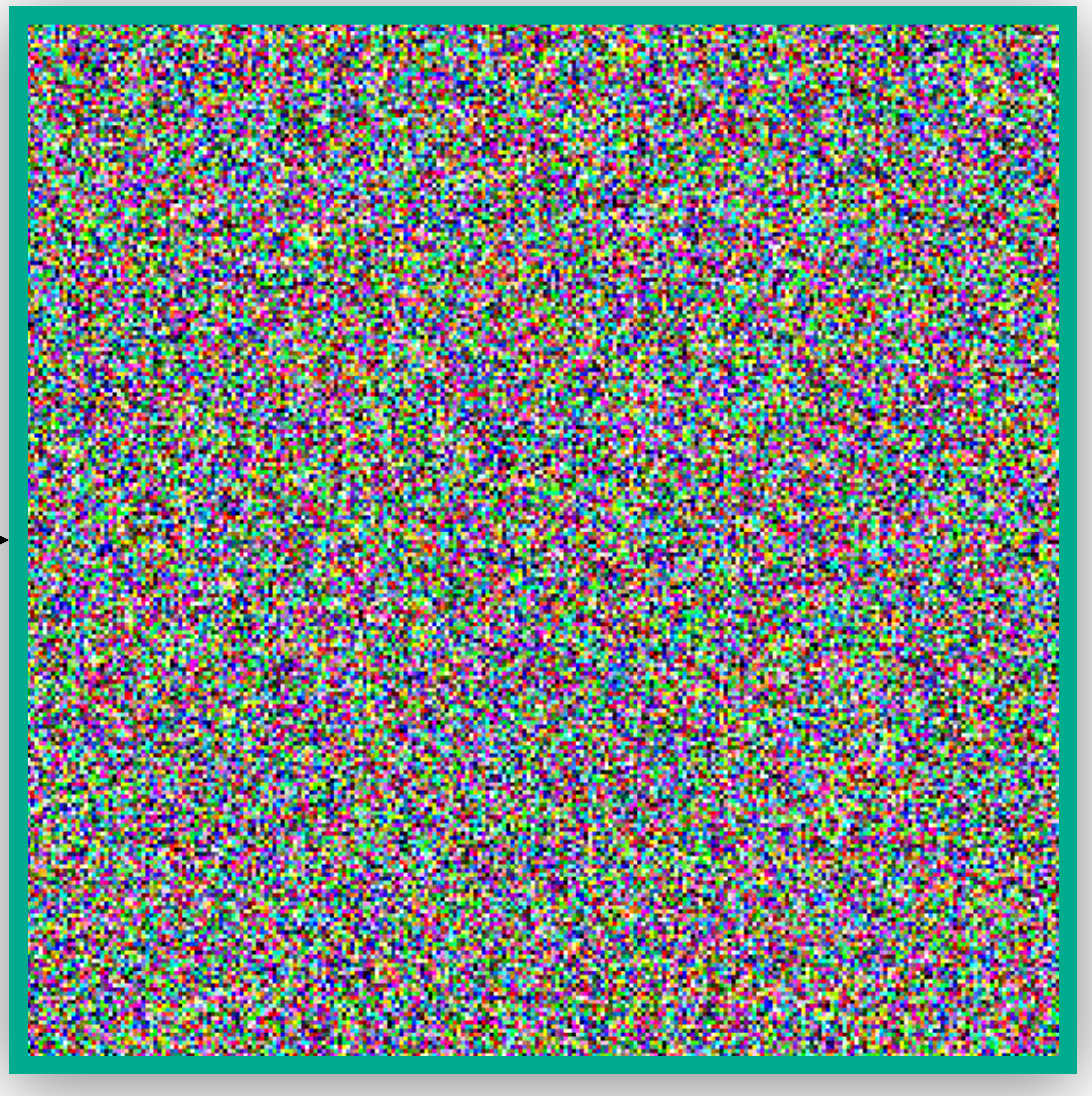


[2006.11239, 2210.02747]

Diffusion Models

Define mapping as **time-dependent** diffusion process

t



$x \sim p_\omega(x) \overset{!}{\simeq} p_{\mathrm{data}}(x)$

Parametrization of time

- discrete: $t = 0,1,\ldots,T$
- continuous: $t \in [0,1]$

$z \sim p_{\mathrm{latent}}(z) = \mathcal{N}(0,1)$

→ Gradually **add noise** to data samples to transform them to gaussians

← Gradually **remove noise** from gaussians to obtain data samples

[2006.11239, 2210.02747]

VAEs Revisited

Bayes' Theorem

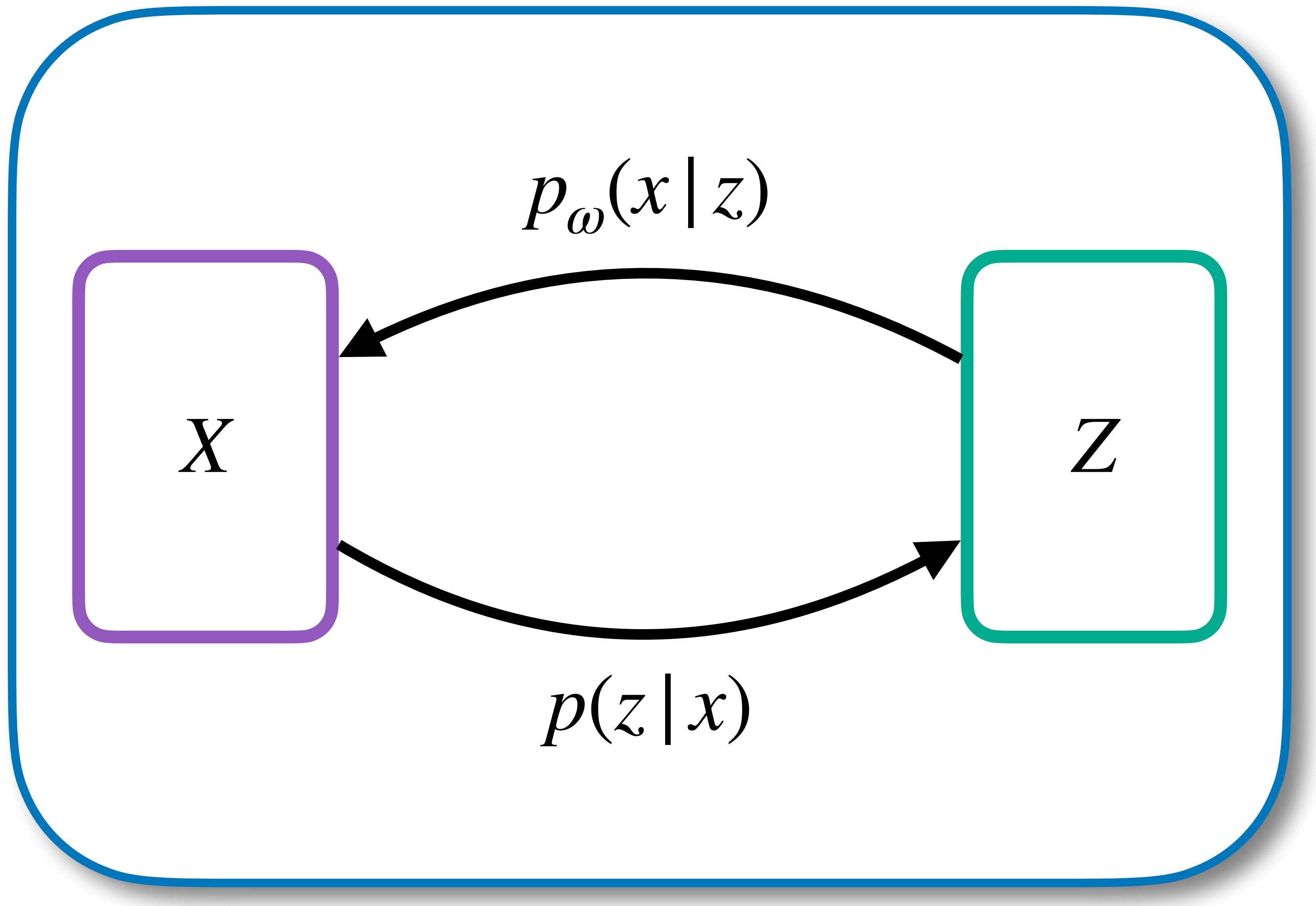
$$\boxed{p(z\,|\,x) = \frac{p(x\,|\,z)\,p(z)}{p(x)}}$$

$p(z)$ ← Prior

$p(x)$ ← Evidence

Problem: evidence intractable

$$\boxed{p(x) = \int \mathrm{d}z\, p(x\,|\,z)\,p(z)}$$



Variational inference

Encoder: $\boxed{q_\phi(z\,|\,x) = \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x))}$

Likelihood: $\boxed{p_\omega(x\,|\,z)}$ ← Decoder

Posterior: $\boxed{p(z\,|\,x)}$ ← Want!

Match!

[1312.6114, 1906.02691]

Evidence Lower Bound — ELBO

Minimize KL

$$\mathrm{KL}(q_\phi(z\,|\,x), p(z\,|\,x)) = \int \mathrm{d}z\, q_\phi(z\,|\,x)\, \log\left(\frac{q_\phi(z\,|\,x)}{p(z\,|\,x)}\right)$$

Maximize ELBO

ELBO $\neq$ Elmo

Evidence Lower Bound — ELBO

Minimize KL

$$\mathrm{KL}(q_\phi(z\,|\,x), p(z\,|\,x)) = \int \mathrm{d}z\, q_\phi(z\,|\,x)\, \log\left(\frac{q_\phi(z\,|\,x)}{p(z\,|\,x)}\right)$$

Maximize ELBO

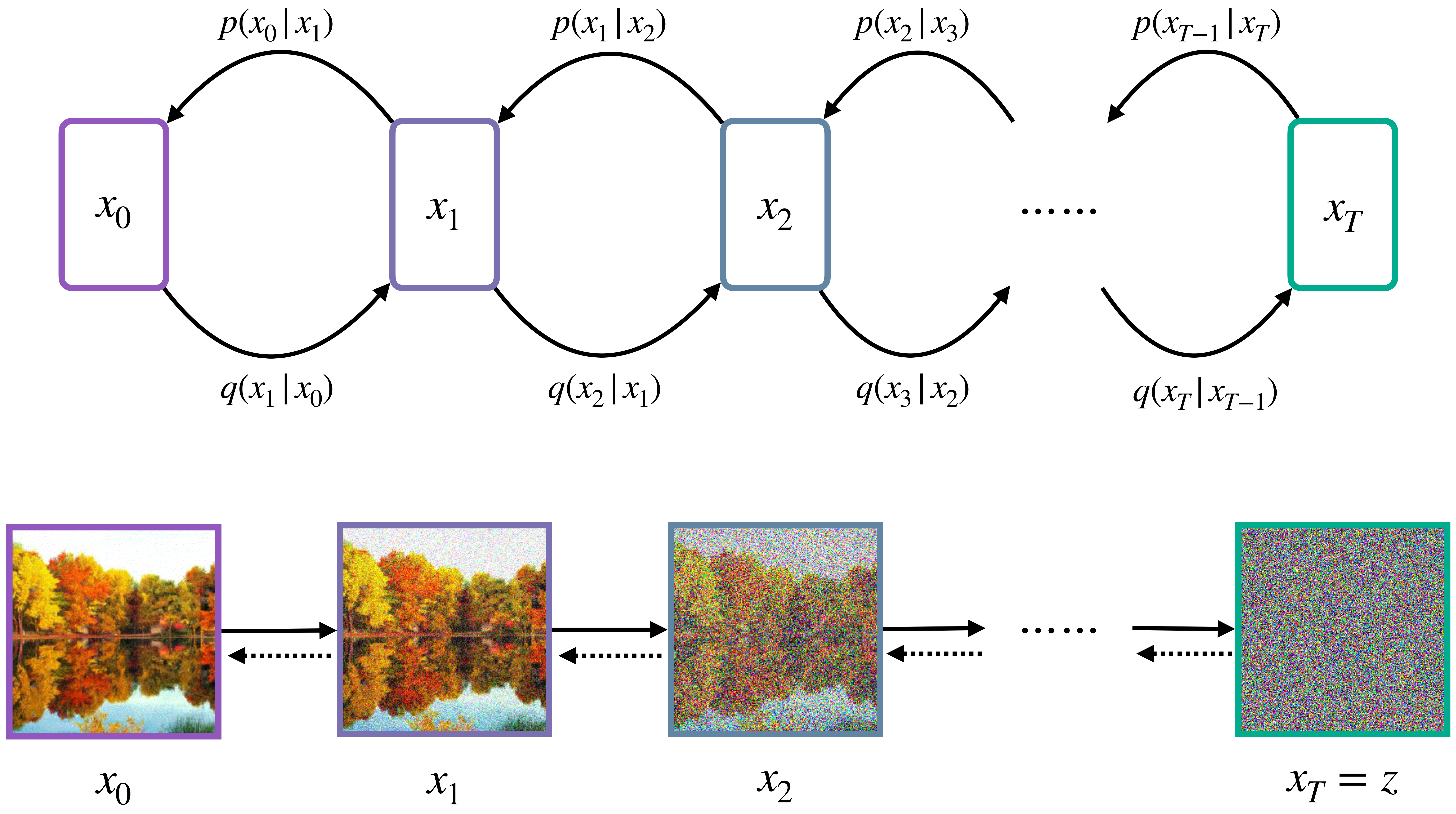
$$\begin{aligned}
\log p(x) &= \int \mathrm{d}z\, q_\varphi(z\,|\,x)\, \log p(x) \\
&= \int \mathrm{d}z\, q_\varphi(z\,|\,x)\, \log \frac{p_\omega(x\,|\,z)\, p(z)}{p(z\,|\,x)} \\
&= \int \mathrm{d}z\, q_\varphi(z\,|\,x) \left[\log p_\omega(x\,|\,z) - \log \frac{q_\phi(z\,|\,x)}{p(z)} + \log \frac{q_\phi(z\,|\,x)}{p(z\,|\,x)}\right] \\
&= \big\langle \log p_\omega(x\,|\,z) \big\rangle_{q_\phi(z|x)} - \mathrm{KL}(q_\phi(z\,|\,x), p(z)) + \underbrace{\mathrm{KL}(q_\phi(z\,|\,x), p(z\,|\,x))}_{\geq 0} \\
&\geq \underbrace{\big\langle \log p_\omega(x\,|\,z) \big\rangle_{q_\phi(z|x)} - \mathrm{KL}(q_\phi(z\,|\,x), p(z))}_{\text{ELBO}}
\end{aligned}$$

Bayes' theorem

SurVAE [2007.02731]

$$\log p(x) = \int \mathrm{d}z\, q_\varphi(z\,|\,x) \left[\log p(z) + \log \frac{p_\omega(x\,|\,z)}{q_\phi(z\,|\,x)} + \log \frac{q_\phi(z\,|\,x)}{p(z\,|\,x)}\right]$$

Denoising Diffusion Model (DDPM)

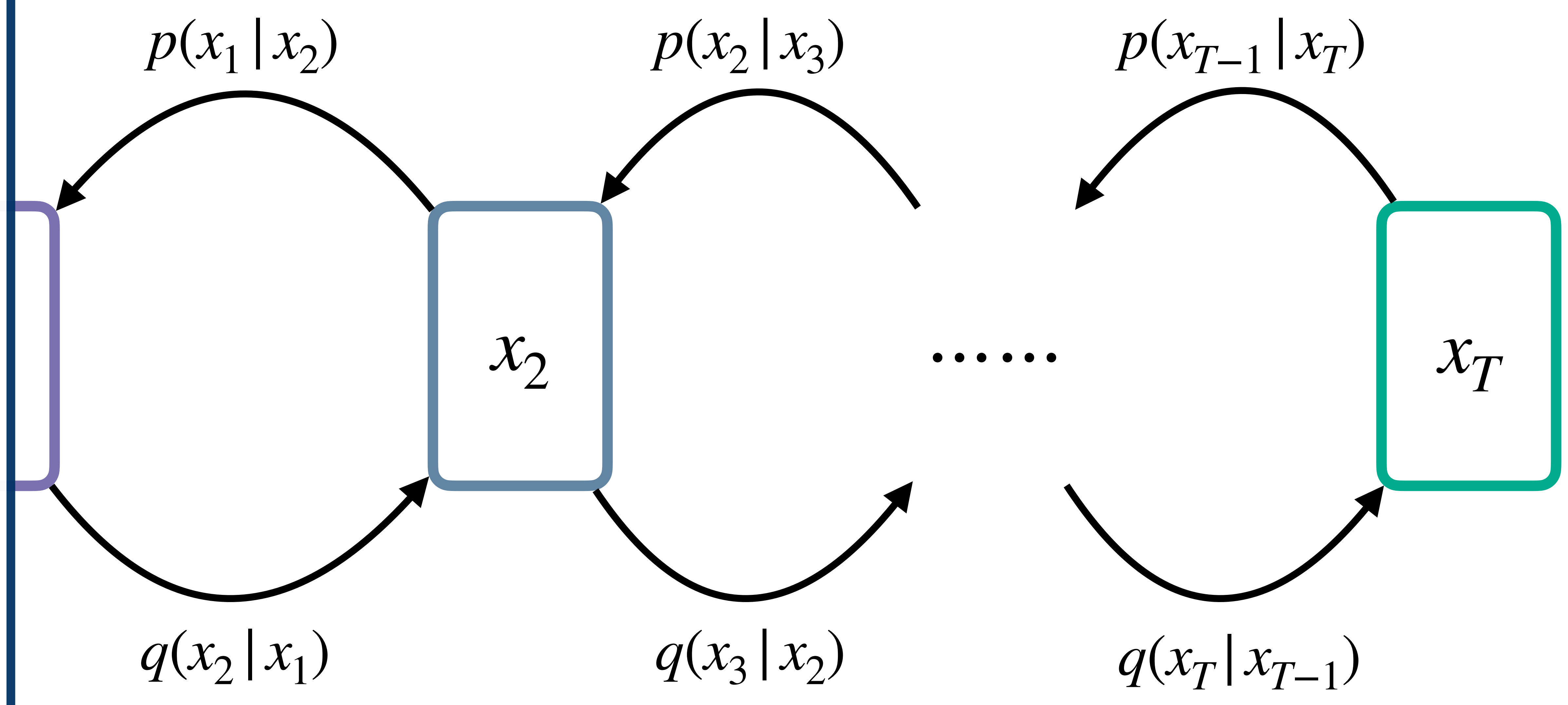


[2006.11239]

Denoising Diffusion Model (DDPM)

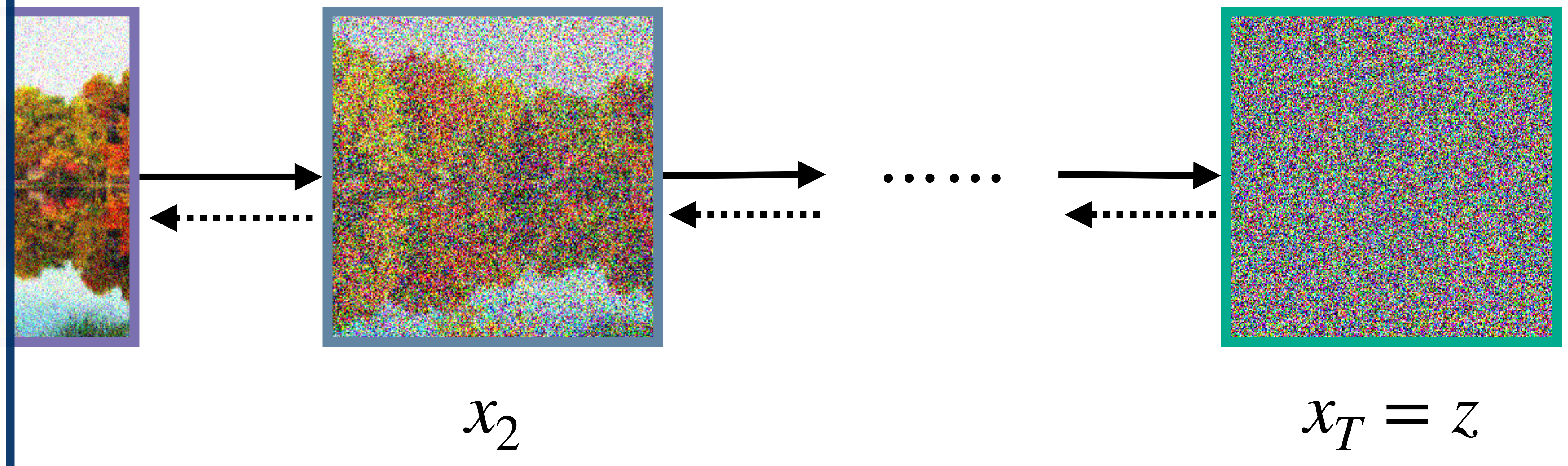
Hierachical VAEs = DDPMs

- explicit **diffusion process**: $q(x_t | x_{t-1})$
- no **bottleneck**: $\dim x_0 = \dim x_i$
- find **reverse process**: $p_\omega(x_t | x_{t+1})$



Key facts - Diffusion Model

- ⊕ State-of-the-art in precision
- ⊕ Fast and stable training
- ⊖ Slow evaluation

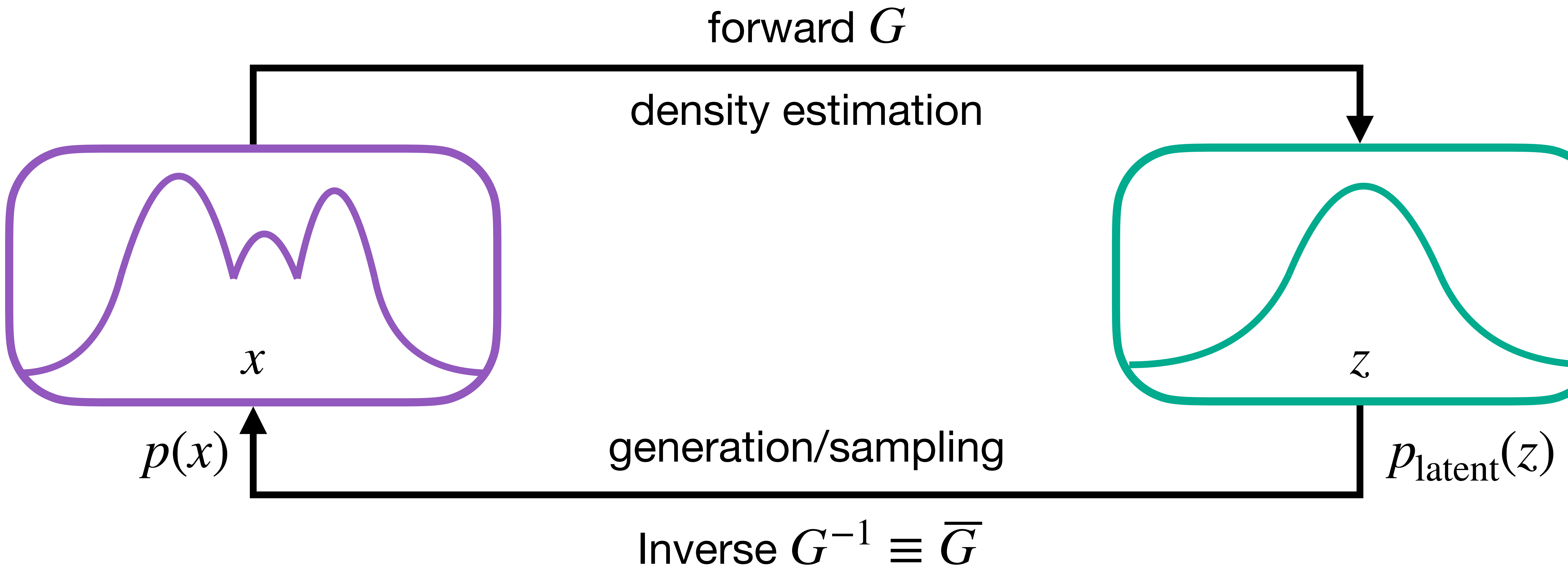


Part III

Normalizing Flows

[1505.05770, 1908.09257]

Normalizing flow — Basics

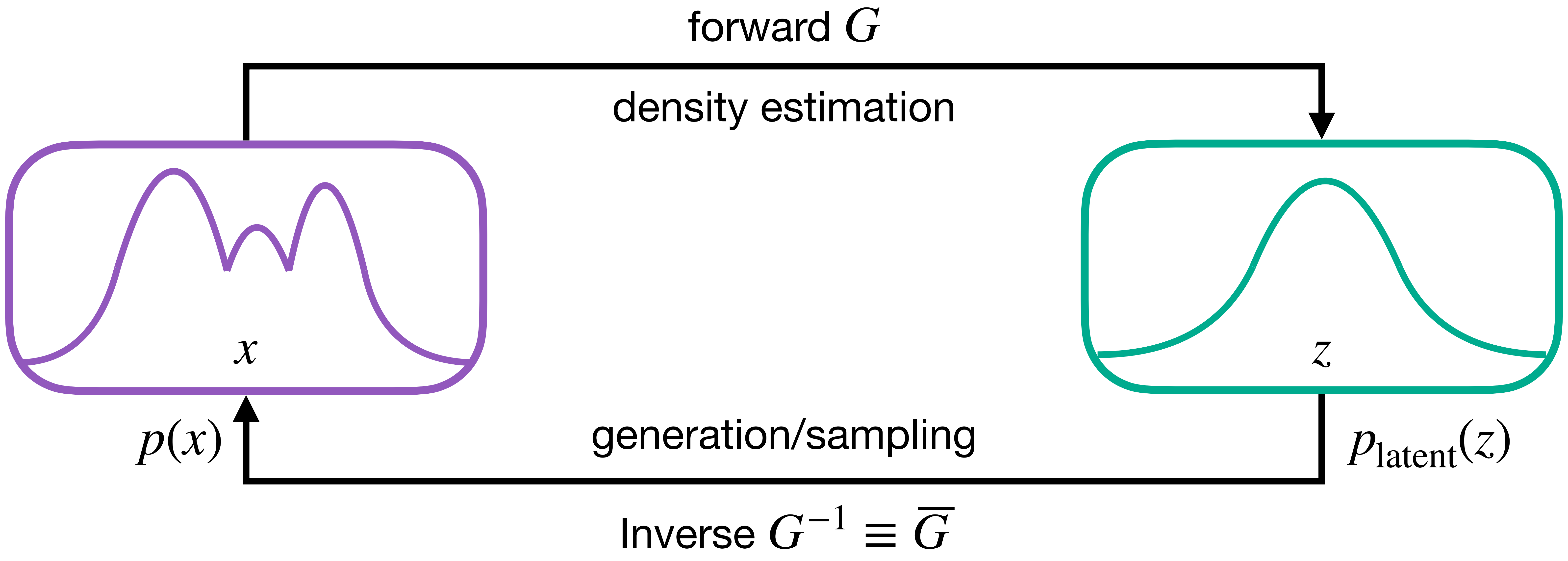


Conservation of probability: $\boxed{p(x)\,\mathrm{d}x = p_{\text{latent}}(z)\,\mathrm{d}z}$ with $\boxed{z = G_\omega(x) \quad x = \overline{G}_\omega(z)}$

Change-of-variables formula: $\boxed{p_\omega(x) = p_{\text{latent}}(z = G_\omega(x)) \cdot \left| \frac{\partial G_\omega(x)}{\partial x} \right|}$

[1505.05770, 1908.09257]

Normalizing flow — Basics



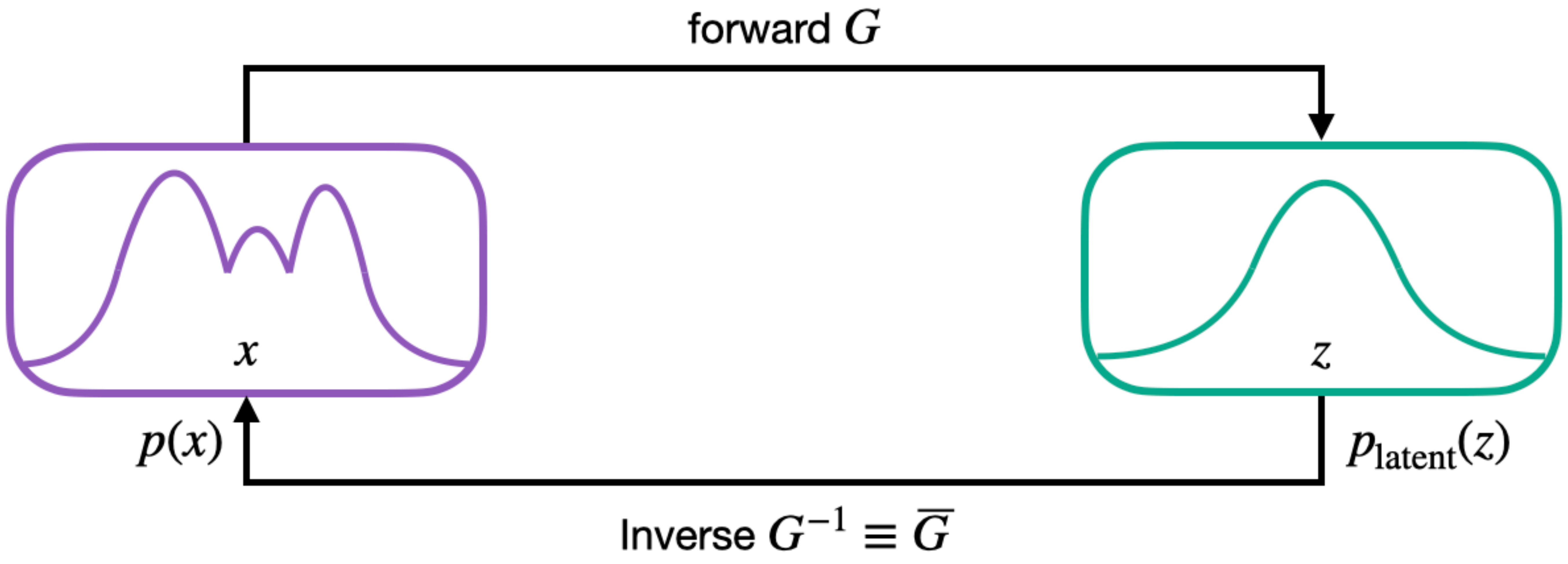
Conservation of probability: $\boxed{p(x)\,\mathrm{d}x = p_{\text{latent}}(z)\,\mathrm{d}z}$ with $\boxed{z = G_\omega(x) \quad x = \overline{G}_\omega(z)}$

Change-of-variables formula: $\boxed{\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|}$

[1505.05770, 1908.09257]

How to train it?

Normalizing flow — Training



$$\log p_\omega(x) = \log p_{\mathrm{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Match $p_\omega(x)$ with $p_{\mathrm{data}}(x)$

Kullback-Leibler divergence:

$$\begin{aligned} \mathrm{KL}(p_{\mathrm{data}}(x)\,|\,p_\omega(x)) &= \int \mathrm{d}x\, p_{\mathrm{data}}(x)\, \log \frac{p_{\mathrm{data}}(x)}{p_\omega(x)} \\ &= -\int \mathrm{d}x\, p_{\mathrm{data}}(x)\, \log p_\omega(x) + \boxed{\int \mathrm{d}x\, p_{\mathrm{data}}(x)\, \log p_{\mathrm{data}}(x)} \end{aligned}$$

No ω dependence

Negative log-likelihood loss:

$$\mathcal{L}_{\mathrm{NLL}} = -\int \mathrm{d}x\, p_{\mathrm{data}}(x)\, \log p_\omega(x) = \big\langle -\log p_\omega(x) \big\rangle_{x \sim p_{\mathrm{data}}}$$

Tractable Jacobian?

$$\log p_\omega(x) = \log p_{\text{latent}}(z = G_\omega(x)) + \log \left| \frac{\partial G_\omega(x)}{\partial x} \right|$$

→ Requires tractable Jacobian!

In general: $g_\omega(x) = \left| \frac{\partial G_\omega(x)}{\partial x} \right|$ is $d \times d$ matrix → Scales with $\mathcal{O}(d^3)$ ☹

Solution: **Autoregressive transformations** $z = \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix}$ $x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$

$$\begin{aligned} z_1 &\equiv z_1(x_1) \\ z_2 &\equiv z_2(x_1, x_2) \\ &\vdots \\ z_d &\equiv z_d(x_1, x_2, \ldots, x_d) \end{aligned}$$

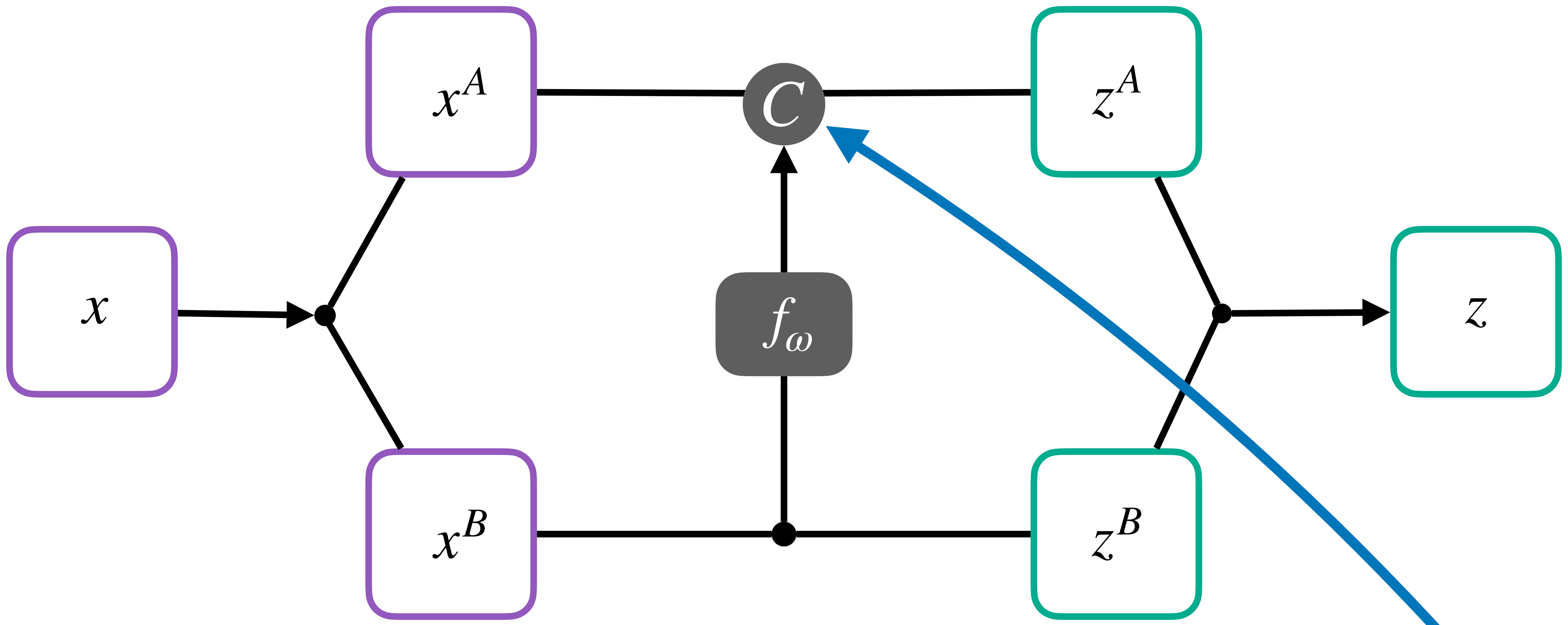
→

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_d}{\partial x_1} \\ 0 & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial z_d}{\partial x_1} \\ \vdots & \ddots & & \vdots \\ 0 & \ldots & 0 & \frac{\partial z_d}{\partial x_d} \end{pmatrix}$$

→

$$\det J = \prod_i J_{ii} \sim \mathcal{O}(d)$$ ☺

Coupling block



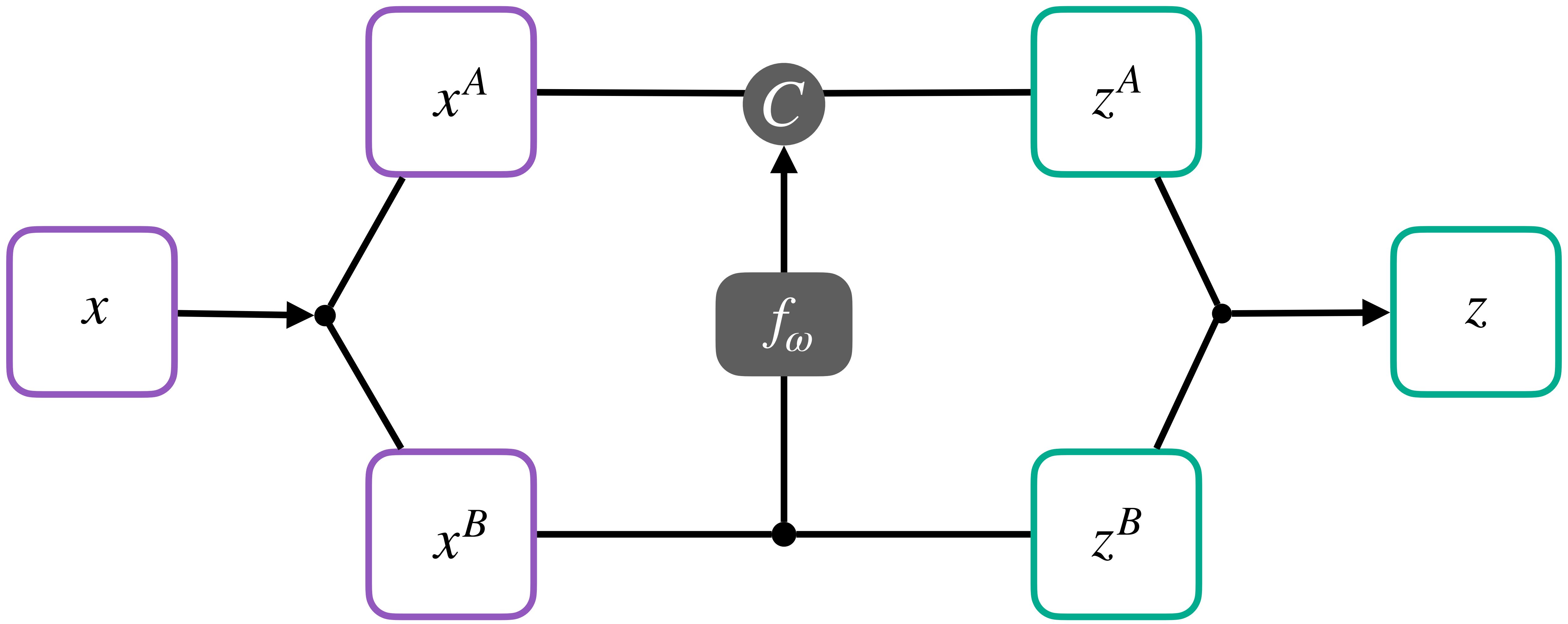
Forward pass:
$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

Inverse pass:
$$x^A = C^{-1}(z^A; f_\omega(z^B))$$
$$x^B = z^B$$

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

What is the function C?

Coupling block



Forward pass: $$\begin{aligned} z^A &= C(x^A; f_\omega(x^B)) \\ z^B &= x^B \end{aligned}$$

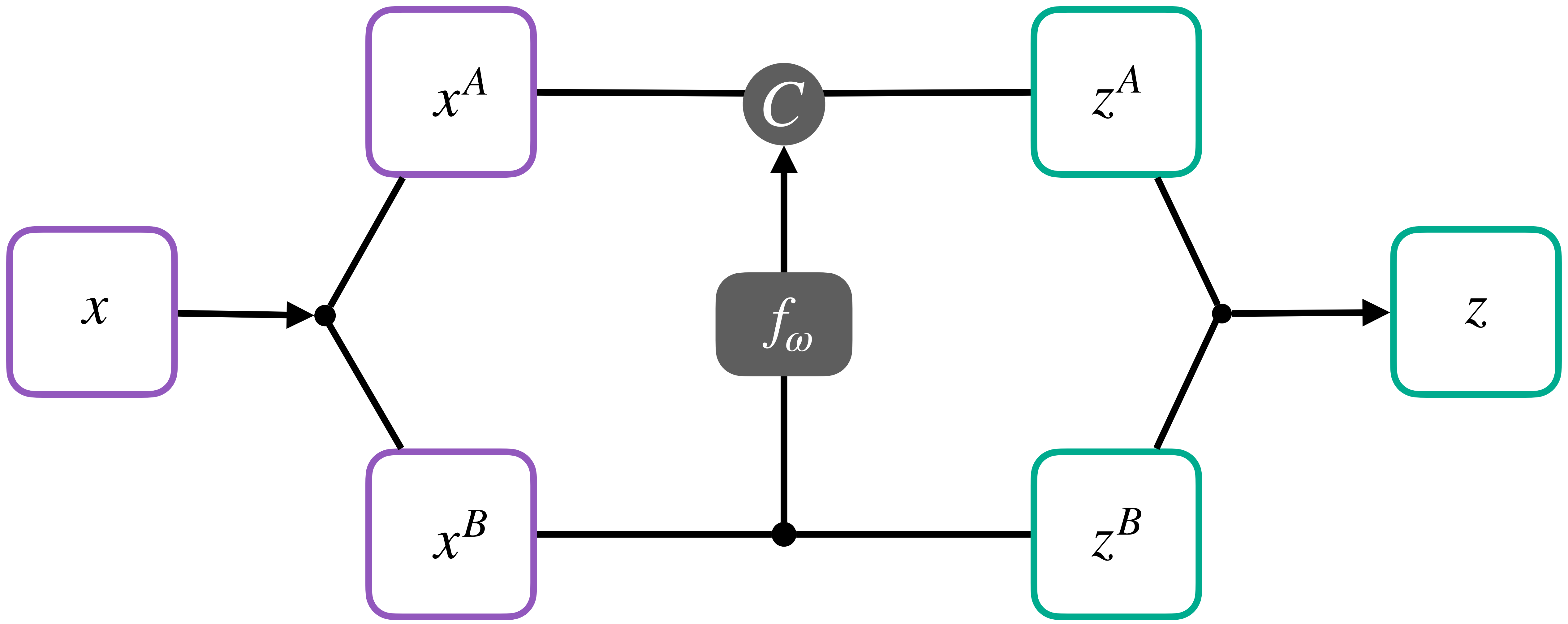
Inverse pass: $$\begin{aligned} x^A &= C^{-1}(z^A; f_\omega(z^B)) \\ x^B &= z^B \end{aligned}$$

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

Affine [1605.08803] $$C^A = \alpha_\omega(x^B) \cdot x^A + \mu_\omega(x^B)$$

parametrized by NN

Coupling block



Forward pass:
$$z^A = C(x^A; f_\omega(x^B))$$
$$z^B = x^B$$

Inverse pass:
$$x^A = C^{-1}(z^A; f_\omega(z^B))$$
$$x^B = z^B$$

$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

Affine [1605.08803]
$$C^A = \alpha_\omega(x^B) \cdot x^A + \mu_\omega(x^B)$$

Quadratic [1808.03856]
$$C = a_\omega x^2 + b_\omega x + c_\omega$$

Rational quadratic [1906.04032]
$$C = \frac{a_\omega x^2 + b_\omega x + c_\omega}{d_\omega x^2 + e_\omega x + f_\omega}$$

Coupling block

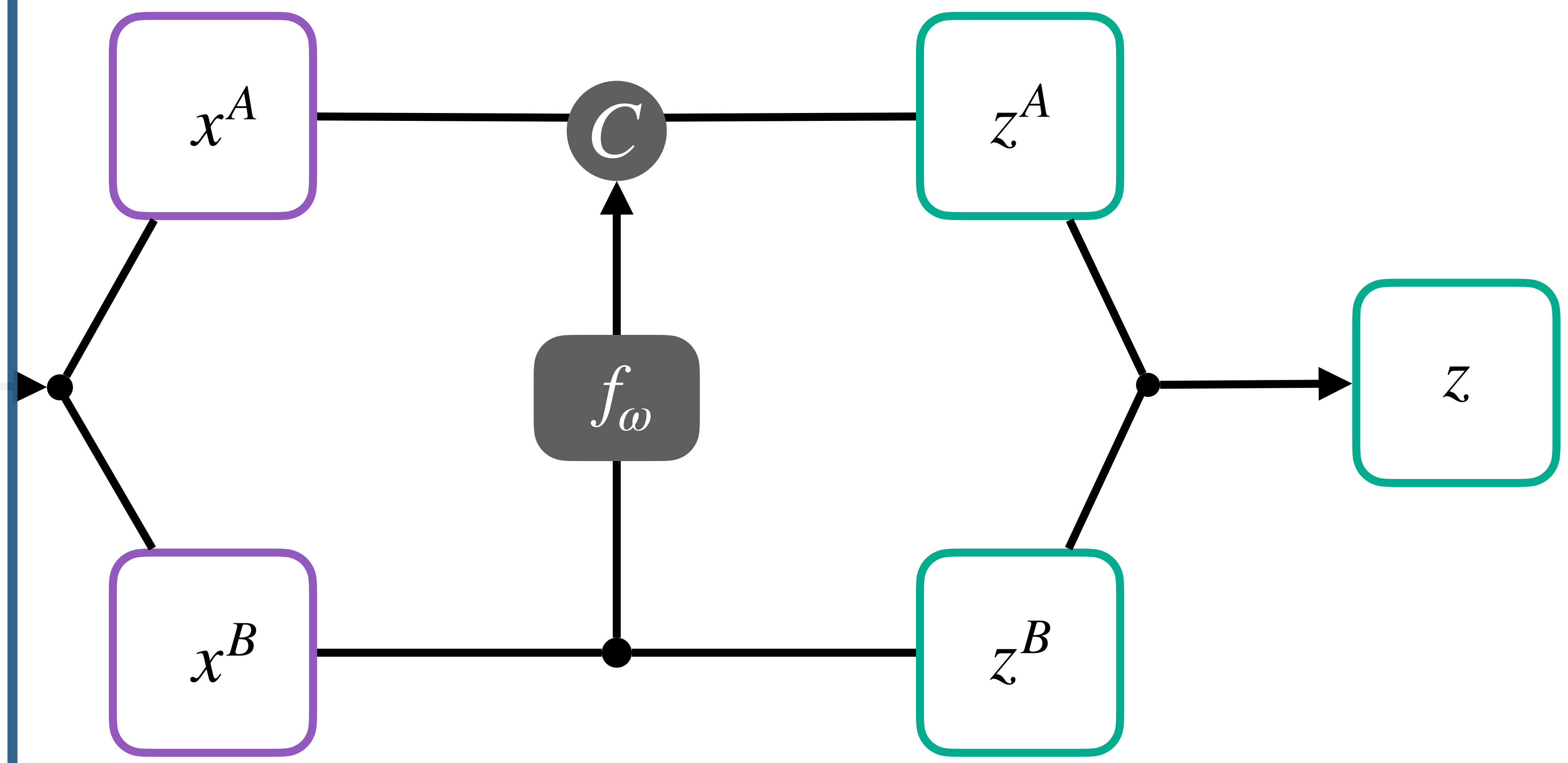
Key facts - Normalizing Flows

- ⊕ Fast training and evaluation
- ⊕ Tractable and fast likelihoods
- ⊖ Reduced flexibility and expressivity

Application dependent

Key facts - Diffusion Model

- ⊕ State-of-the-art in precision
- ⊕ Fast and stable training
- ⊖ Slow evaluation

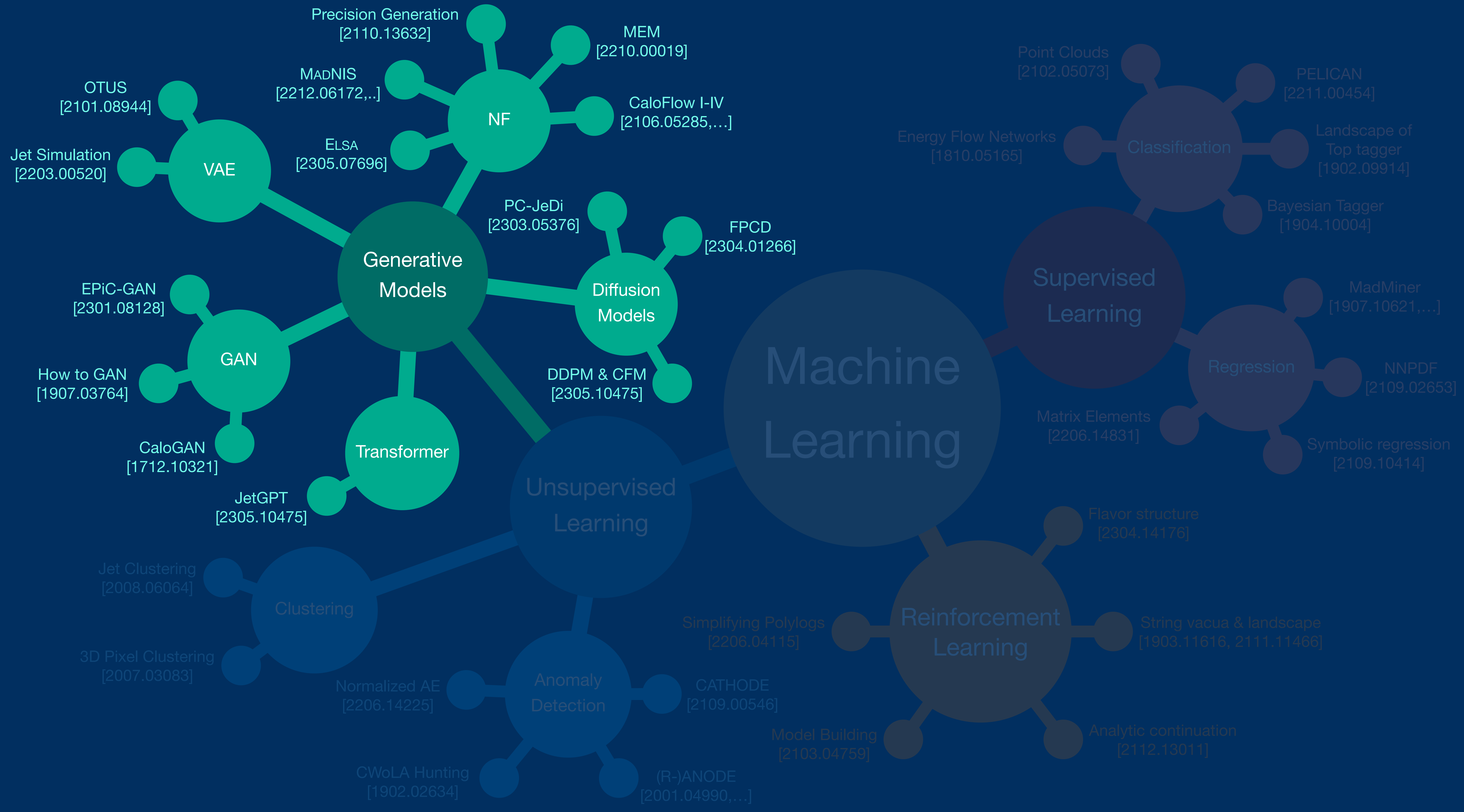


$$J_{ij}(x) = \begin{pmatrix} \frac{\partial C}{\partial x^A} & \frac{\partial C}{\partial f_\omega}\frac{\partial f_\omega}{\partial x^B} \\ 0 & I_m \end{pmatrix}$$

Affine [1605.08803]	$C^A = \alpha_\omega(x^B)\cdot x^A + \mu_\omega(x^B)$
Quadratic [1808.03856]	$C = a_\omega x^2 + b_\omega x + c_\omega$
Rational quadratic [1906.04032]	$C = \dfrac{a_\omega x^2 + b_\omega x + c_\omega}{d_\omega x^2 + e_\omega x + f_\omega}$

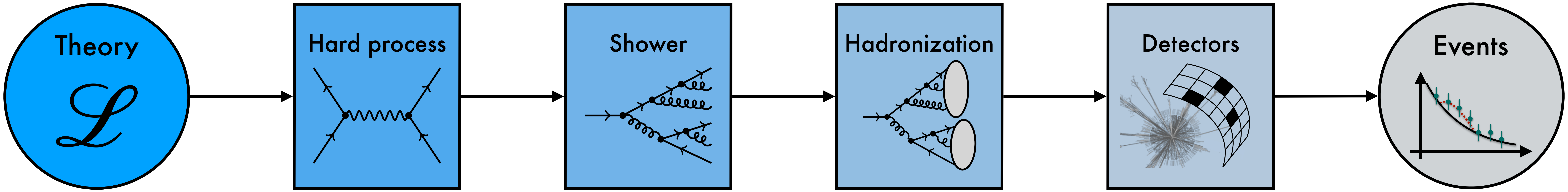
Questions?


Machine Learning
Supervised Learning
Classification
Point Clouds [2102.05073]
PELICAN [2211.00454]
Energy Flow Networks [1810.05165]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Matrix Elements [2206.14831]
Symbolic regression [2109.10414]
Reinforcement Learning
Flavor structure [2304.14176]
Simplifying Polylogs [2206.04115]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CATHODE [2109.00546]
CWoLA Hunting [1902.02634]
(R-)ANODE [2001.04990,…]
Generative Models
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
NF
Precision Generation [2110.13632]
MEM [2210.00019]
MADNIS [2212.06172,..]
CaloFlow I-IV [2106.05285,…]
ELSA [2305.07696]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]

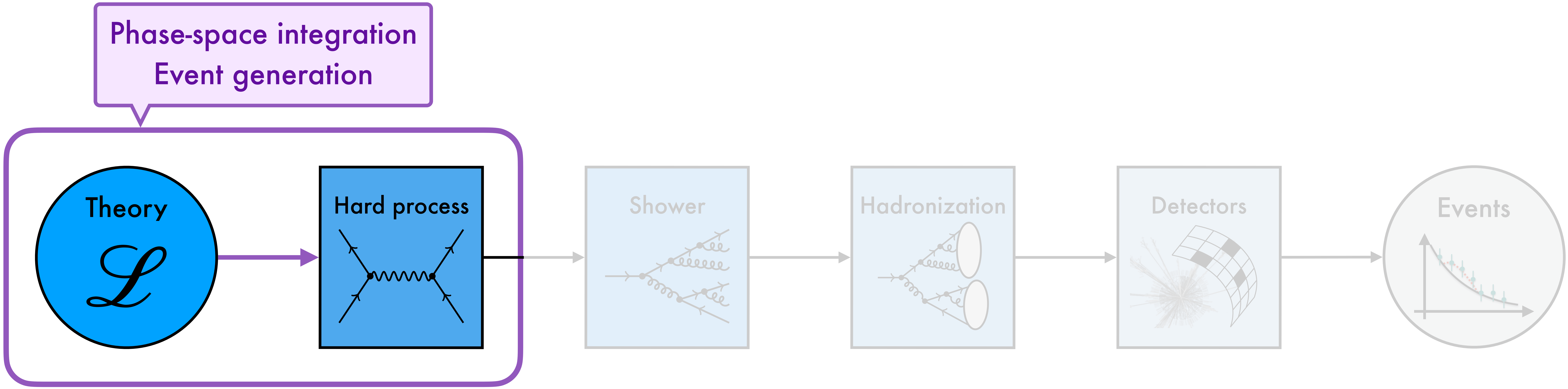

Machine Learning
Generative Models
NF
Precision Generation [2110.13632]
MEM [2210.00019]
MADNIS [2212.06172,..]
CaloFlow I-IV [2106.05285,…]
ELSA [2305.07696]
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CATHODE [2109.00546]
CWoLA Hunting [1902.02634]
(R-)ANODE [2001.04990,…]
Supervised Learning
Classification
Point Clouds [2102.05073]
PELICAN [2211.00454]
Energy Flow Networks [1810.05165]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Matrix Elements [2206.14831]
Symbolic regression [2109.10414]
Reinforcement Learning
Flavor structure [2304.14176]
Simplifying Polylogs [2206.04115]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]


Generative Models
VAE
OTUS
[2101.08944]
Jet Simulation
[2203.00520]
NF
Precision Generation
[2110.13632]
MEM
[2210.00019]
MADNIS
[2212.06172,..]
CaloFlow I-IV
[2106.05285,…]
ELSA
[2305.07696]
Diffusion
Models
PC-JeDi
[2303.05376]
FPCD
[2304.01266]
DDPM & CFM
[2305.10475]
GAN
EPiC-GAN
[2301.08128]
How to GAN
[1907.03764]
CaloGAN
[1712.10321]
Transformer
JetGPT
[2305.10475]
?
Theory
$\mathscr{L}$
Hard process
Shower
Hadronization
Detectors
Events

The LHC simulation chain



The LHC simulation chain + ML



Importance sampling
BDT [1707.00028], NN [1810.11509, 2009.07819]
NF [2001.05486, 2001.05478, 2001.10028, 2005.12719, 2112.09145, 2212.06172, 2311.01548]
Chili [2302.10449]

Surrogate regression
Full weight [2109.11964],
Matrix element [1912.11055, 2002.07516, 2006.16273, 2106.09474, 2107.06625, 2109.11964, 2206.14831, 2301.13562, 2302.04005, 2306.07726]

Example I

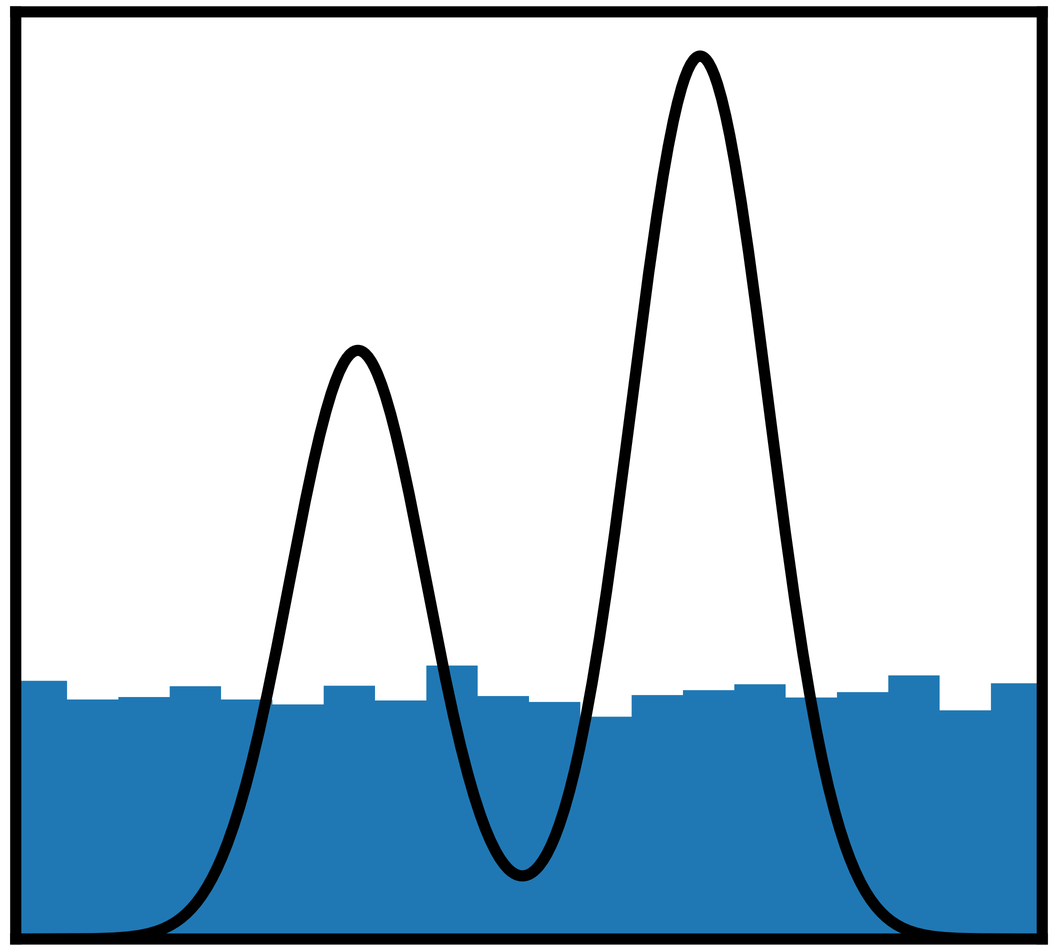
Neural importance sampling with MadNIS

Heimel, Huetsch, Maltoni, Mattelaer, Plehn, RW [2311.01548]
Heimel, RW, Butter, Isaacson, Krause, Maltoni, Mattelaer, Plehn [2212.06172]

Monte Carlo integration

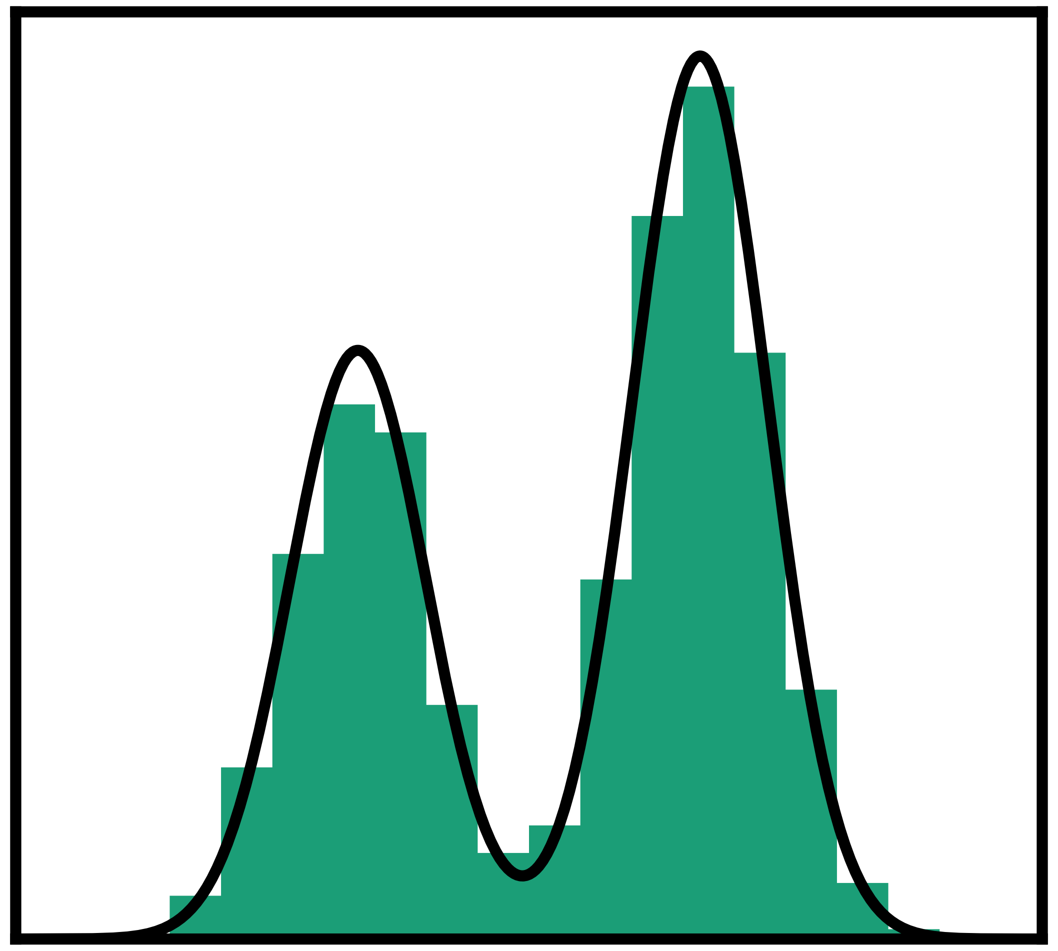
Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$



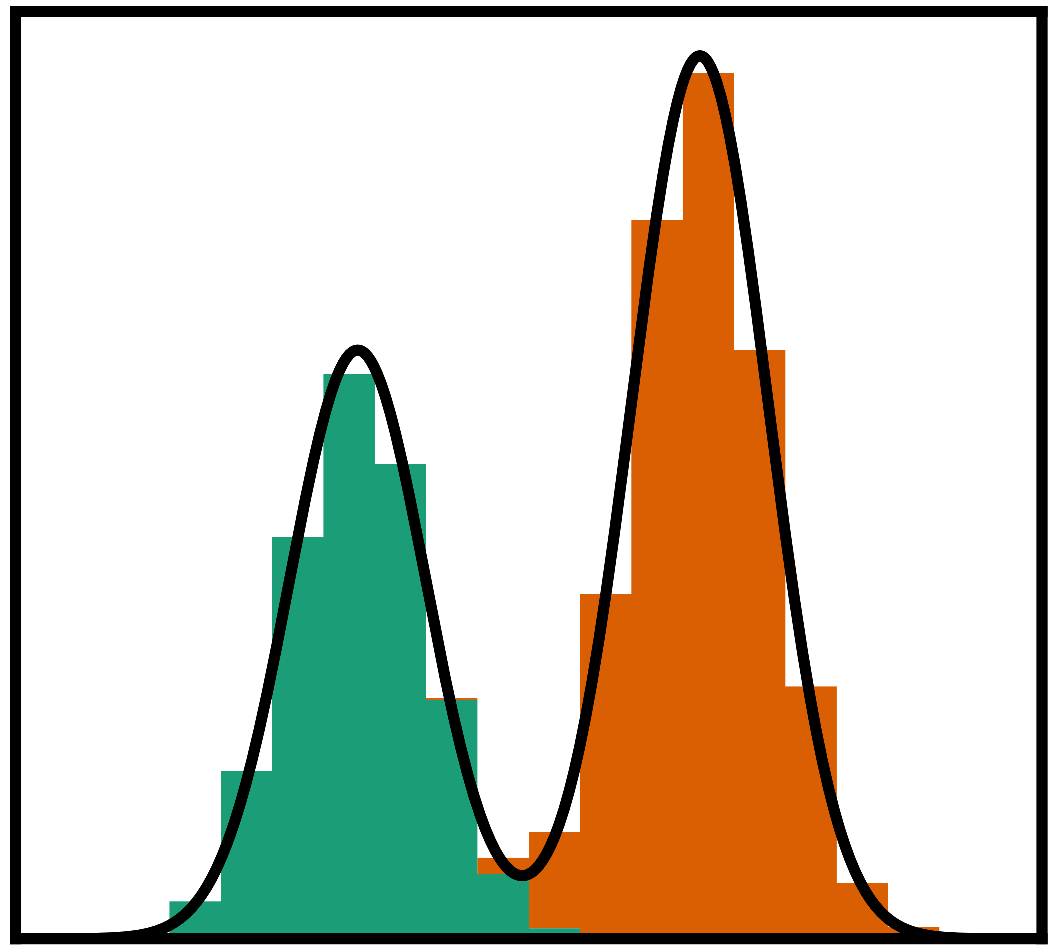
Flat sampling:
inefficient

$$I = \langle f(x) \rangle_{x \sim \text{unif}}$$



Importance sampling:
find p close to f

$$I = \left\langle \frac{f(x)}{p(x)} \right\rangle_{x \sim p(x)}$$



Multi-channel:
one map for each channel

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{p_i(x)} \right\rangle_{x \sim p_i(x)}$$

Event generation

Calculate (differential) cross sections

$$\mathrm{d}\sigma = \frac{1}{\text{flux}} \mathrm{d}x_a \mathrm{d}x_b\, f(x_a) f(x_b)\, \mathrm{d}\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \,\right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $\mathrm{d}\sigma/\mathrm{d}x$

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{p_i(x)} \right\rangle_{x \sim p_i(x)}$$

Channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

Channel mappings

MadGraph: use amplitude structure, …

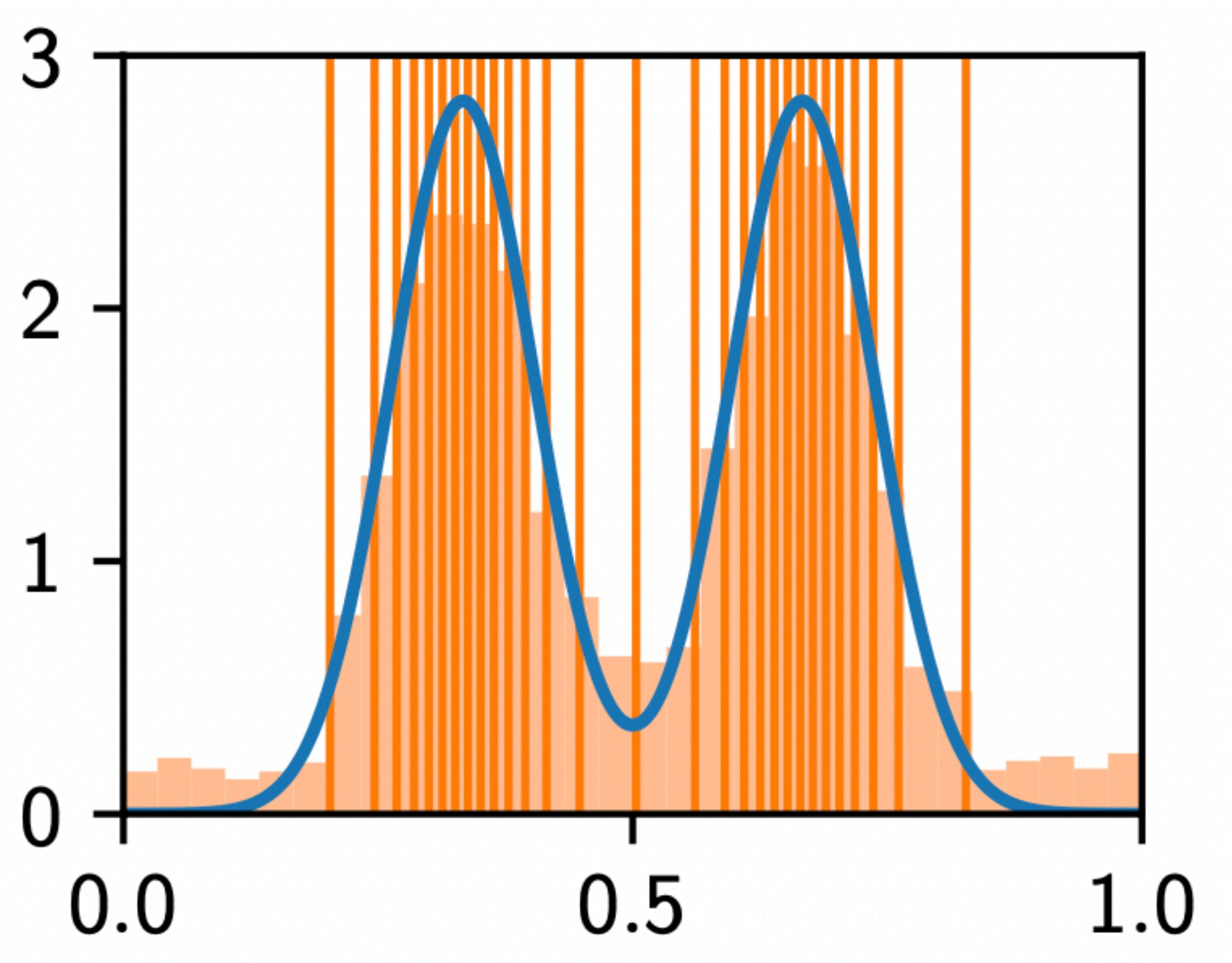
Analytic mappings + refine with **VEGAS**

Importance sampling — VEGAS

Factorize probability

$$p(x) = p(x_1)\cdots p(x_n)$$

Fit bins with equal probability
and varying width

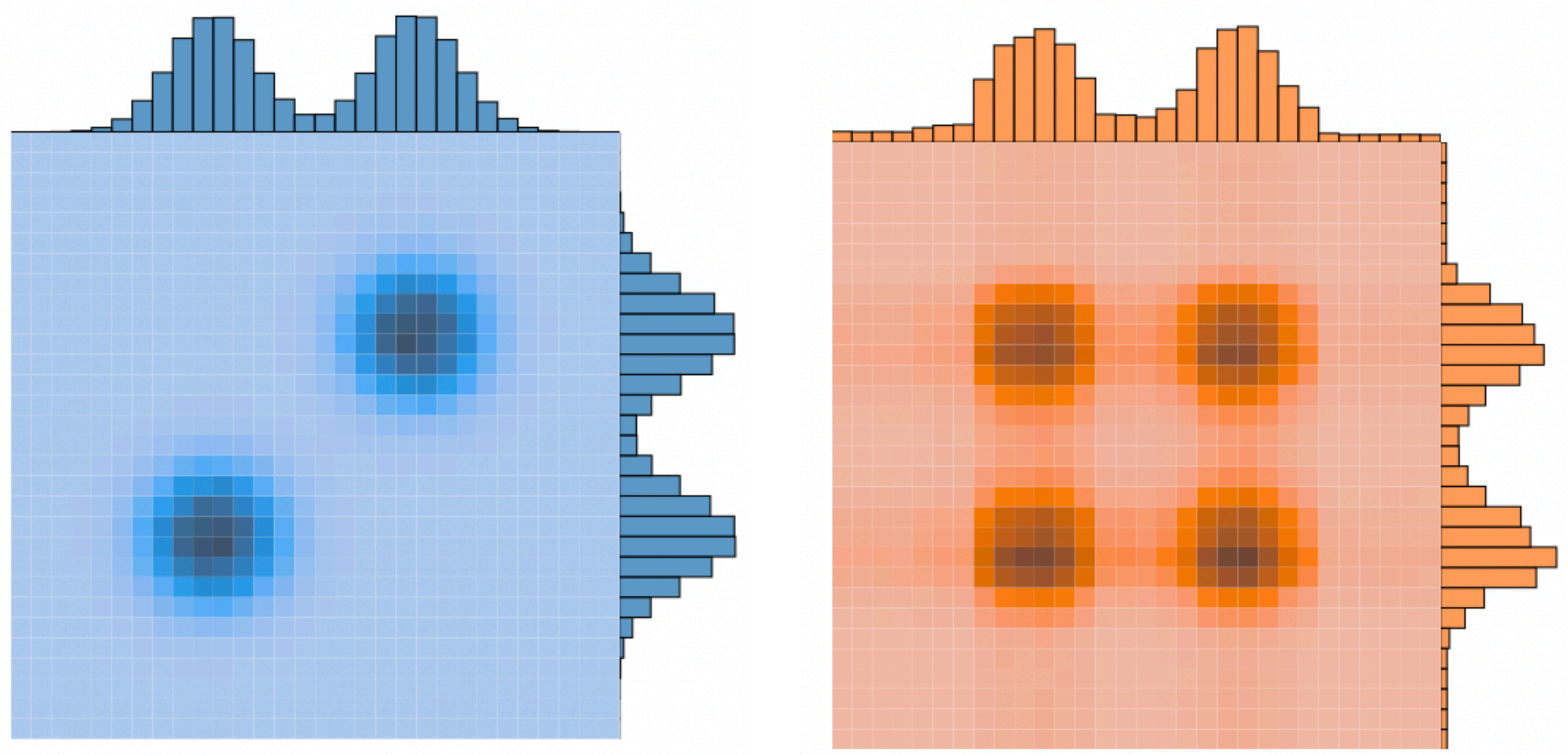


[G. P. Lepage, 1978]

⊕ Computationally cheap

⊖ High-dim and rich peaking functions
→ **slow convergence**

⊖ Peaks not aligned with grid axes
→ **phantom peaks**



Event generation

Calculate (differential) cross sections

$$\mathrm{d}\sigma = \frac{1}{\text{flux}}\mathrm{d}x_a\mathrm{d}x_b\, f(x_a)f(x_b)\,\mathrm{d}\Phi_n \left\langle\, |M_{\lambda,c,\dots}(p_a,p_b\,|\,p_1,\dots,p_n)|^2 \,\right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $\mathrm{d}\sigma/\mathrm{d}x$

$$I = \sum_i \left\langle \alpha_i(x)\,\frac{f(x)}{p_i(x)} \right\rangle_{x\sim p_i(x)}$$

Channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

Channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + refine with **VEGAS**

Event generation + MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \,\right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{p_i^{\omega}(x)} \right\rangle_{x \sim p_i^{\omega}(x)}$$

Channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

Learned channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with VEGAS~~

refine with NF

Event generation + MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

$$I = \sum_i \left\langle \alpha_i^{\xi}(x) \frac{f(x)}{p_i^{\omega}(x)} \right\rangle_{x \sim p_i^{\omega}(x)}$$

Learned Channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

$$\alpha_i(x) \to \alpha_i^{\xi}(x) = \alpha_i^{\text{MG}}(x) \cdot K_i^{\xi}(x)$$

parametrize with **NN**

Learned channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with VEGAS~~

refine with **NF**

MadNIS — Overview

MadNIS

Basic functionality

- Neural Channel Weights
- Normalizing Flow
- MadGraph matrix elements
- MadEvent channel mappings

Improved multi-channeling

- Stratified sampling/ training
- Symmetries between channels
- Channel Dropping
- Partial weight buffering

Improved training

- VEGAS Initialization
- Buffered Training
- Surrogate Integrand

LHC Examples

MadNIS — Overview

Basic functionality

- Neural Channel Weights
- Normalizing Flow
- MadGraph matrix elements
- MadEvent channel mappings

MadNIS

Improved multi-channeling

- Stratified sampling/ training
- Symmetries between channels
- Channel Dropping
- Partial weight buffering

Improved training

- **VEGAS Initialization**
- Buffered Training
- Surrogate Integrand

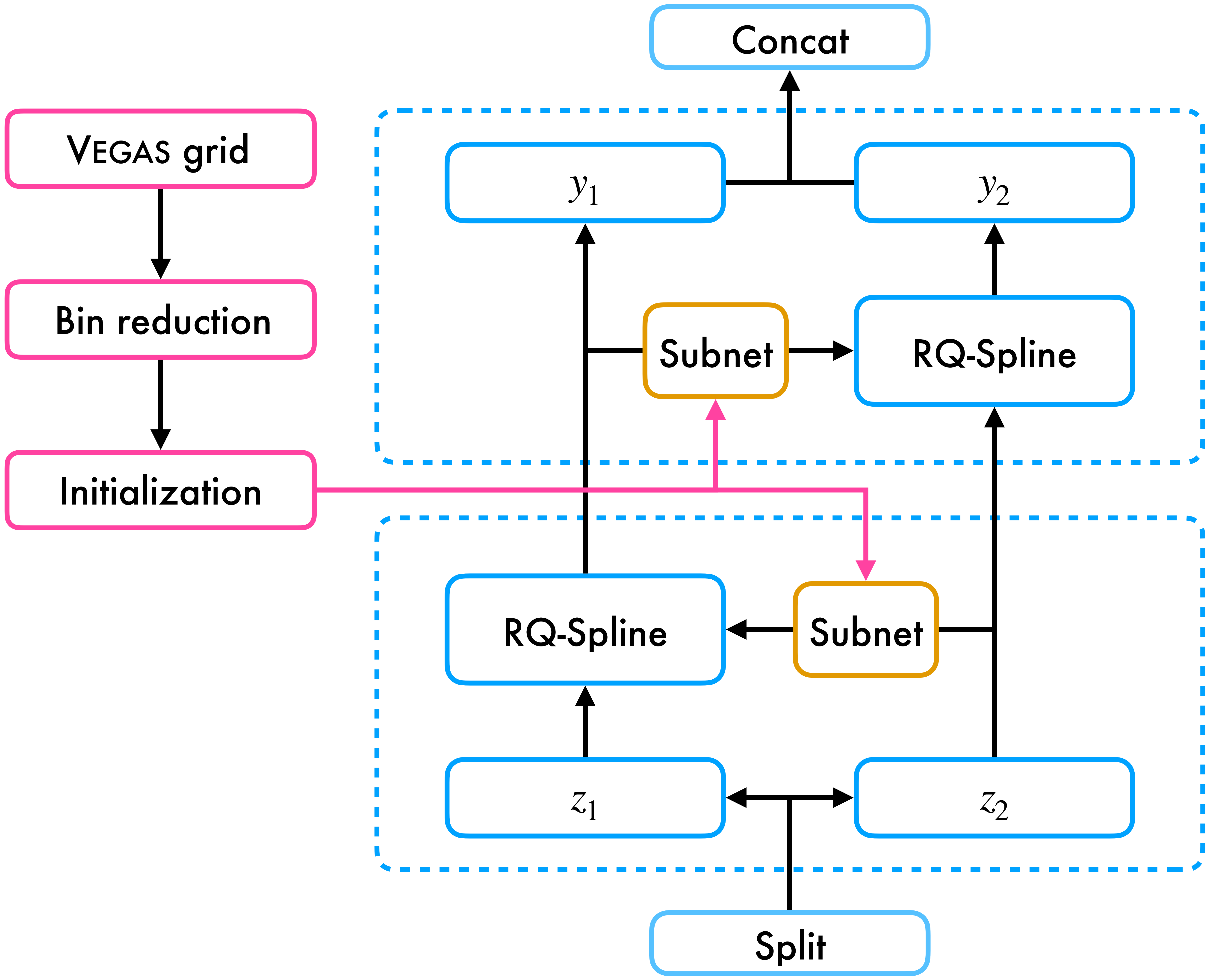
LHC Examples

VEGAS initialization

	VEGAS	Flow
Training	Fast	Slow
Correlations	No	Yes

Combine advantages:

Pre-trained VEGAS grid as starting point for flow training



MadNIS — Overview

MadNIS

Basic functionality

- Neural Channel Weights
- Normalizing Flow
- MadGraph matrix elements
- MadEvent channel mappings

Improved multi-channeling

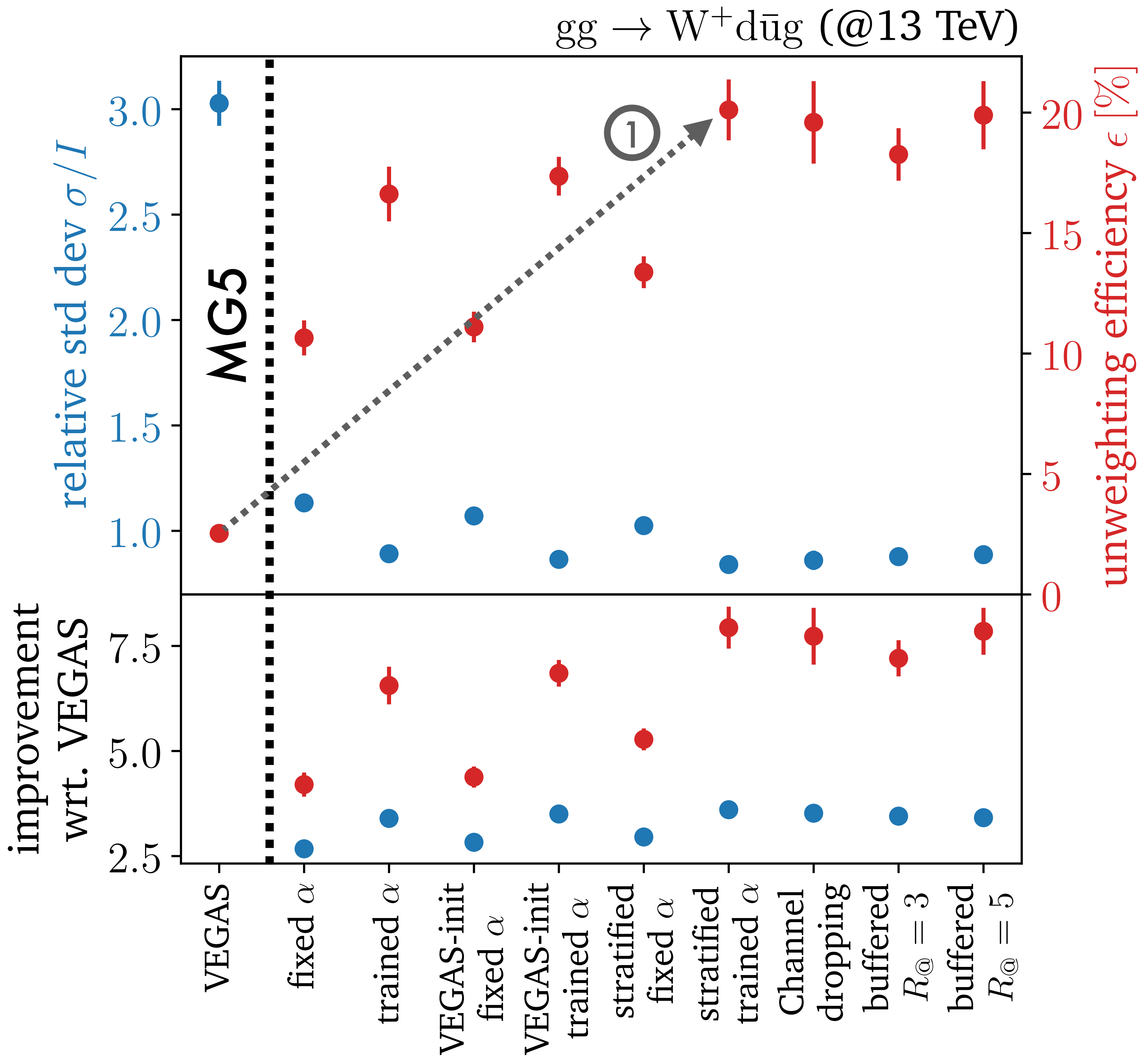
- Stratified sampling/ training
- Symmetries between channels
- Channel Dropping
- Partial weight buffering

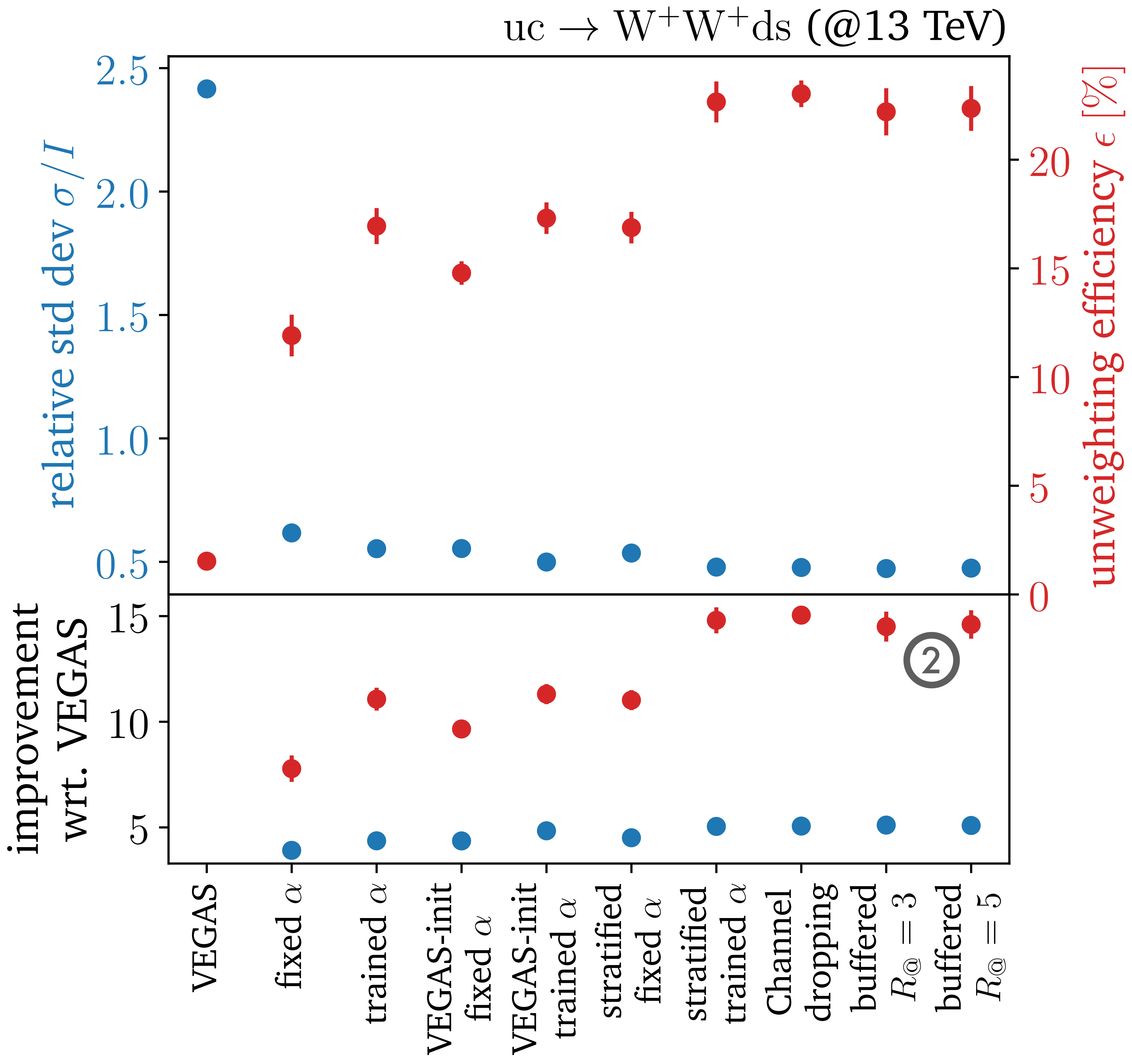
Improved training

- VEGAS Initialization
- Buffered Training
- Surrogate Integrand

LHC Examples

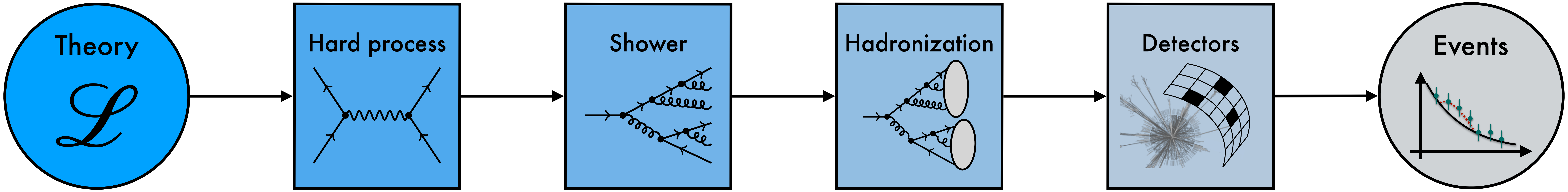
LHC processes



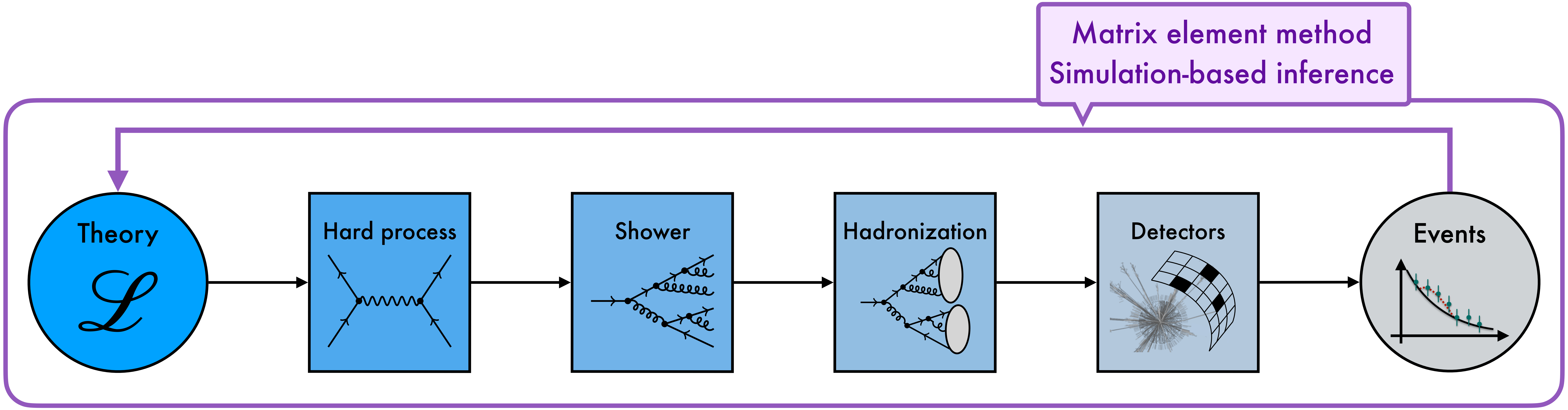


1. excellent results with all improvements
2. Larger improvements for processes with large interference terms

The LHC simulation chain



The LHC simulation chain + ML



Simulation-based inference
[1506.02169, 1601.07913, 1805.00013, 1805.00020, 1805.12244, 1907.10621, 2101.07263, 2210.01680, 2305.10500, 2308.05704,…]

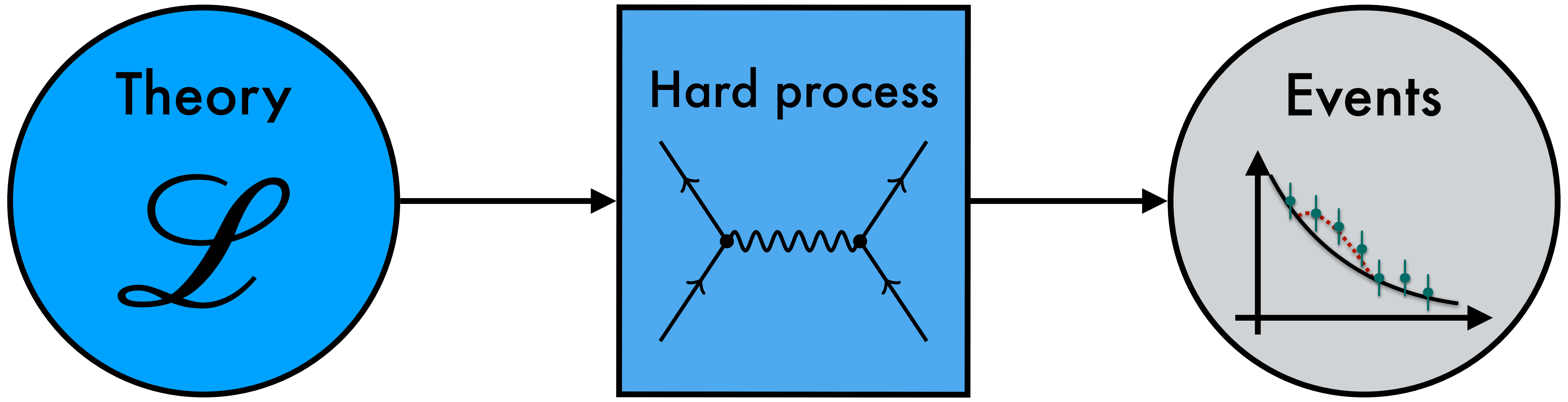
Matrix element method (MEM)
[hep-ex/9808029, hep-ex/0406031, hep-ex/0605118, 1003.1316, 1007.3300, 1010.2263, 1211.3011, 1304.6414, 1502.02485, 1511.05980, 1511.06170, 1512.03429, 1606.03107, 1710.10699, 1712.03266, 1805.08555, 2008.10949, 2210.00019, 2310.07752,…..]

Example II

Matrix Element Method

Heimel, Huetsch, RW, Plehn, Butter [2310.07752]
Butter, Heimel, Martini, Peitzsch, Plehn [2210.00019]

Matrix Element Method — Theory



Theory parameter α

Known from theory

Hard-scattering momenta z_p

Likelihood from differential cross section

$$p(z_p\,|\,\alpha) = \frac{1}{\sigma(\alpha)} \frac{\mathrm{d}\sigma(\alpha)}{\mathrm{d}z_p}$$

Classical analysis

⊖ hand-crafted observables

⊖ binned data

→ not all information is used ☹

Matrix Element Method (MEM)

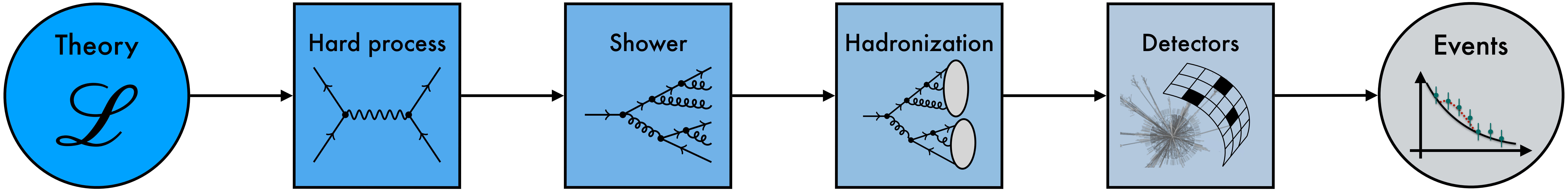
⊕ based on first principles

⊕ estimates uncertainties reliably

⊕ optimal use of information

→ perfect for processes with few events ☺

Matrix Element Method — Reality



Theory parameter
α

Known from theory

Likelihood intractable

Reconstructed momenta
x

MEM master formula: $$p(x\,|\,\alpha) = \int \mathrm{d}z_p \;\; p(z_p\,|\,\alpha) \;\; p(x\,|\,z_p) \;\; \epsilon(z_p)$$

Integrate out hard momenta

Acceptance function

Matrix Element Method + ML

$$p(x\,|\,\alpha) = \int \mathrm{d}z_p \; p(z_p\,|\,\alpha) \; p(x\,|\,z_p) \; \epsilon(z_p)$$

Efficient MC integration

importance sampling:
Normalizing Flow

$$z_p \sim p(z_p\,|\,x, \alpha)$$

Theory knowledge

differential
cross-section

$$\frac{1}{\sigma(\alpha)} \frac{\mathrm{d}\sigma(\alpha)}{\mathrm{d}z_p}$$

Transfer function

density estimation:
Normalizing Flow

solve combinatorics:
Transformer

Transfermer

Acceptance function

learn with simple
Classifier network

LHC example

Single Higgs production with anomalous non-CP-conserving Higgs coupling

↓

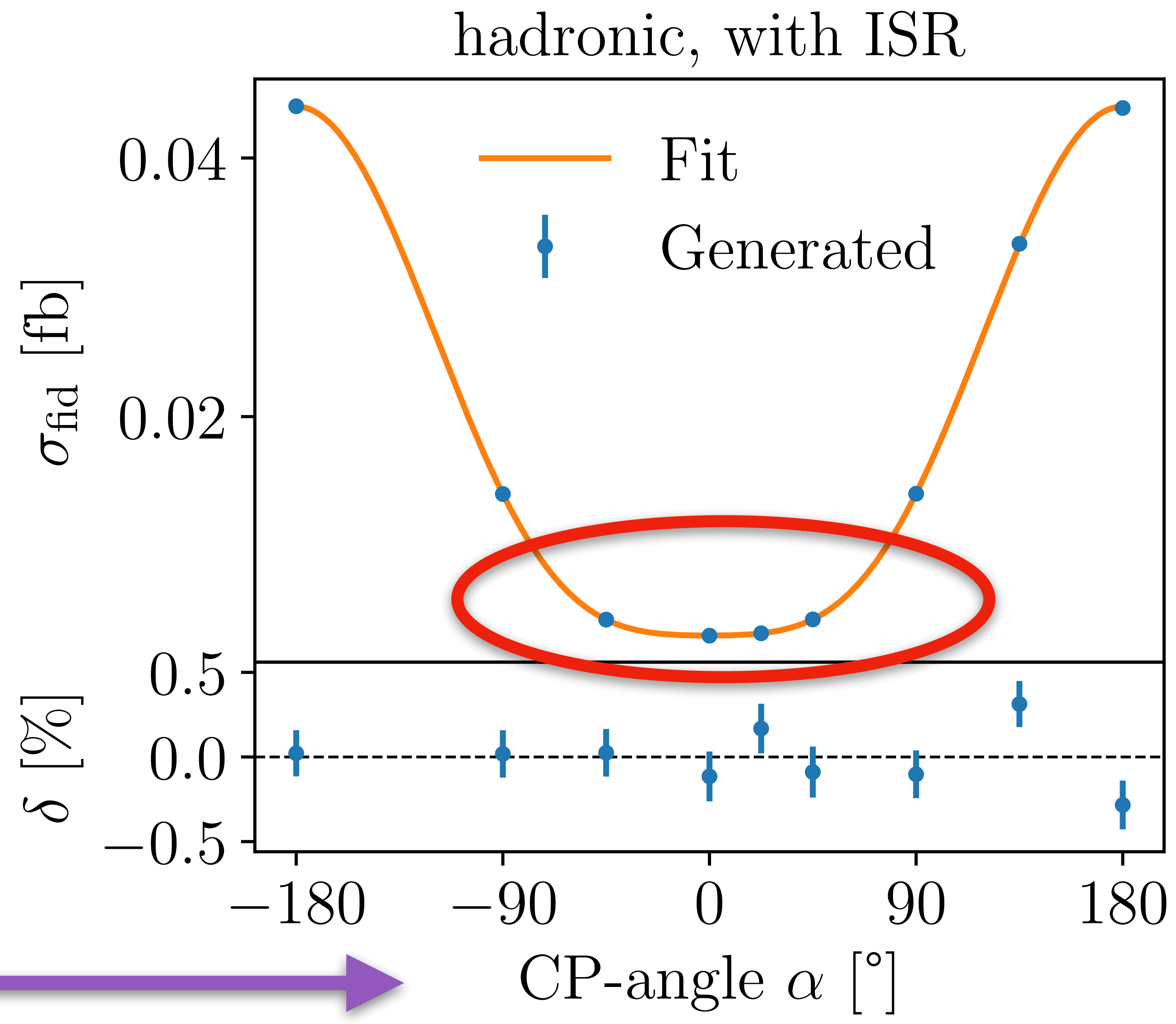
Hadronic decay of top + ISR

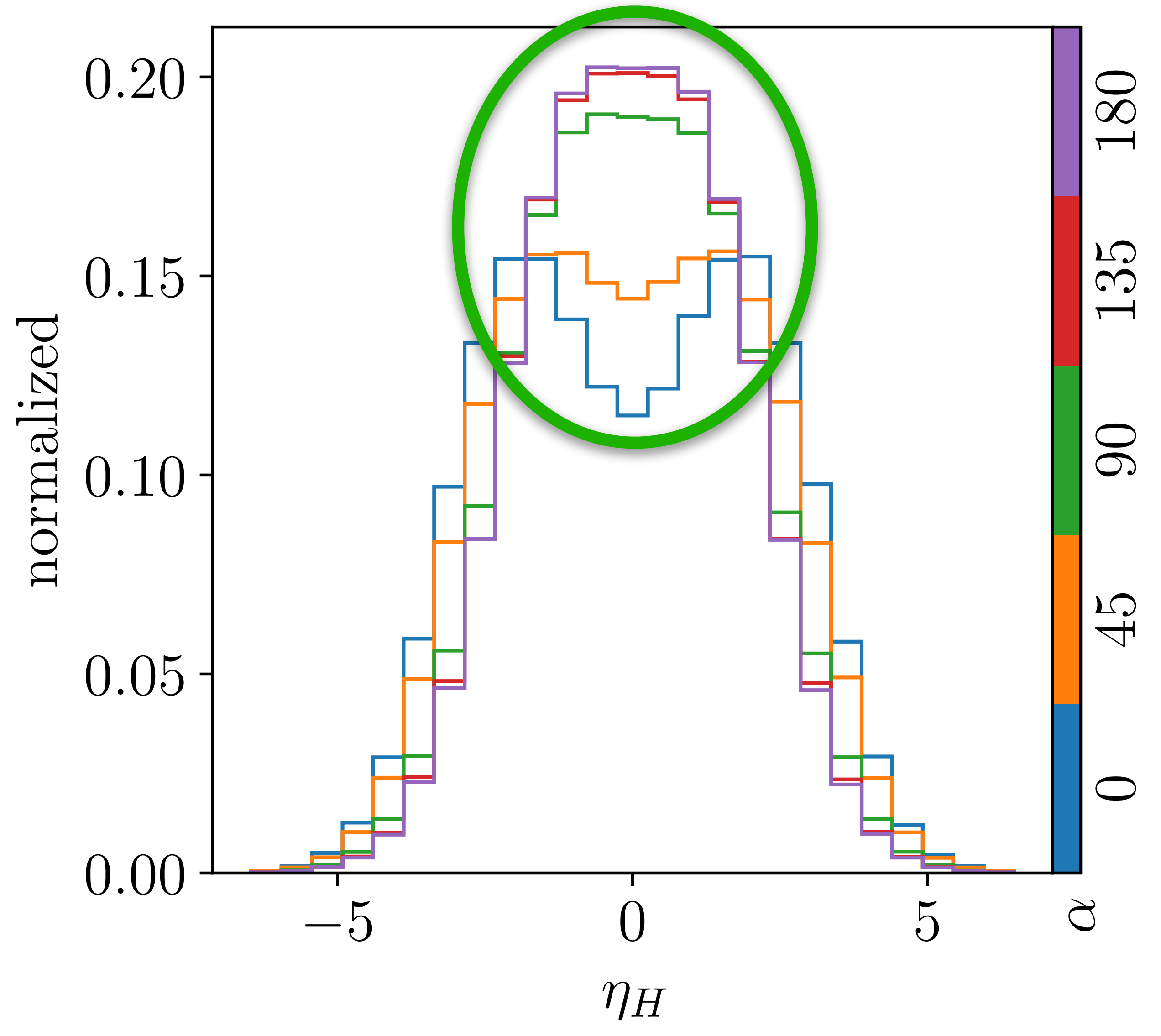
tHq → (bjj) ($\gamma\gamma$) j + QCD jets

↓

$$\mathscr{L}_{t\bar{t}H} = -\frac{y_t}{\sqrt{2}}\left[\cos\alpha\ \bar{t}t + \frac{2}{3}\mathrm{i}\sin\alpha\ \bar{t}\gamma_5 t\right]H$$

Anomalous coupling with **CP-angle α**



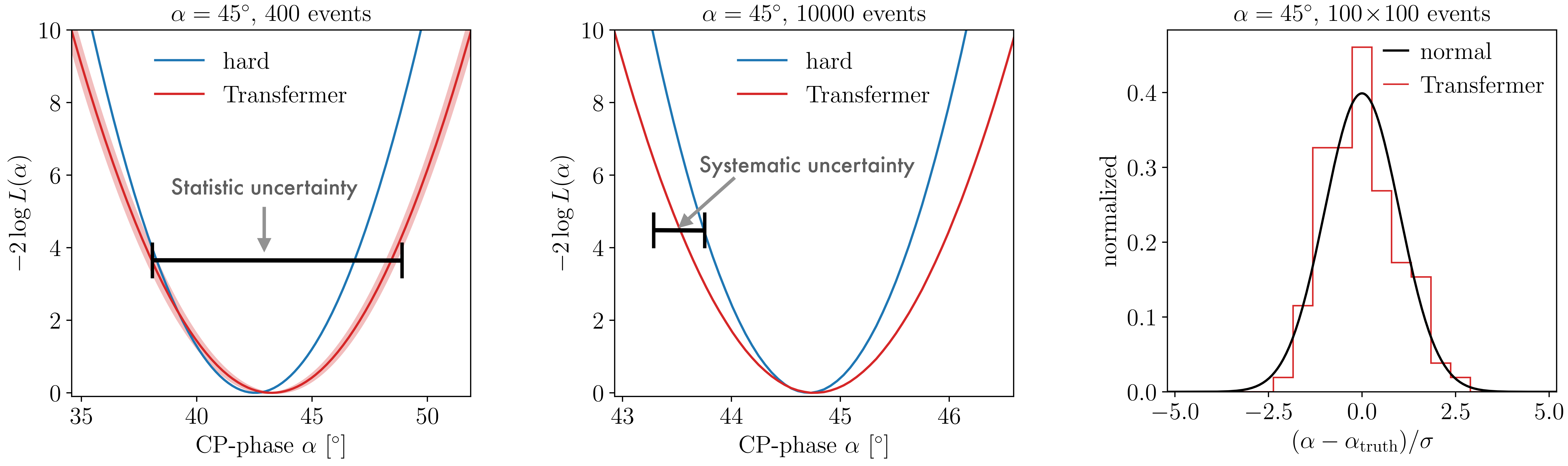


⊖ low total cross section
⊖ low variation of rate

⊕ kinematics sensitive

⟶ ideal use case for **MEM**

Inference results



→ **well-calibrated likelihoods**, both for low and high event count

→ Uncertainty bands: **MC integration error** &
syst. error from limited training statistics (Bayesian NN)