

Machine Learning in HEP

Lecture II — Generative Models for the LHC



UNIVERSITÀ
DEGLI STUDI
DI MILANO



BND School 2026 | Heimbach
Ramon Winterhalder

Plan of attack



Yesterday

1. Introduction to Machine Learning

- Classification and Regression
- Example: *Top Tagging, MadMiner*

Lecture I (90min)

Today

2. Generative Models for the LHC

- Diffusion Models, Normalizing flows, GANs, VAEs
- Example: *MadNIS, MEM-ML*

Lecture II (90min)

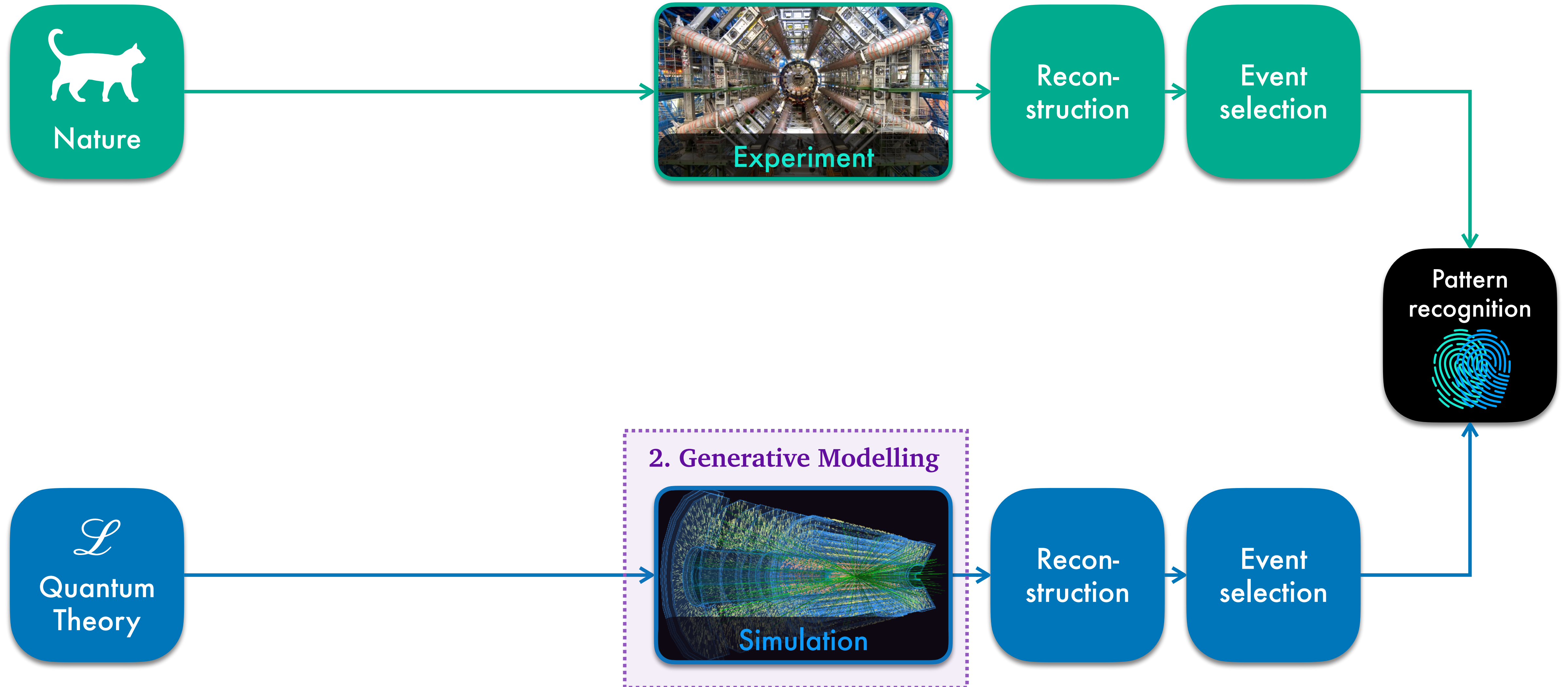
----- COFFEE BREAK -----

3. Anomaly detection

- Autoencoders and CWoLA
- Examples: *CWoLA-Hunting, ANODE, CATHODE,...*

Lecture III (90min)

Machine learning for collider physics



Lecture II

Generative Models for the LHC

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



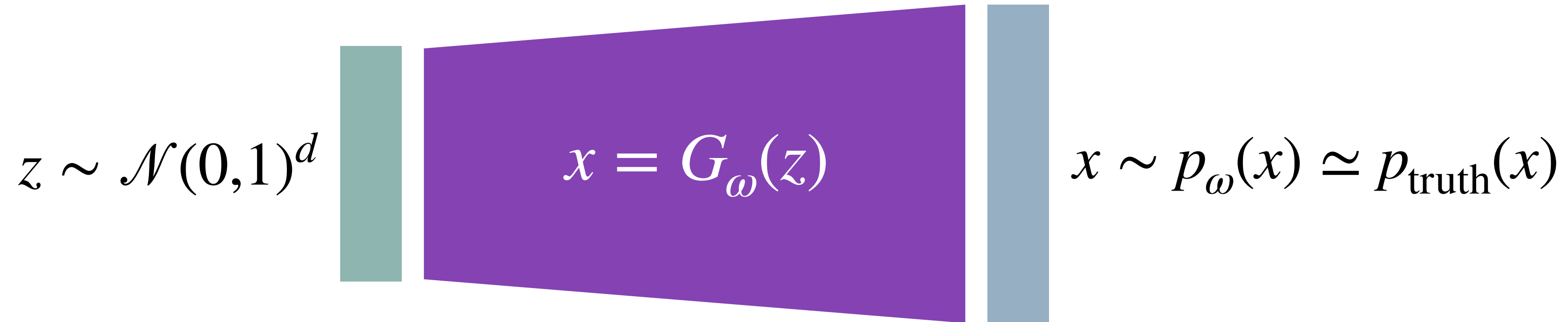
State-of-the-art
image generation

Transformer



State-of-the-art
text generation

What is a generative model?



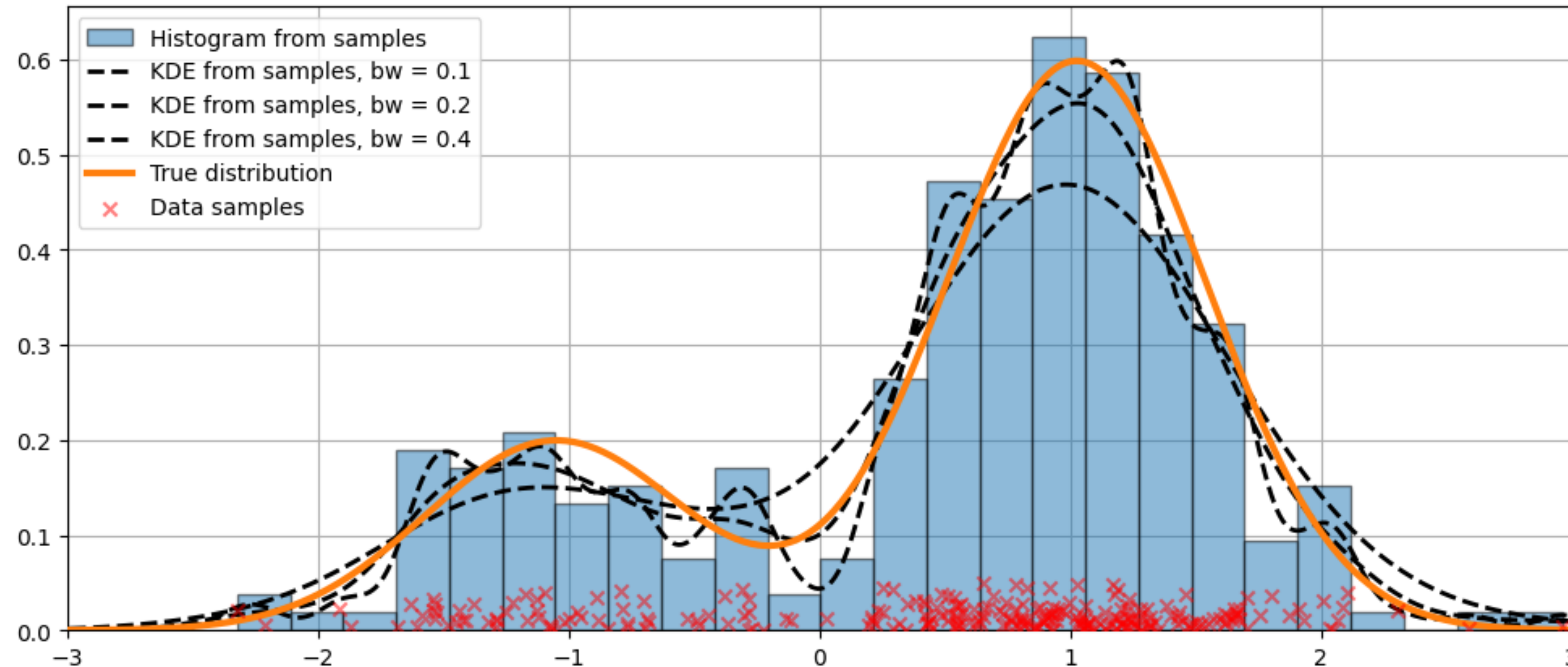
- ▶ A model $G_{\omega}(z)$ that can produce samples $x \sim p_{\omega}(x)$ starting from random noise z
- ▶ The distribution $p_{\text{truth}}(x)$ is usually given as:
 - **explicit** as function (e.g. $d\sigma \propto$ differential cross-section)
 - **implicit** via a set of training data $\{x\} \sim p_{\text{data}}(x)$
 - ▶ ChatGPT: $\text{word} \sim p(\text{word} | \text{previous words})$
 - ▶ Midjourney: $\text{image} \sim p(\text{image} | \text{caption})$

In *particle physics*:

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

(Note: both are **conditional** generative models)

Generative models and density estimation



► Generative models closely connected to *density estimation*

- **Generative model:** sample from $p_{\omega}(x)$
- **Density estimation:** estimate $p_{\omega}(x) \approx p_{\text{truth}}(x)$

Types of deep generative models

Deep generative models



β -VAE

Hierarchical
VAE

**Variational
Autoencoder**

VQ-VAE

Diffusion Probabilistic
Model

Diffusion Model

Score-matching
Model

Conditional Flow
Matching

Wasserstein GAN

**Generative
Adversarial Network**

LS-GAN

Relativistic
GAN

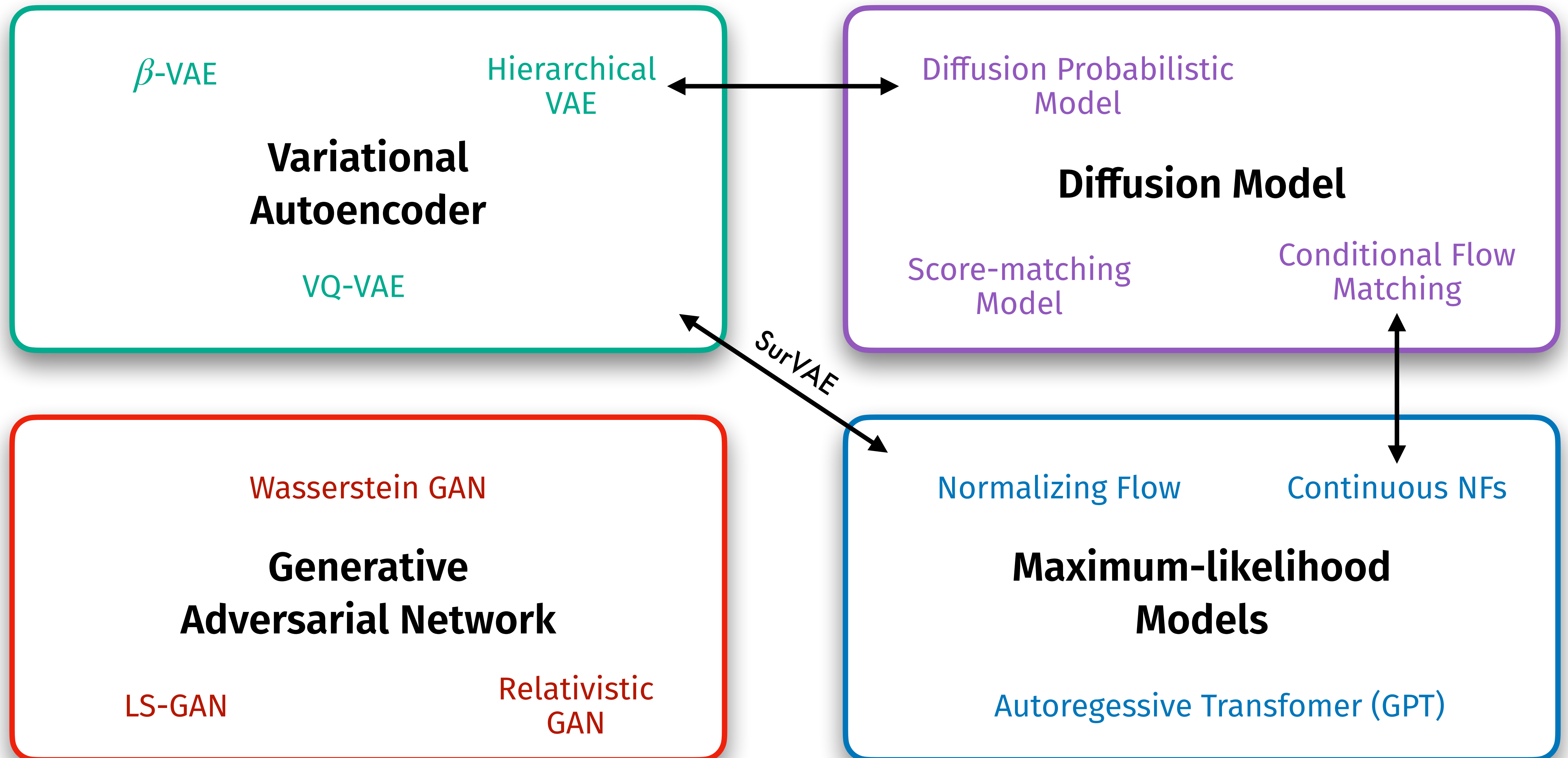
Normalizing Flow

Continuous NFs

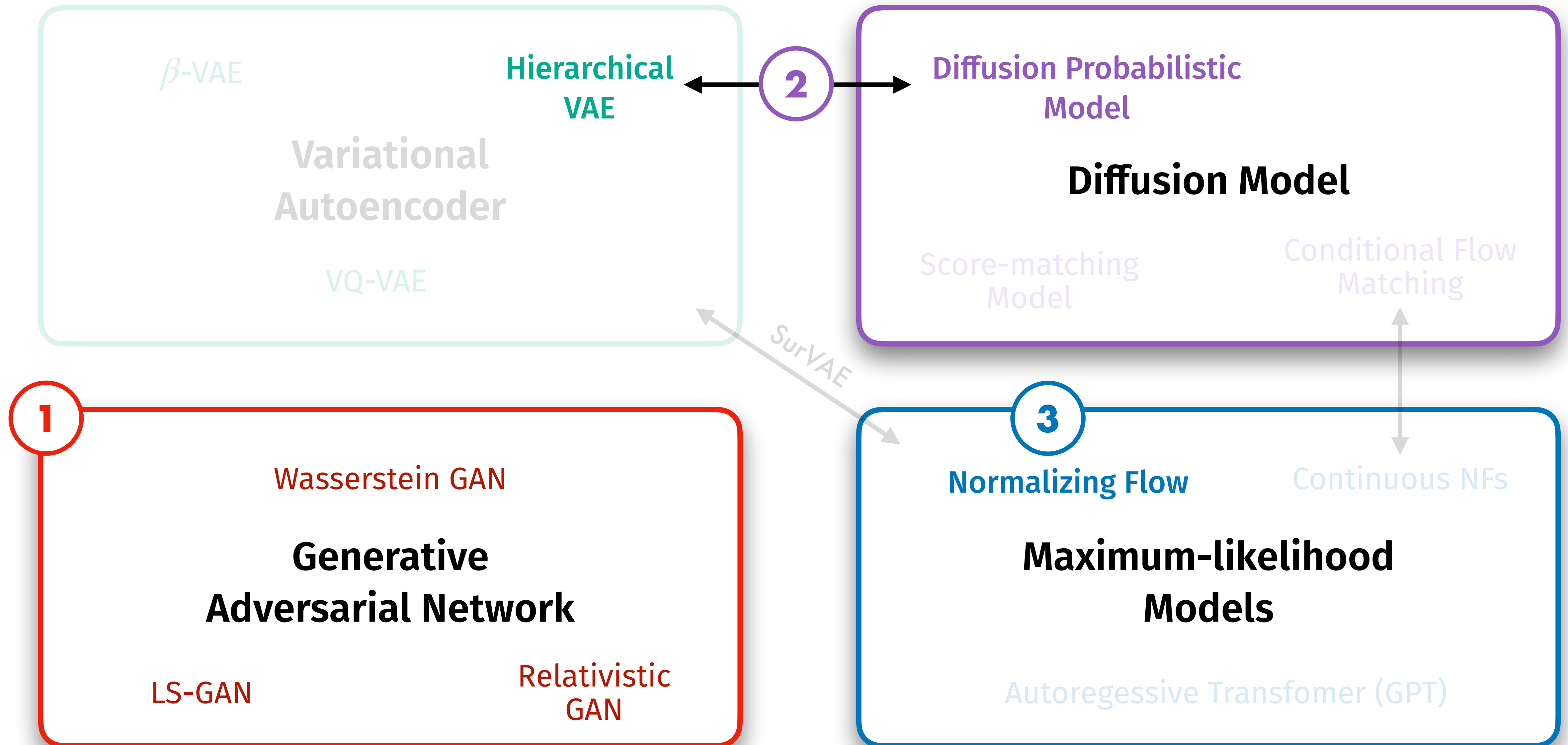
**Maximum-likelihood
Models**

Autoregressive Transformer (GPT)

Deep generative models



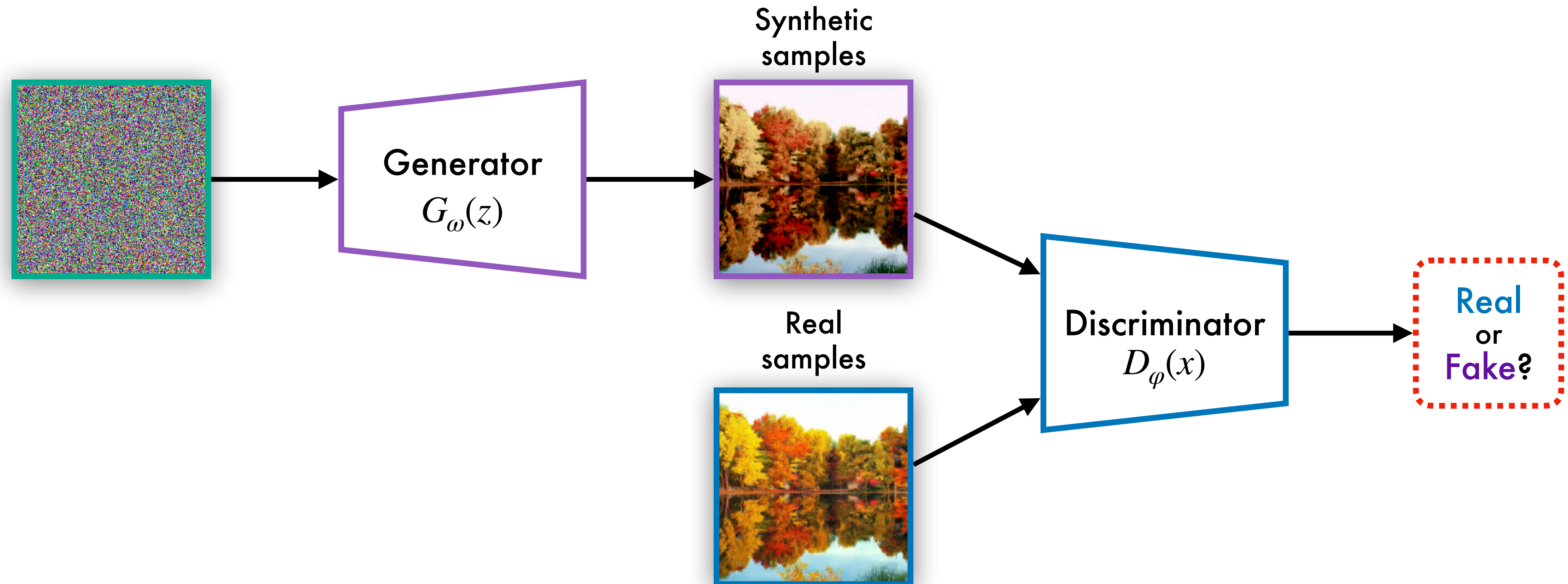
Deep generative models



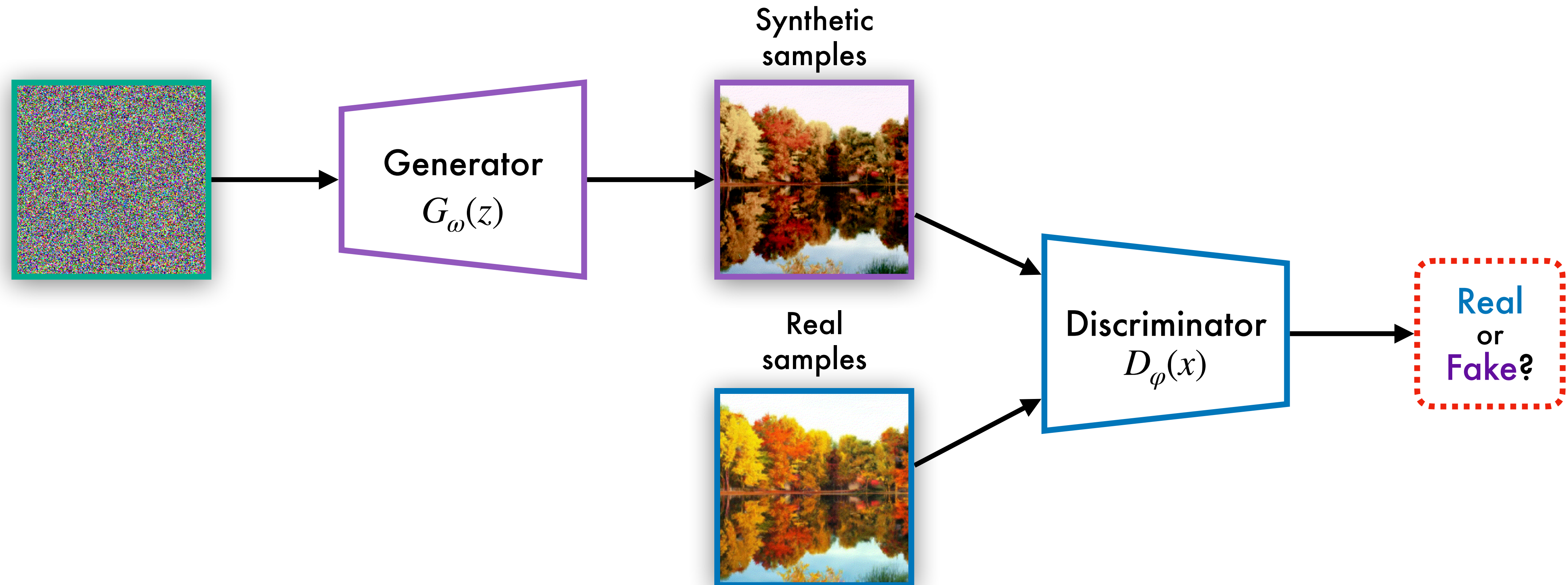
Part I

Generative Adversarial Networks

Generative adversarial network



Generative adversarial network



$$\begin{aligned} L_D &= - \left\langle \log D_{\phi}(x) \right\rangle_{x \sim p_{\text{data}}} - \left\langle \log(1 - D_{\phi}(x)) \right\rangle_{x \sim p_{\hat{\omega}}} \\ &= - \left\langle \log D_{\phi}(x) \right\rangle_{x \sim p_{\text{data}}} - \left\langle \log(1 - D_{\phi}(G_{\hat{\omega}}(z))) \right\rangle_{z \sim p_{\text{latent}}} \end{aligned}$$

Iterative
Training

$$\begin{aligned} L_G &= \left\langle \log(1 - D_{\hat{\phi}}(x)) \right\rangle_{x \sim p_{\omega}} \\ &= \left\langle \log(1 - D_{\hat{\phi}}(G_{\omega}(z))) \right\rangle_{z \sim p_{\text{latent}}} \end{aligned}$$

Generative adversarial network

13

Problems with GANs

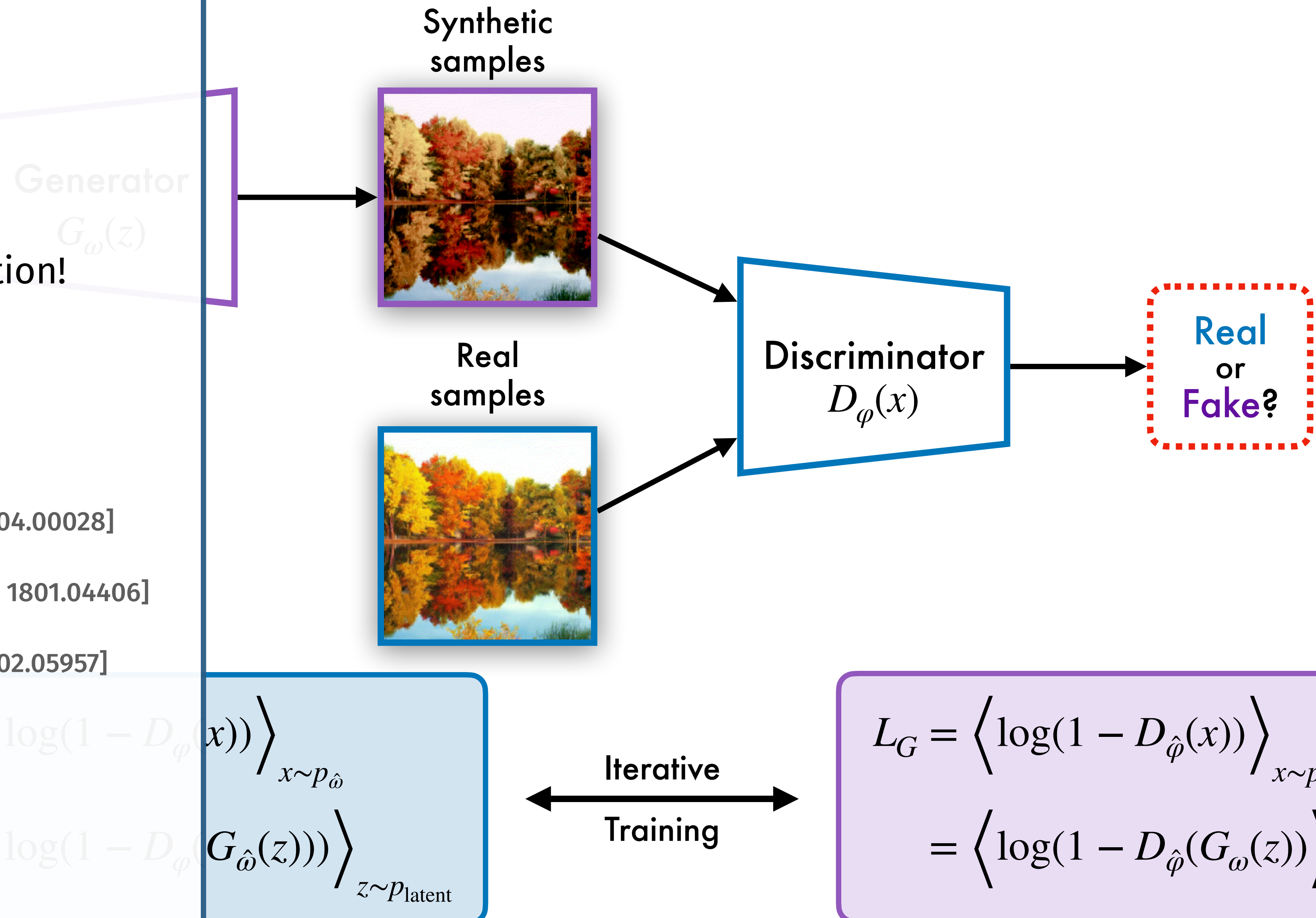
- ⊖ Min-max training unstable
→ vanishing gradients
- ⊖ Metric for success?
→ loss only shows competition!
- ⊖ Mode collapse

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]
- ...



Part II

Diffusion Models

Midjourney

Dall-E

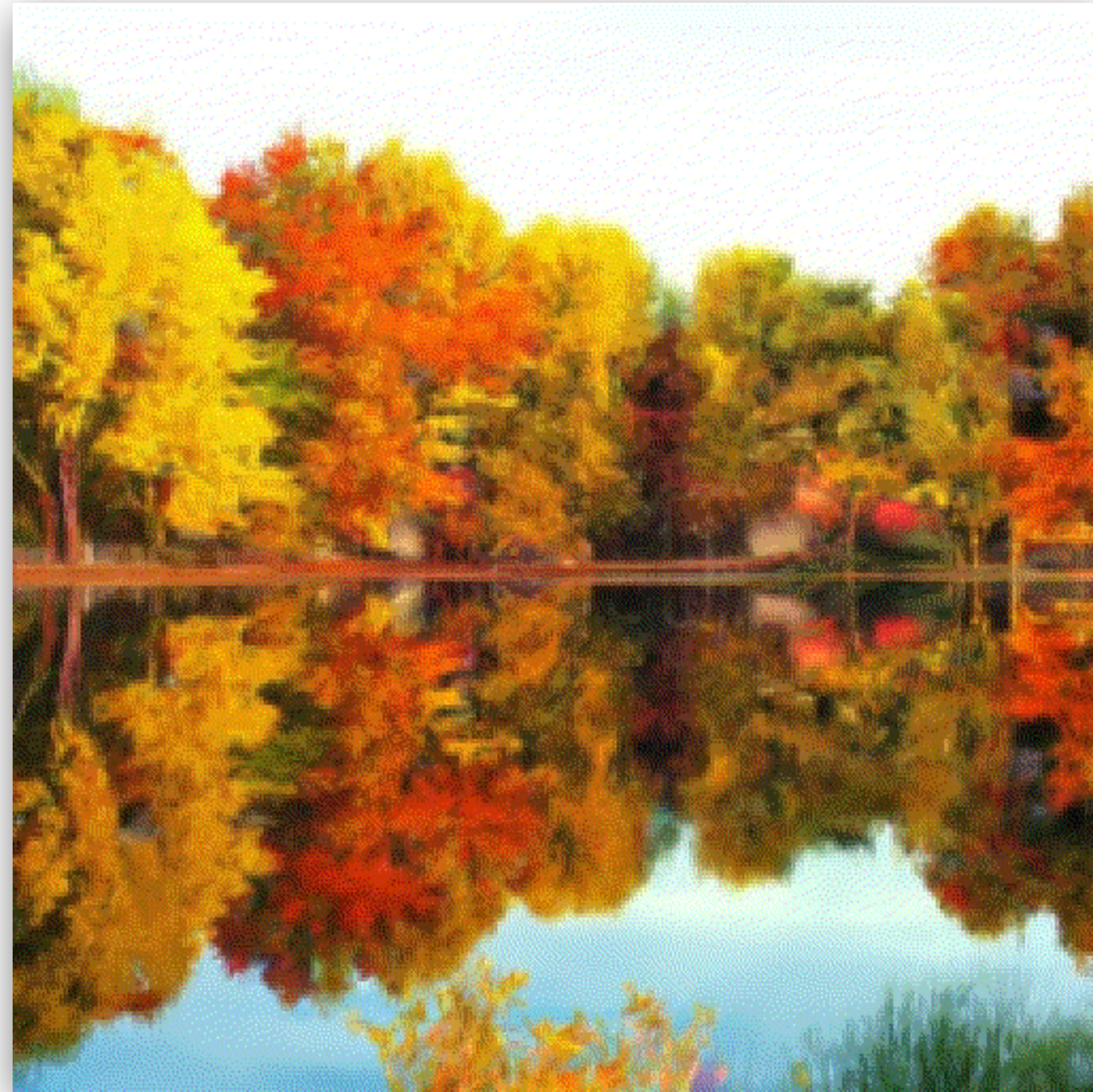
Recently...

Stable Diffusion



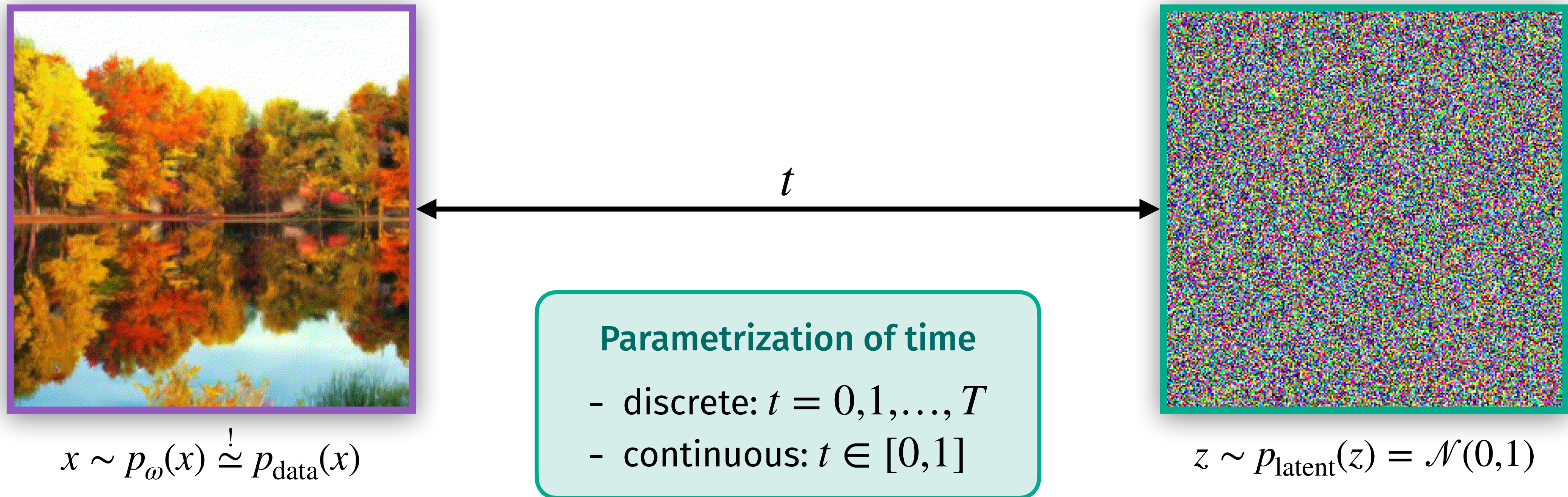
Midjourney AI

Diffusion Models



Diffusion Models

Define mapping as *time-dependent* diffusion process



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

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

VAEs Revisited

18

Bayes' Theorem

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

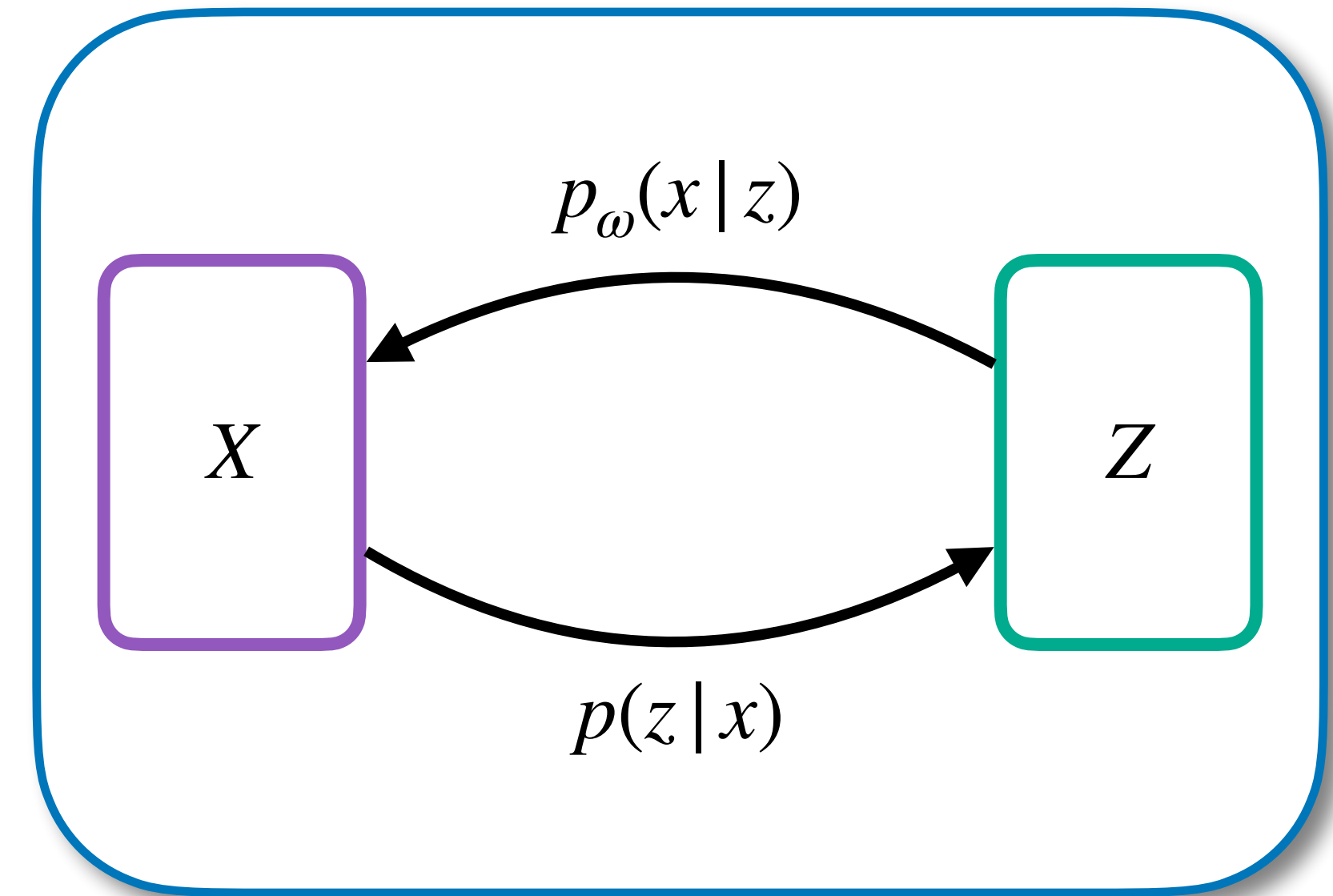
← Prior
← Evidence

Problem: evidence intractable

$$p(x) = \int dz p(x|z)p(z)$$

Variational inference

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



Likelihood $p_\omega(x|z)$ ← Decoder

Posterior: $p(z|x)$ ← Want!

Match!

Evidence Lower Bound — ELBO

Minimize KL

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



Maximize ELBO



ELBO $\neq$ Elmo

Evidence Lower Bound — ELBO

20

Minimize KL

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

Maximize ELBO

$$\begin{aligned} \log p(x) &= \int dz q_\phi(z|x) \log p(x) \\ &= \int dz q_\phi(z|x) \log \frac{p_\omega(x|z) p(z)}{p(z|x)} \end{aligned}$$

Bayes' theorem

Bayes' Theorem

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

Evidence Lower Bound — ELBO

22

Minimize KL

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

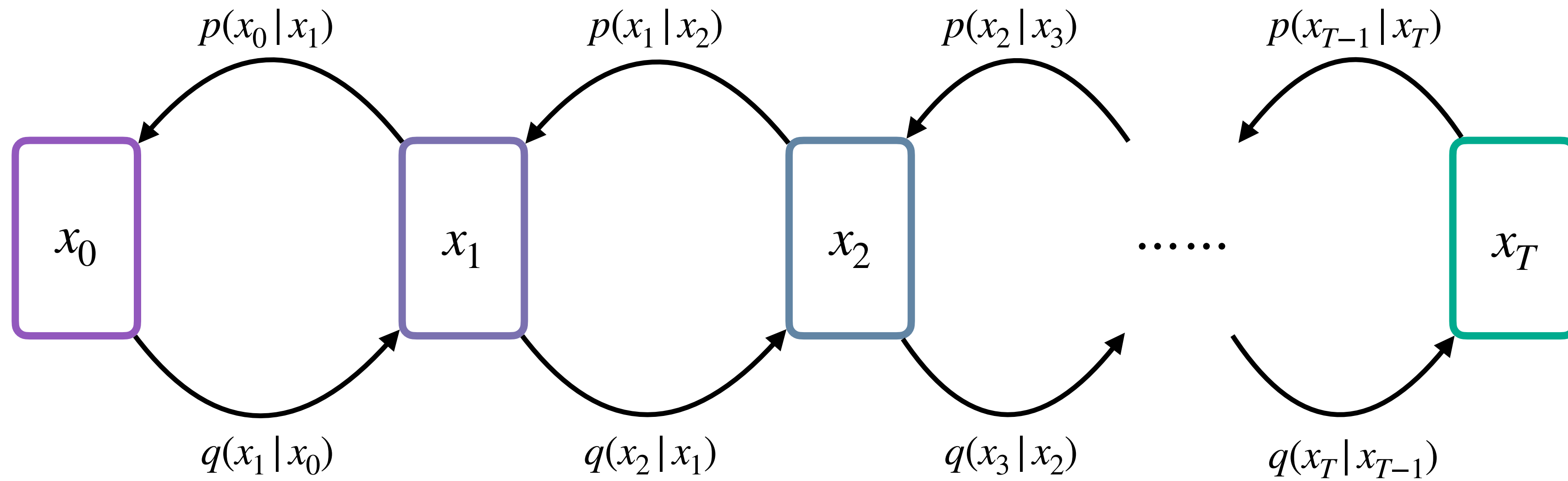
Maximize ELBO

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

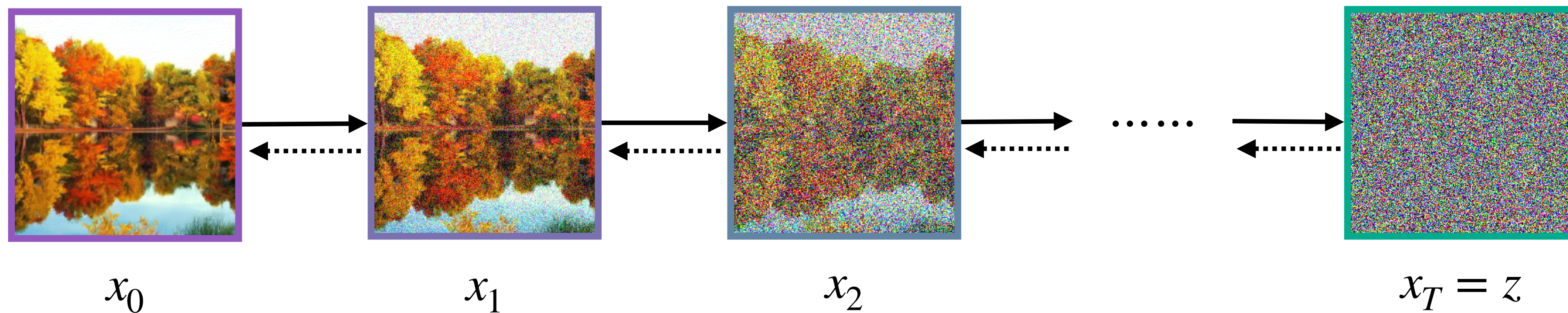
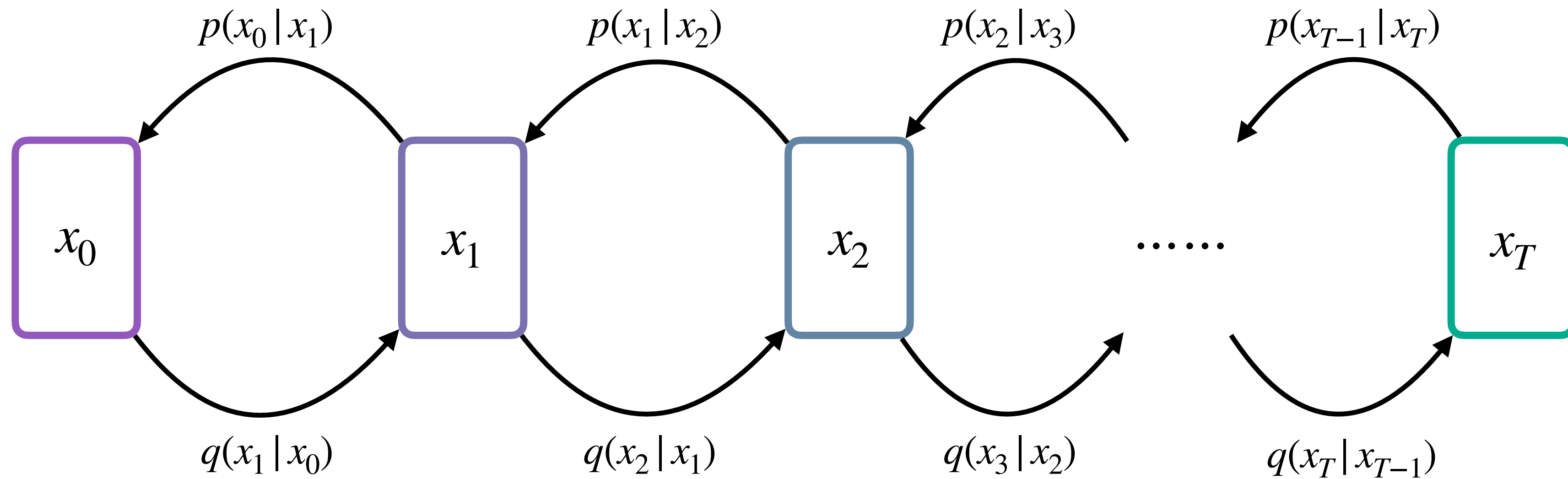
SurVAE [2007.02731]

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

Hierarchical VAEs



Denoising Diffusion Model (DDPM)



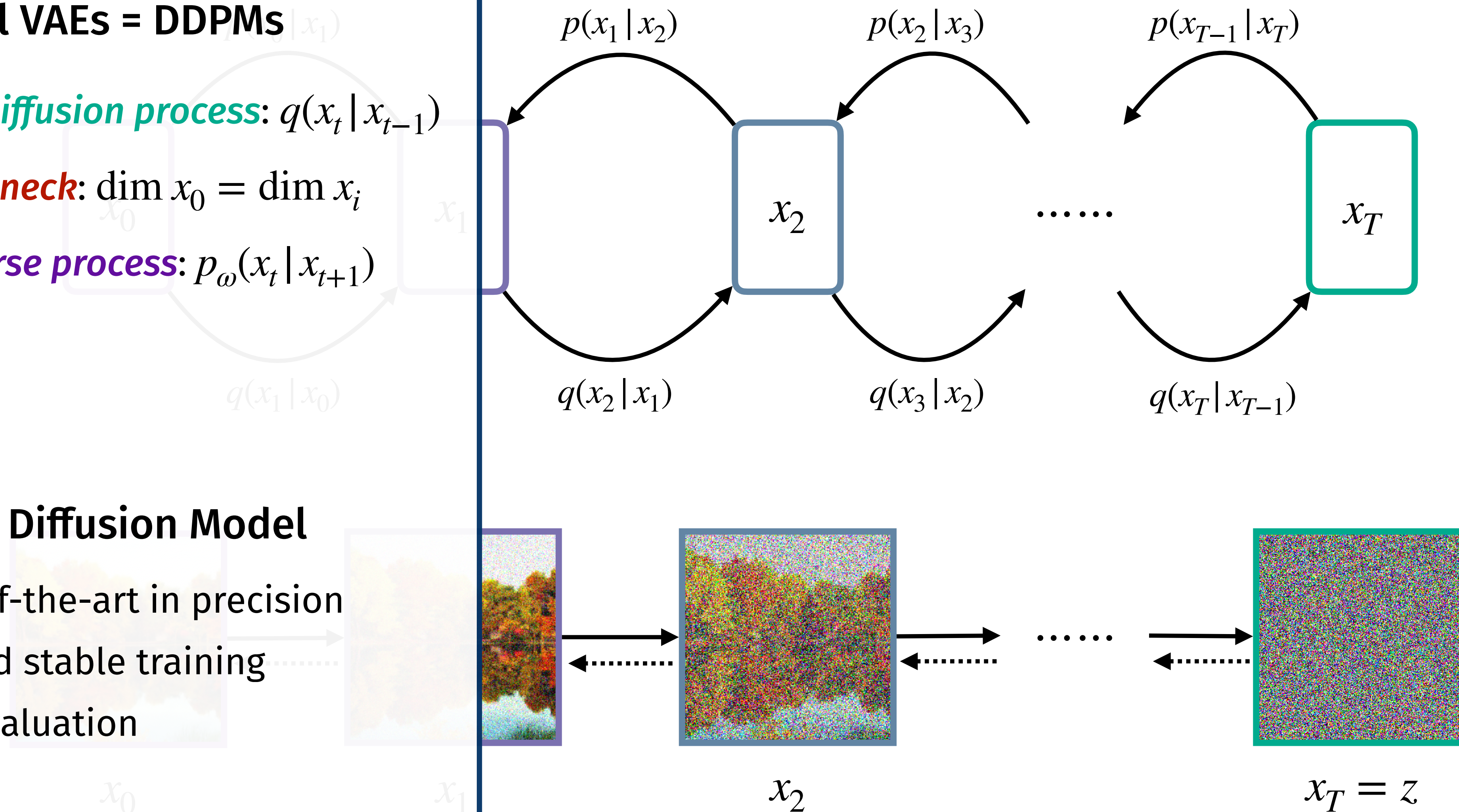
Denoising Diffusion Model (DDPM)

Hierarchical 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

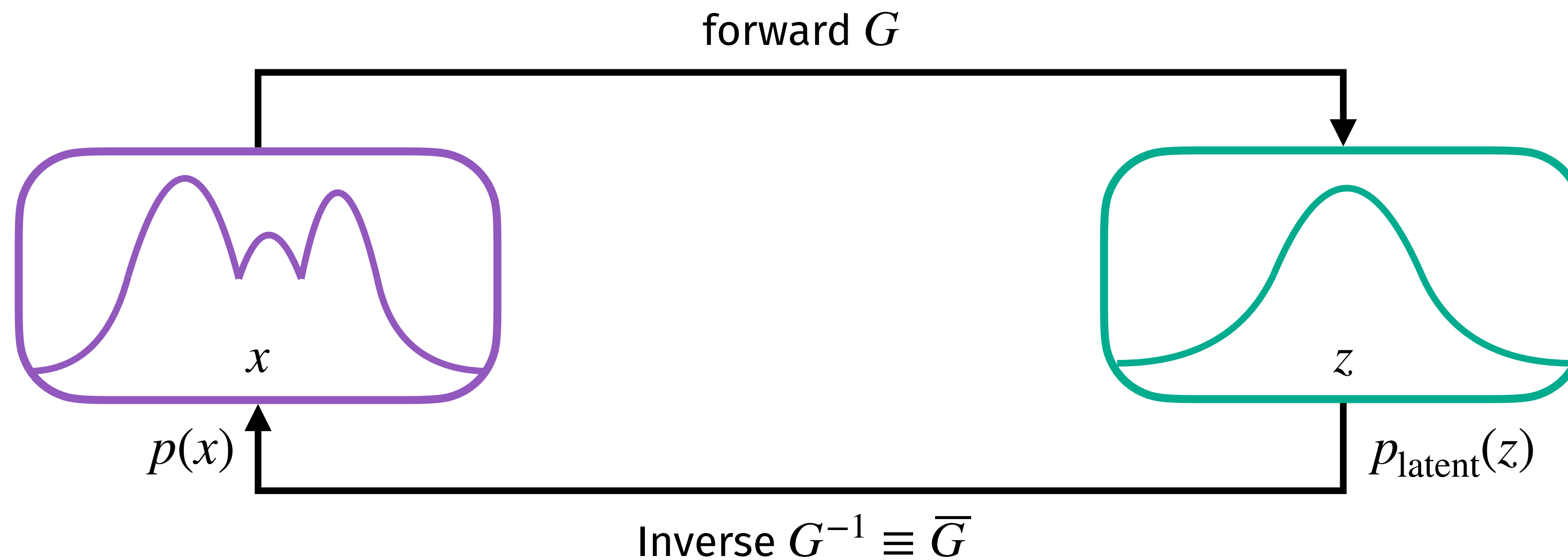
- ⊕ State-of-the-art in precision
- ⊕ Fast and stable training
- ⊖ Slow evaluation



Part III

Normalizing Flows

Normalizing flow — Basics



Conservation of probability:

$$p(x) dx = p_{\text{latent}}(z) dz$$

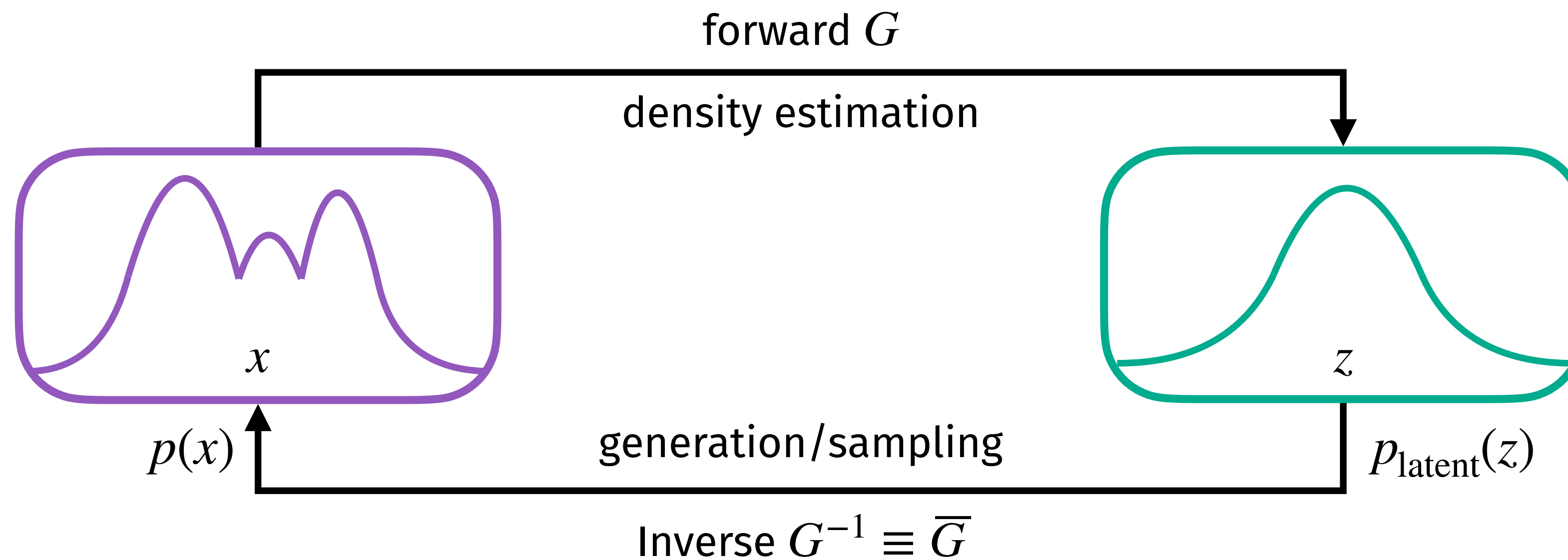
with

$$z = G_{\omega}(x) \quad x = \bar{G}_{\omega}(z)$$

Change-of-variables formula:

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

Normalizing flow — Basics



Conservation of probability:

$$p(x) dx = p_{\text{latent}}(z) dz$$

with

$$z = G_{\omega}(x) \quad x = \bar{G}_{\omega}(z)$$

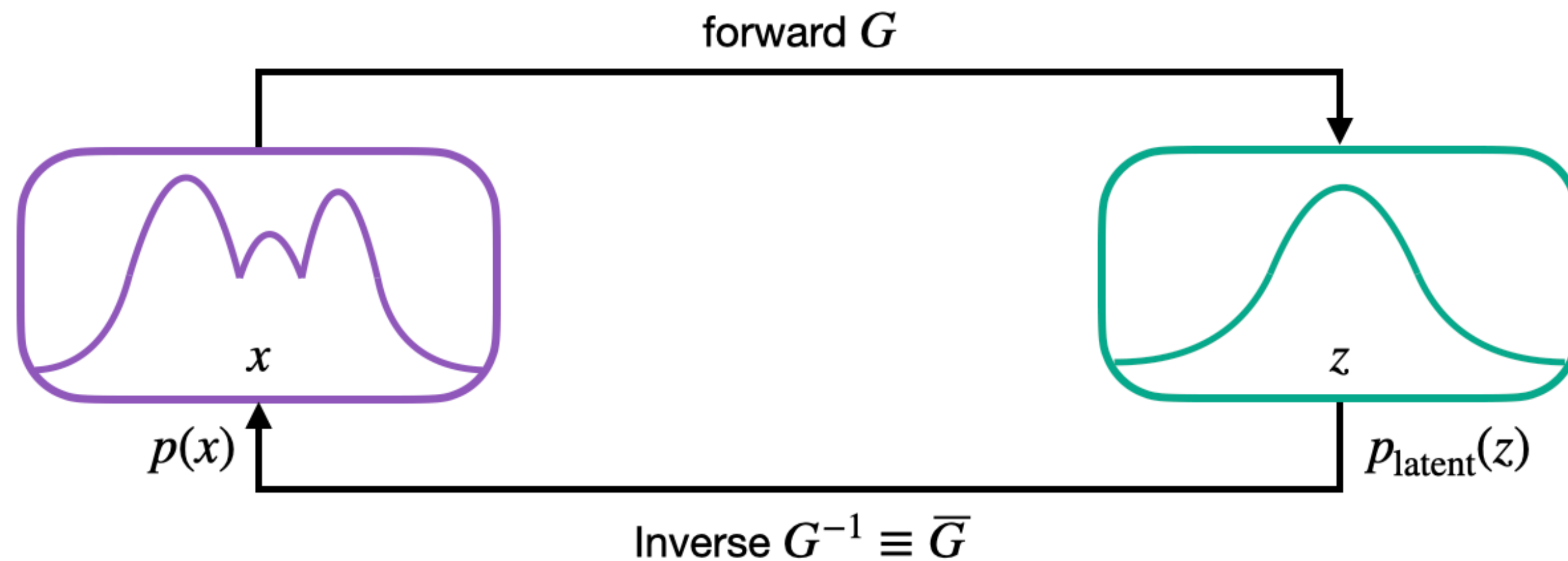
Change-of-variables formula:

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

How to train it?

Normalizing flow — Training

27



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

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

Kullback-Leibler divergence:

$$\begin{aligned} \text{KL}(p_{\text{data}}(x) | p_{\omega}(x)) &= \int dx p_{\text{data}}(x) \log \frac{p_{\text{data}}(x)}{p_{\omega}(x)} \\ &= - \int dx p_{\text{data}}(x) \log p_{\omega}(x) + \int dx p_{\text{data}}(x) \log p_{\text{data}}(x) \end{aligned}$$

No ω
dependence

Negative log-likelihood loss:

$$\mathcal{L}_{\text{NLL}} = - \int dx p_{\text{data}}(x) \log p_{\omega}(x) = \langle -\log p_{\omega}(x) \rangle_{x \sim p_{\text{data}}}$$

Tractable Jacobian?

$$\log p_{\omega}(x) = \log p_{\text{latent}}(z = G_{\omega}(x)) + \log \left| \frac{\partial G_{\omega}(x)}{\partial x} \right| \quad \rightarrow \text{Requires tractable Jacobian!}$$

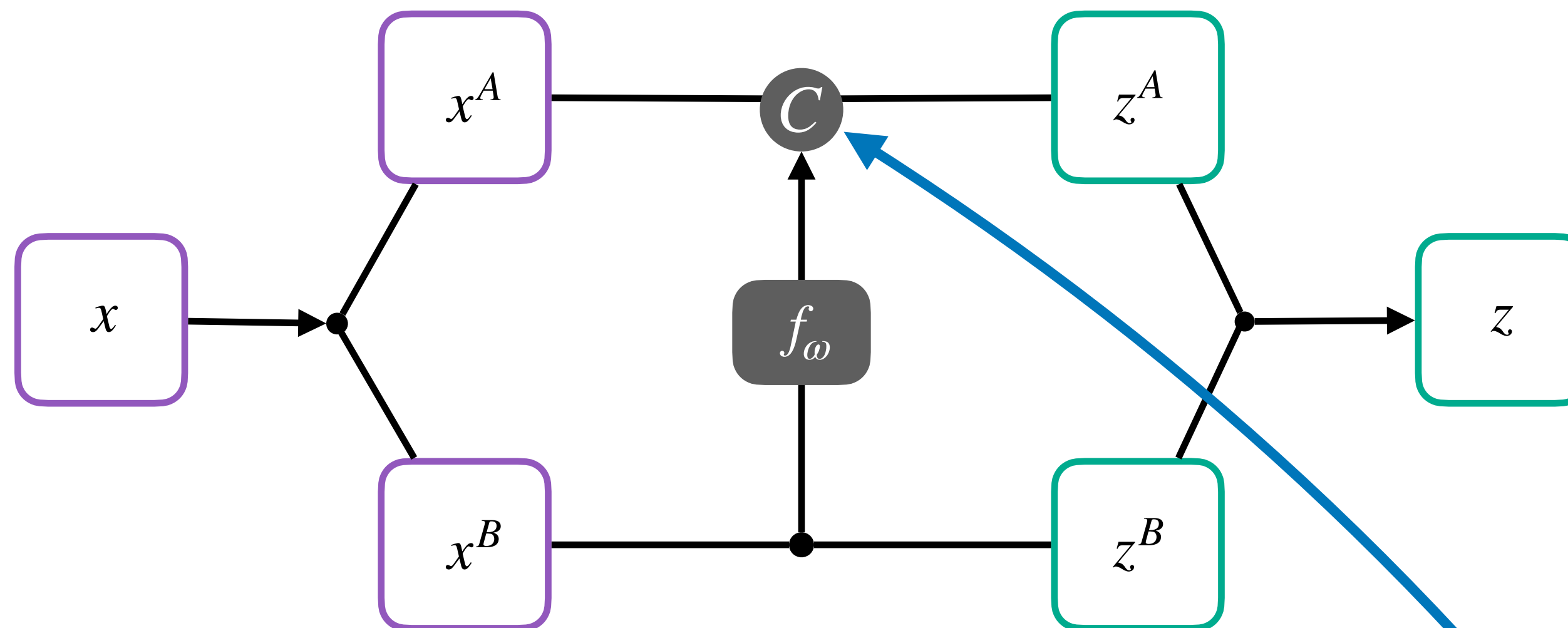
In general: $g_{\omega}(x) = \left| \frac{\partial G_{\omega}(x)}{\partial x} \right|$ is $d \times d$ matrix $\rightarrow$ Scales with $\mathcal{O}(d^3)$ 😞

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

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

$$\rightarrow J_{ij}(x) = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_d}{\partial x_1} \\ 0 & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial z_d}{\partial x_1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{\partial z_d}{\partial x_d} \end{pmatrix} \rightarrow \det J = \prod_i J_{ii} \sim \mathcal{O}(d) \quad \text{😊}$$

Coupling block



Forward pass:

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

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

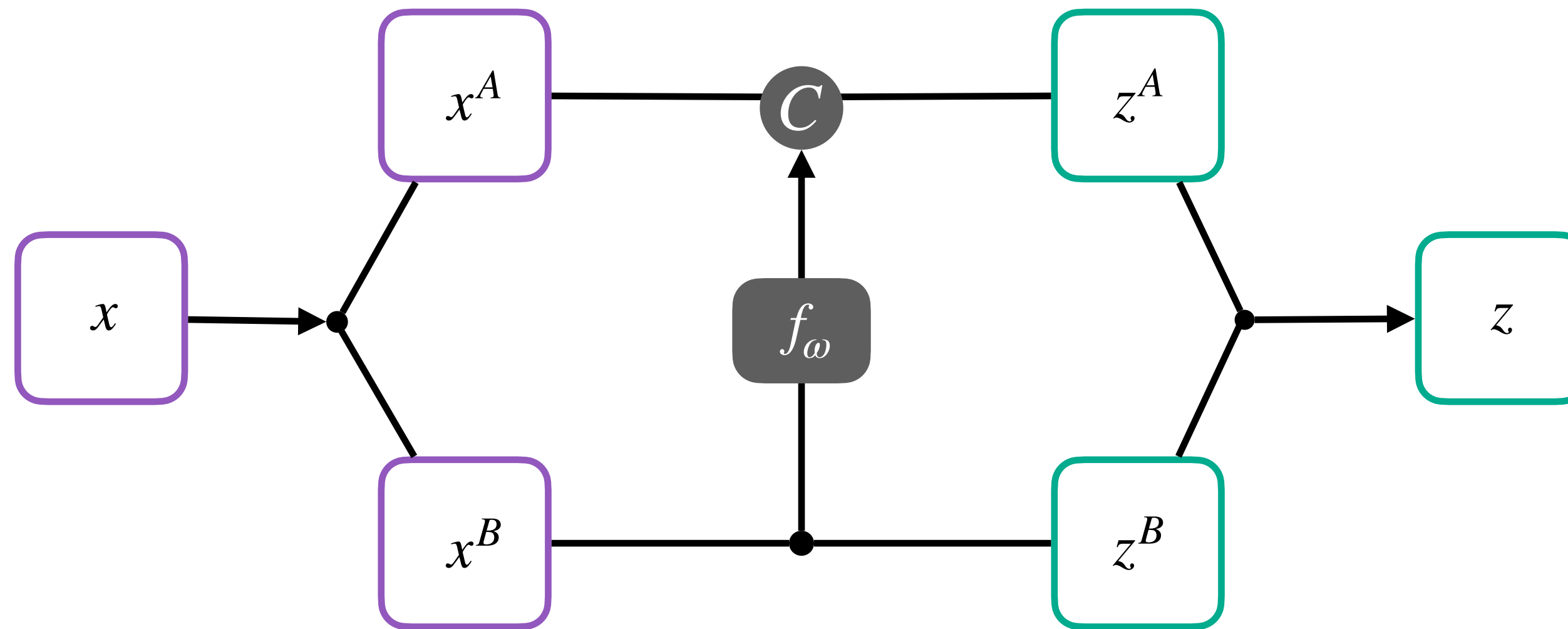
Inverse pass:

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

What is the function C ?

Coupling block

30



Forward pass:

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

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

Inverse pass:

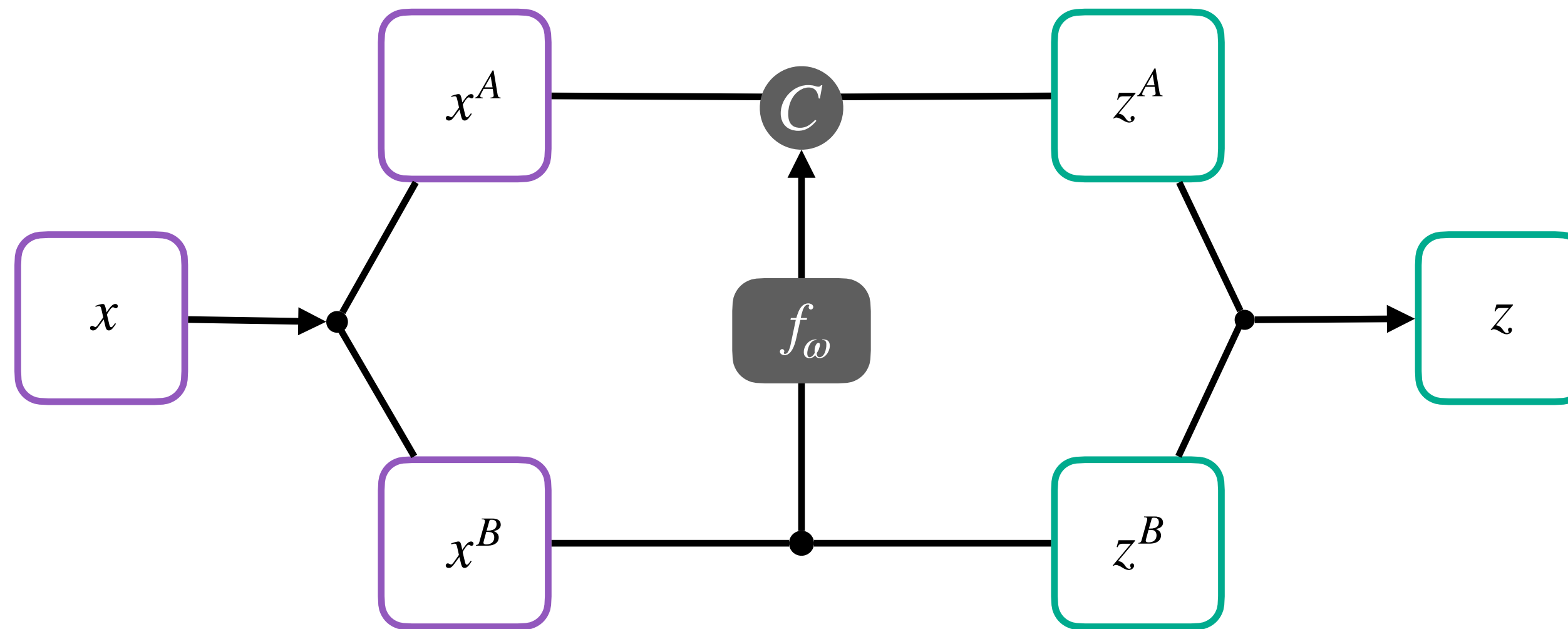
$$\begin{aligned} x^A &= C^{-1}(z^A; f_\omega(z^B)) \\ x^B &= z^B \end{aligned}$$

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

parametrized by NN

Coupling block

31



Forward pass:

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

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

Inverse pass:

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

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

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

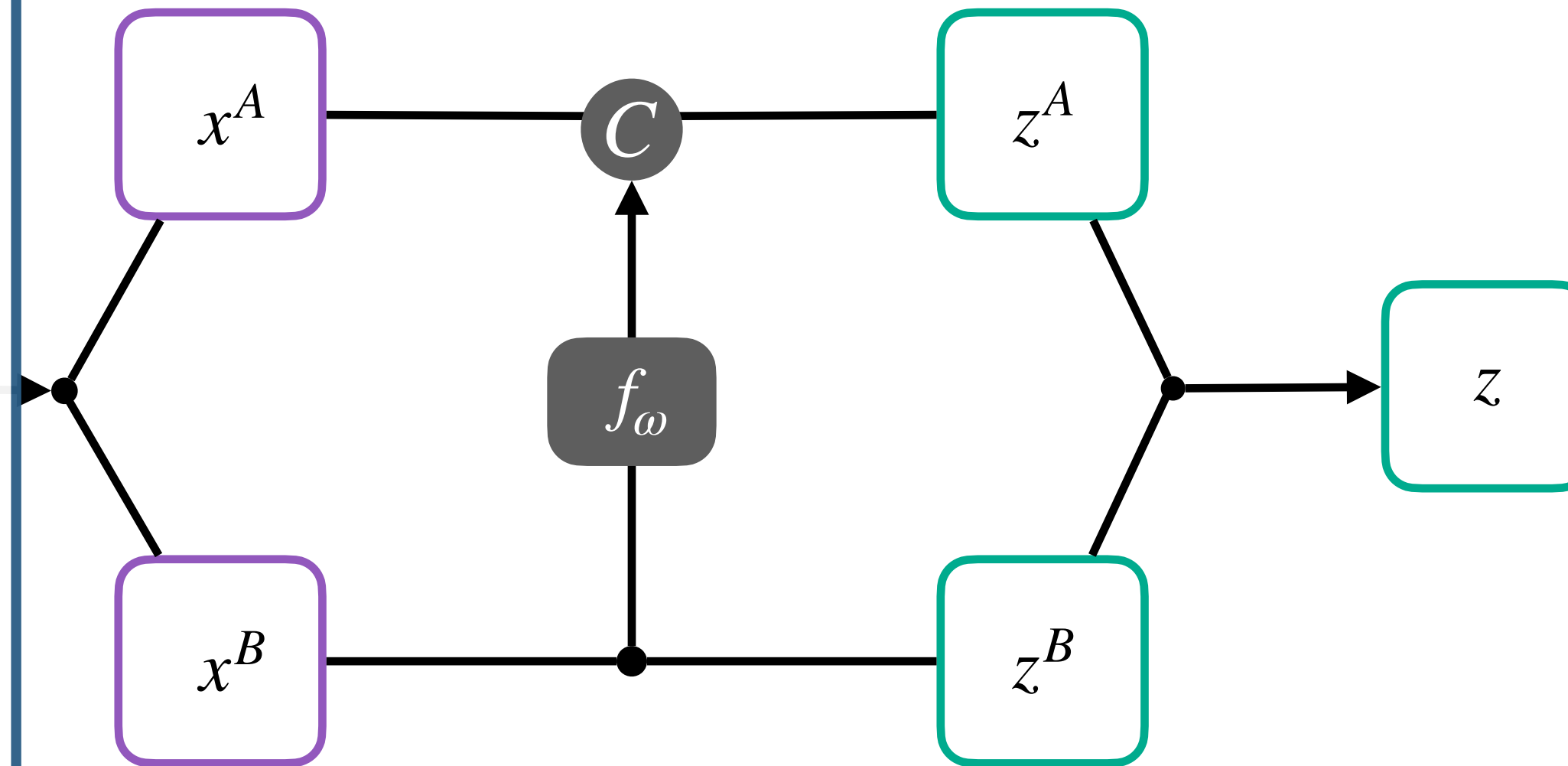
Rational quadratic
[1906.04032] $C = \frac{a_\omega x^2 + b_\omega x + c_\omega}{d_\omega x^2 + e_\omega x + f_\omega}$

Coupling block

Key facts - Normalizing Flows

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

Application
dependent



Forward pass

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

Key facts - Diffusion Model

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

Inverse pass

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

$$x^B = z^B$$

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

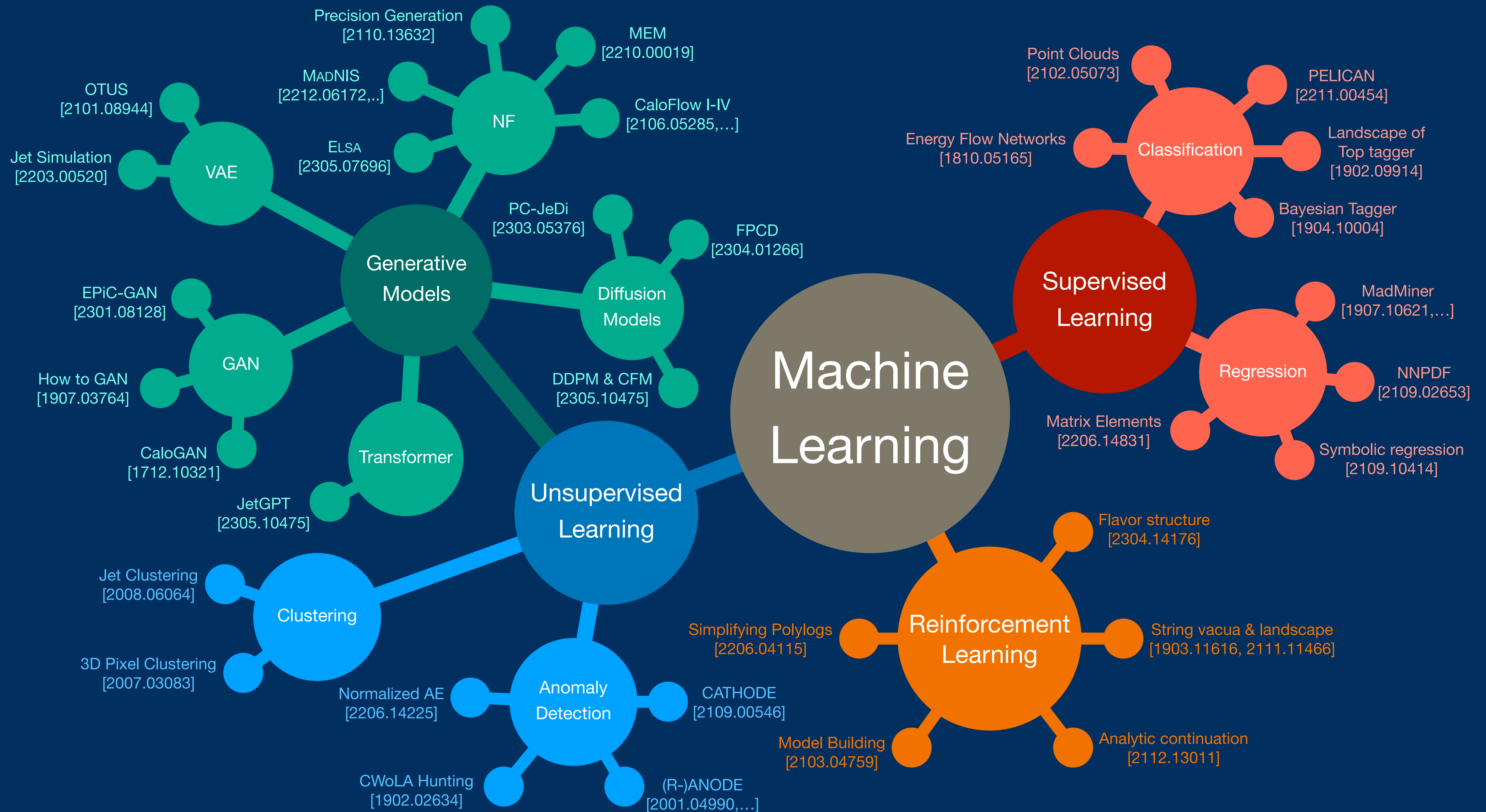
Affine
[1605.08803] $C^A = \alpha_\omega(x^B) \cdot x^A + \mu_\omega(x^B)$

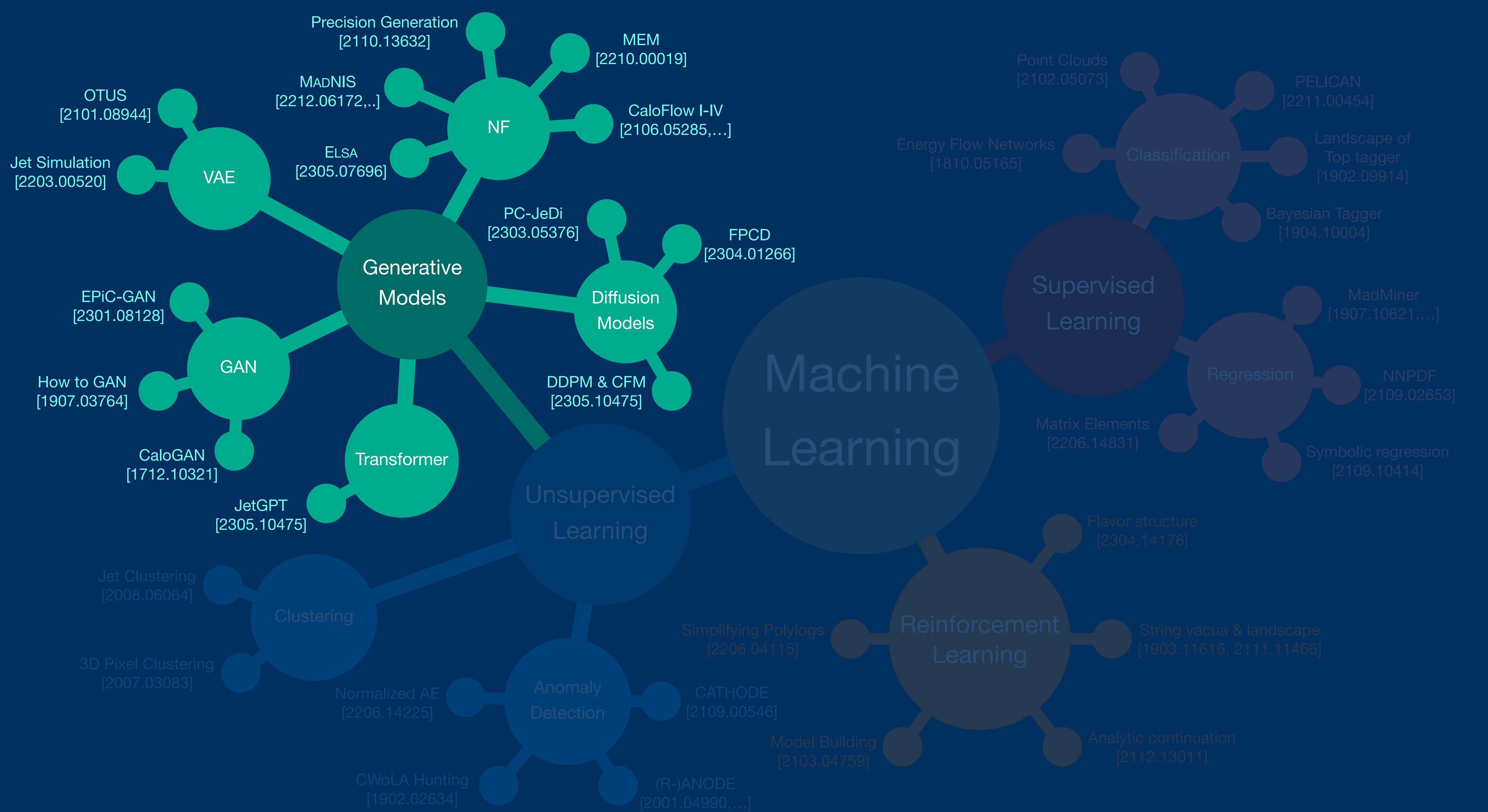
Quadratic
[1808.03856] $C = a_\omega x^2 + b_\omega x + c_\omega$

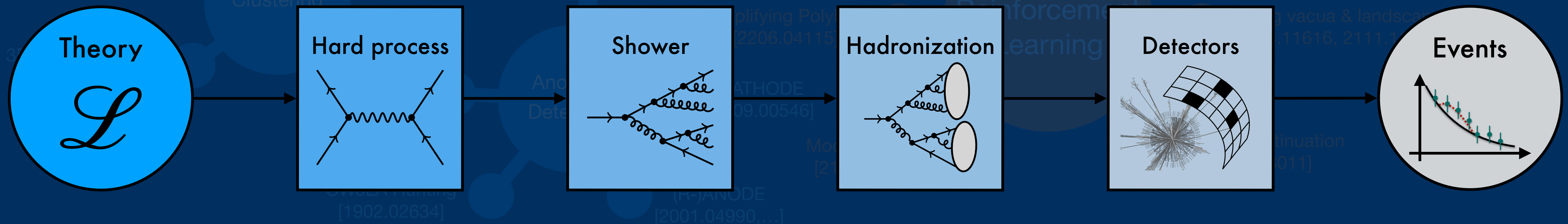
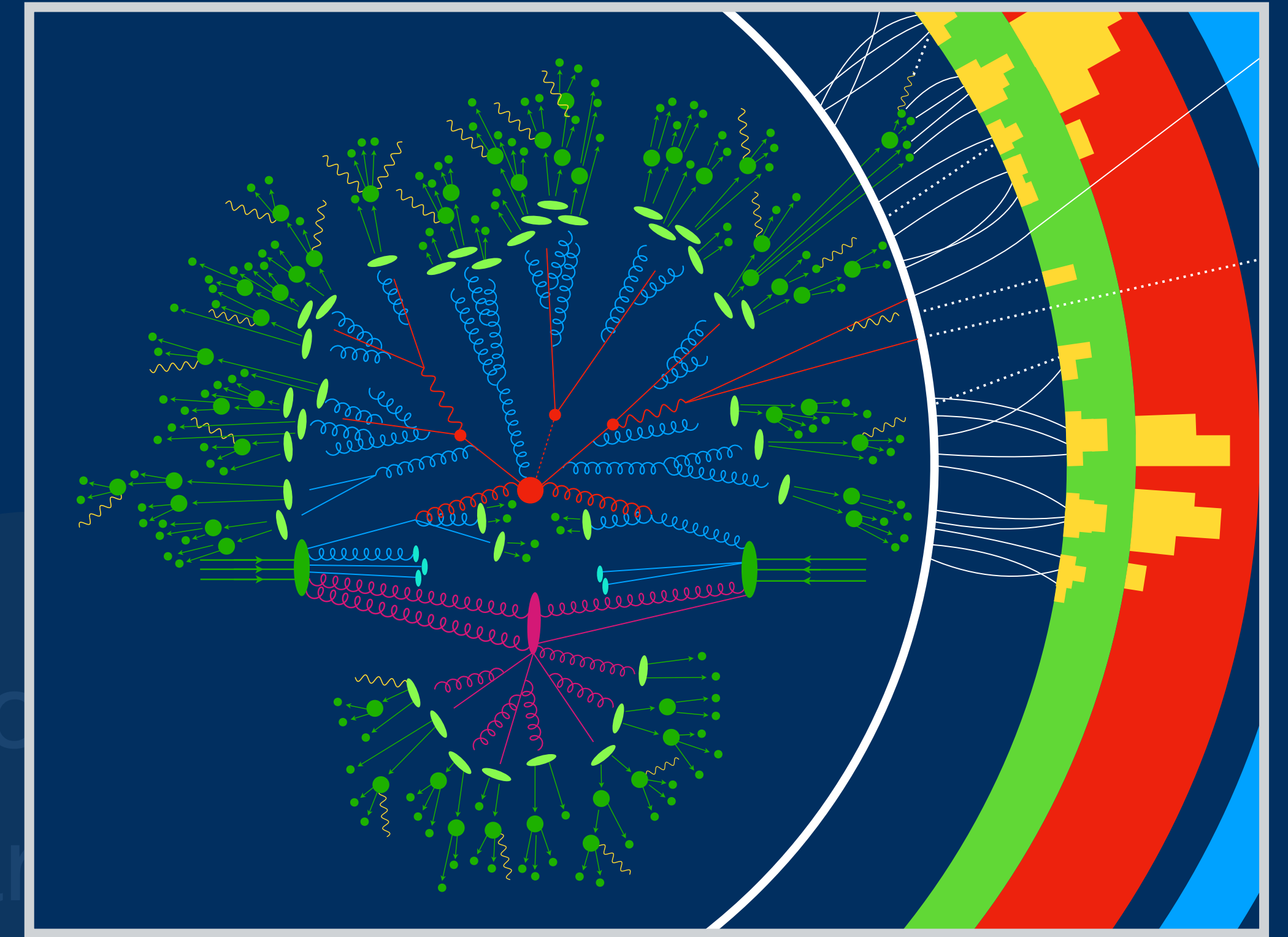
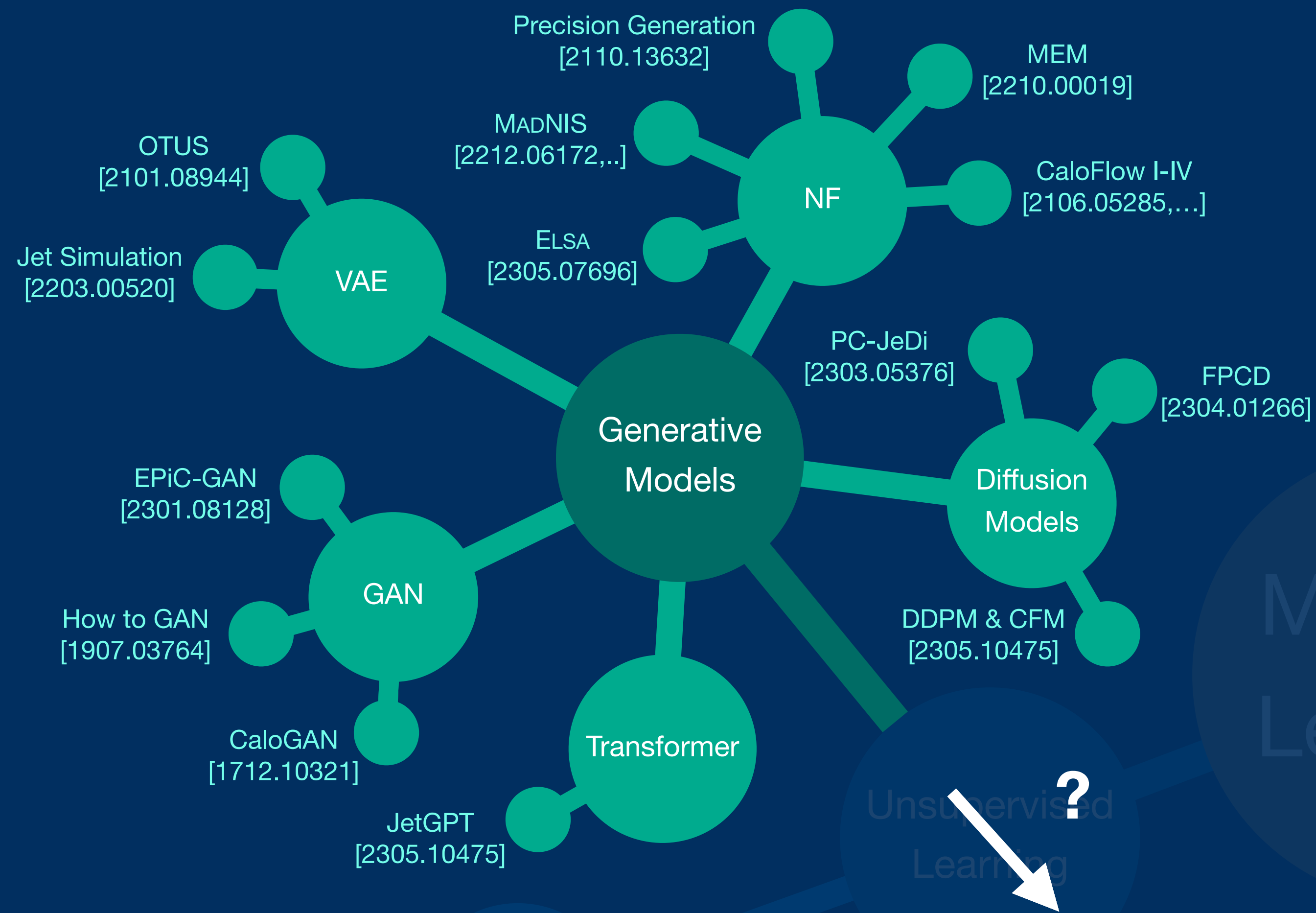
Rational quadratic
[1906.04032] $C = \frac{a_\omega x^2 + b_\omega x + c_\omega}{d_\omega x^2 + e_\omega x + f_\omega}$



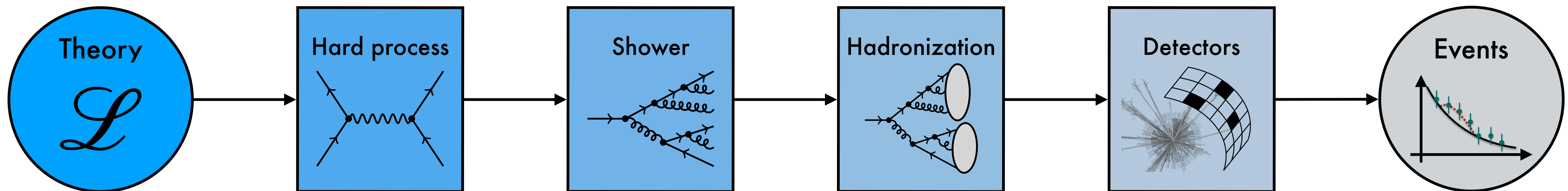
Questions?



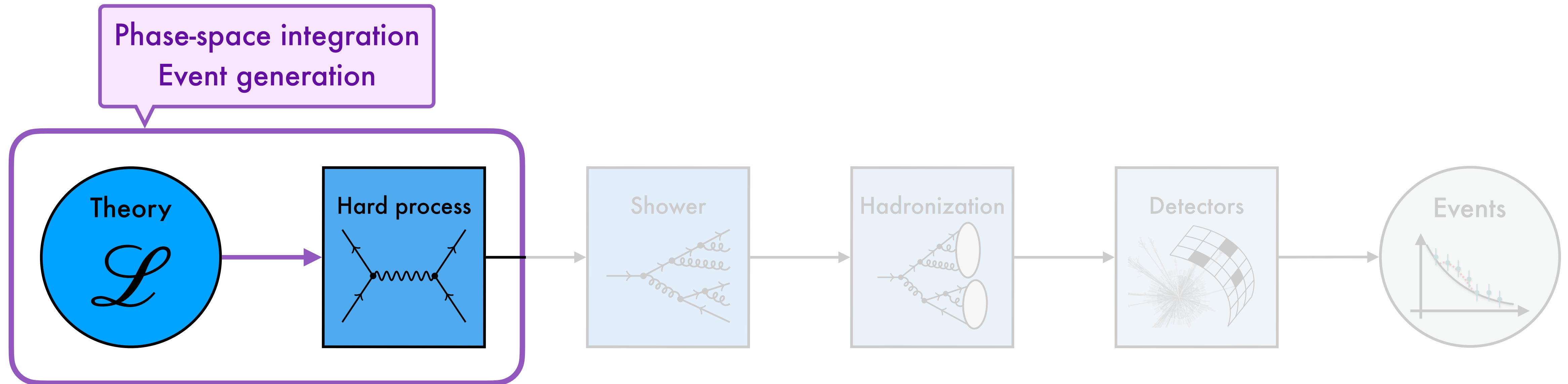




The LHC simulation chain



The LHC simulation chain + ML



Example I

Neural importance sampling with MadNIS

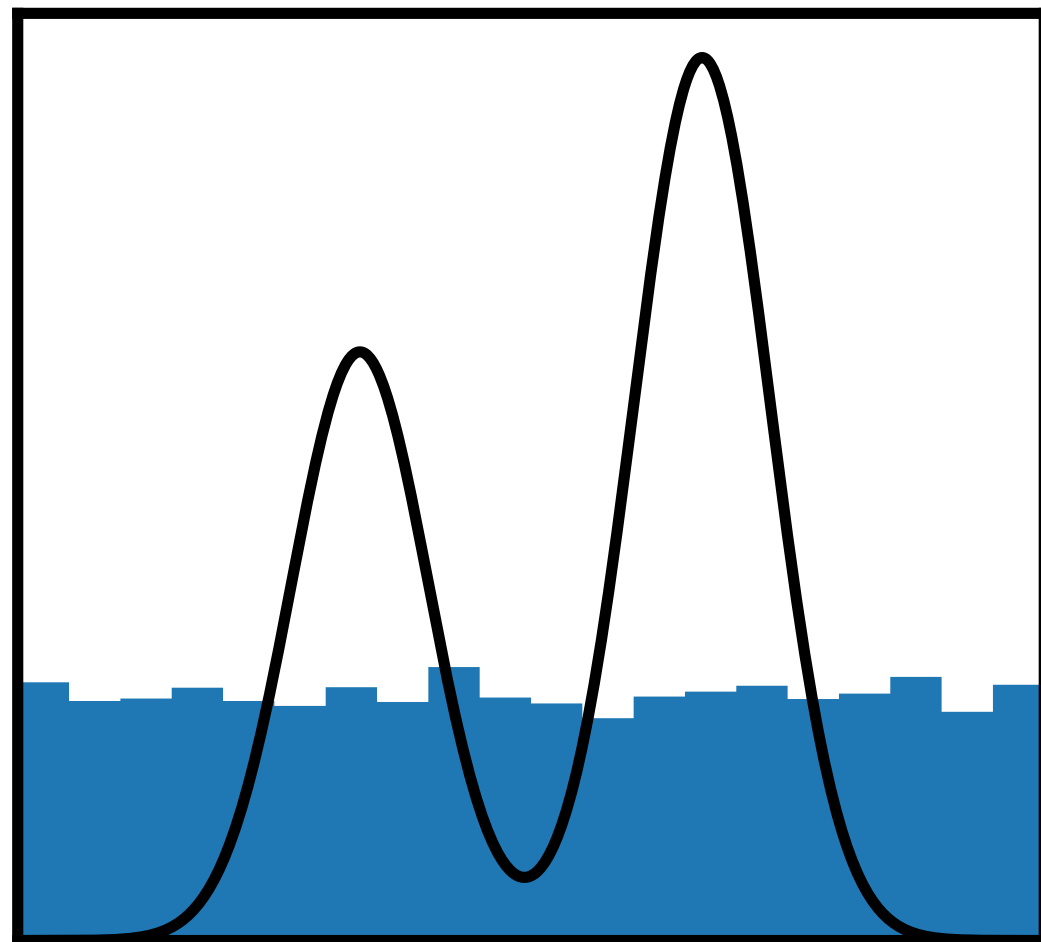
Heimel, Huetsch, Maltoni, Mattelaer, Plehn, RW [[2311.01548](#)]

Heimel, RW, Butter, Isaacson, Krause, Maltoni, Mattelaer, Plehn [[2212.06172](#)]

Monte Carlo integration

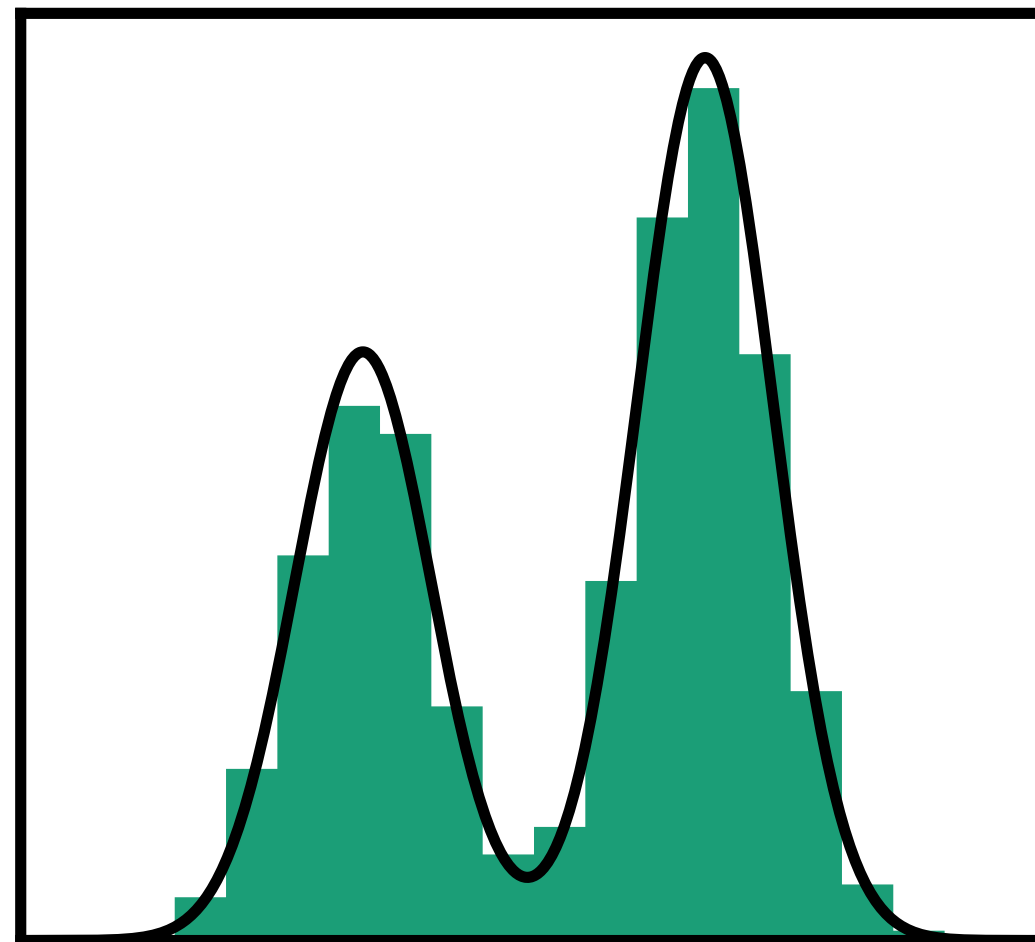
39

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



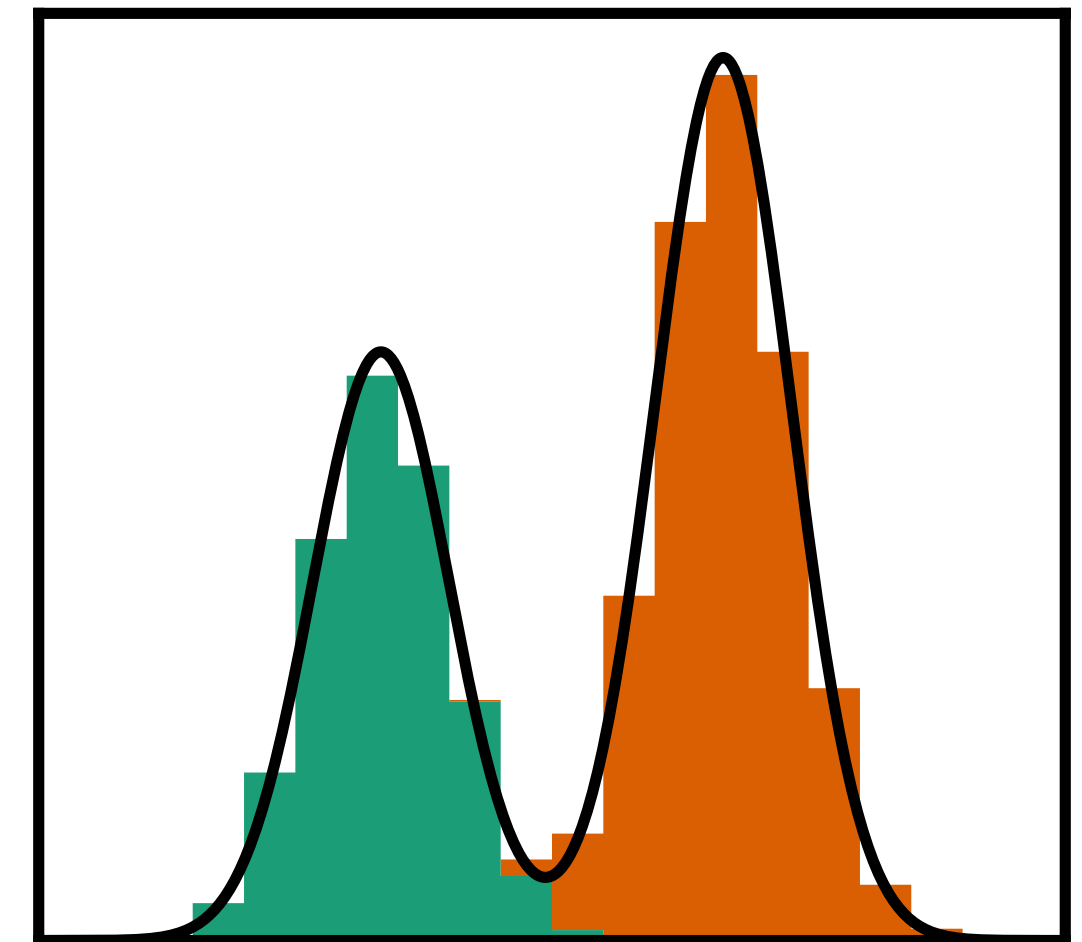
Flat sampling:
inefficient

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



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 in MadGraph



Sum over channels

MadGraph: build channels
from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

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

Channel weights

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

Channel mappings

MadGraph: use amplitude structure,...
Analytic mappings + refine with *Vegas*

Importance sampling — VEGAS

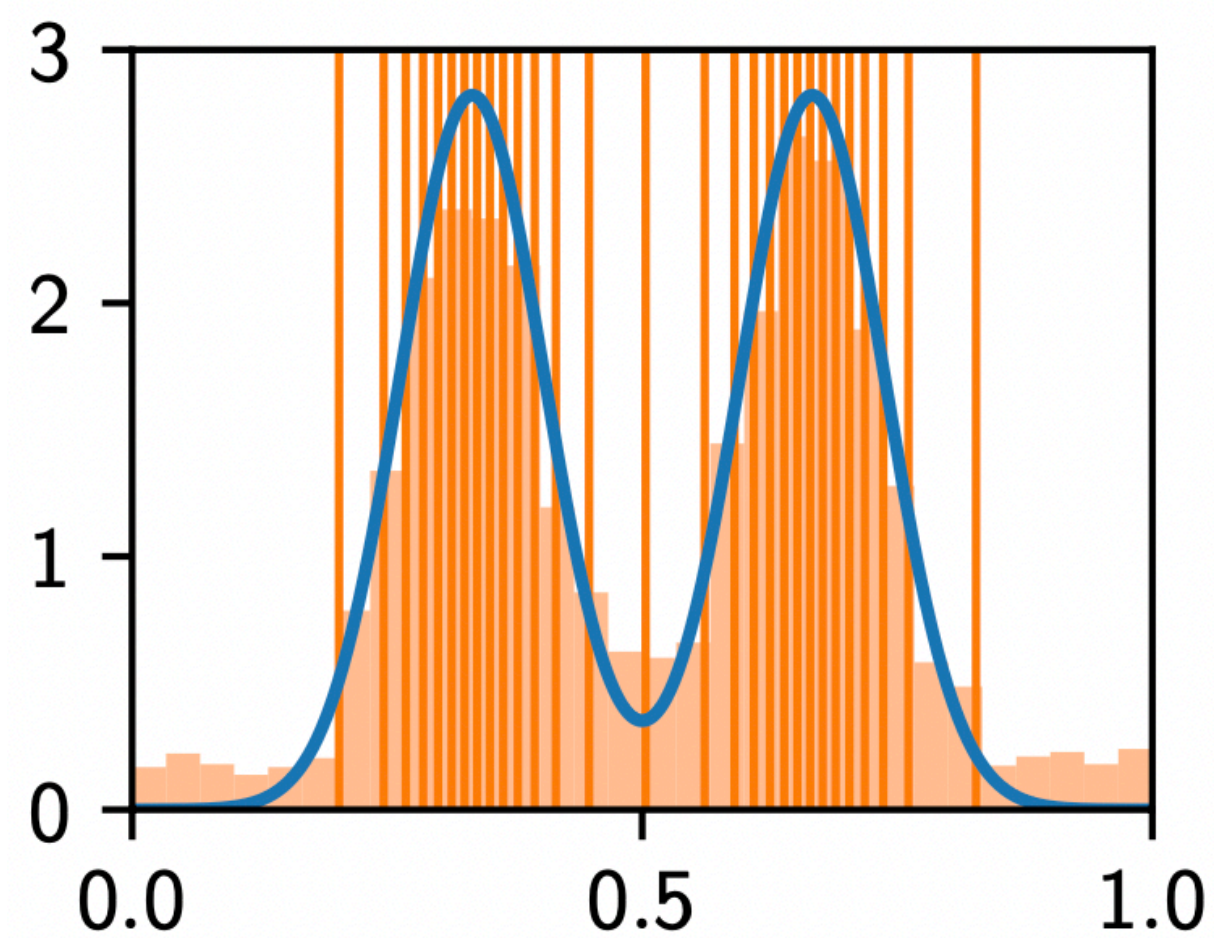
41

Factorize probability

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



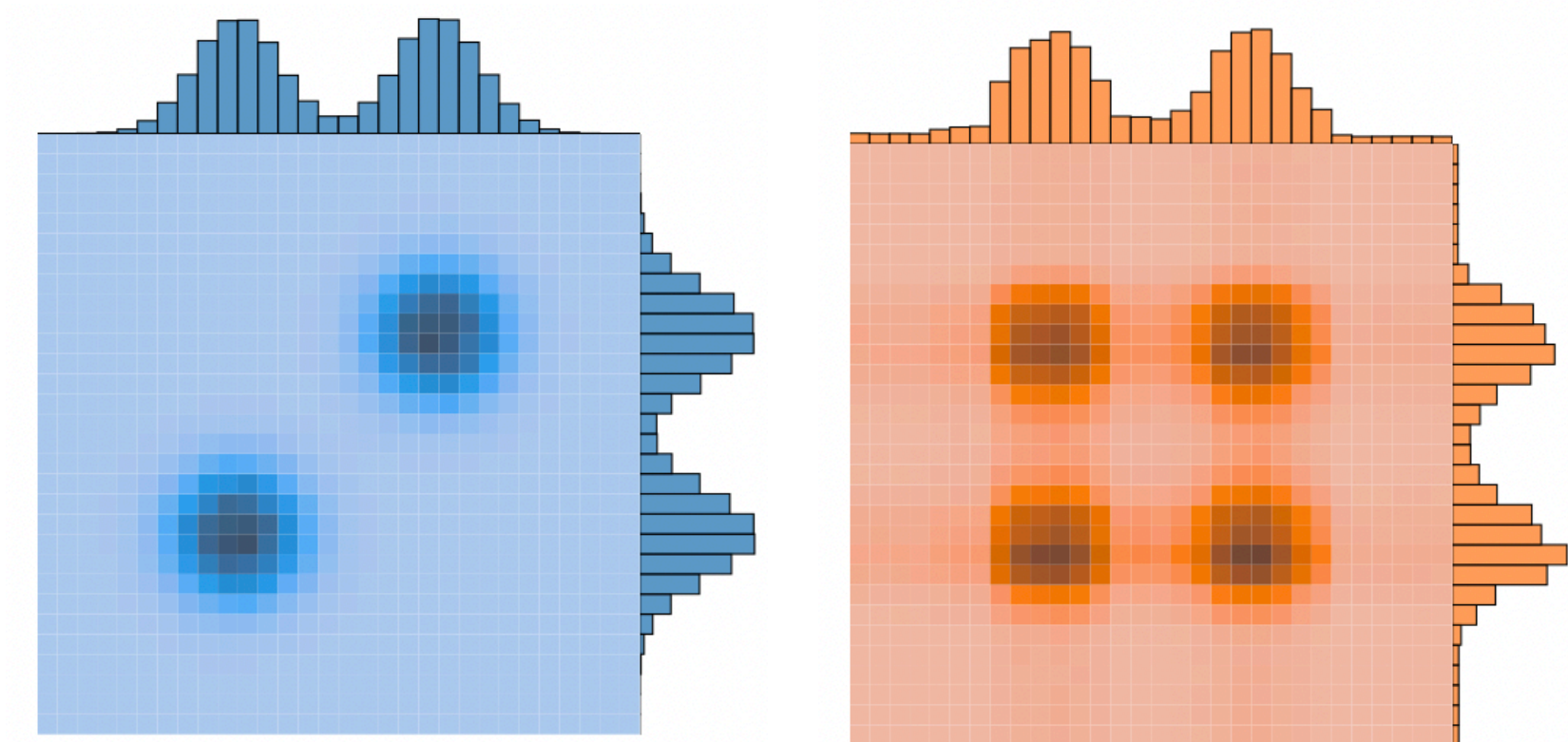
Fit bins with equal probability
and varying width



[G. P. Lepage, 1978]



- ⊕ Computationally cheap
- ⊖ High-dim and rich peaking functions
→ **slow convergence**
- ⊖ Peaks not aligned with grid axes
→ **phantom peaks**



Event generation in MadGraph



Sum over channels

MadGraph: build channels
from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

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

Channel weights

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

Channel mappings

MadGraph: use amplitude structure,...
Analytic mappings + refine with *Vegas*

Event generation in MadGraph + MadNIS



Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{g_i^\phi(x)} \right\rangle$$

$x \sim g_i^\phi(x)$

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 *Normalizing Flow*

Event generation in MadGraph + MadNIS



Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

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

Learned channel weights

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

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

Learned channel mappings

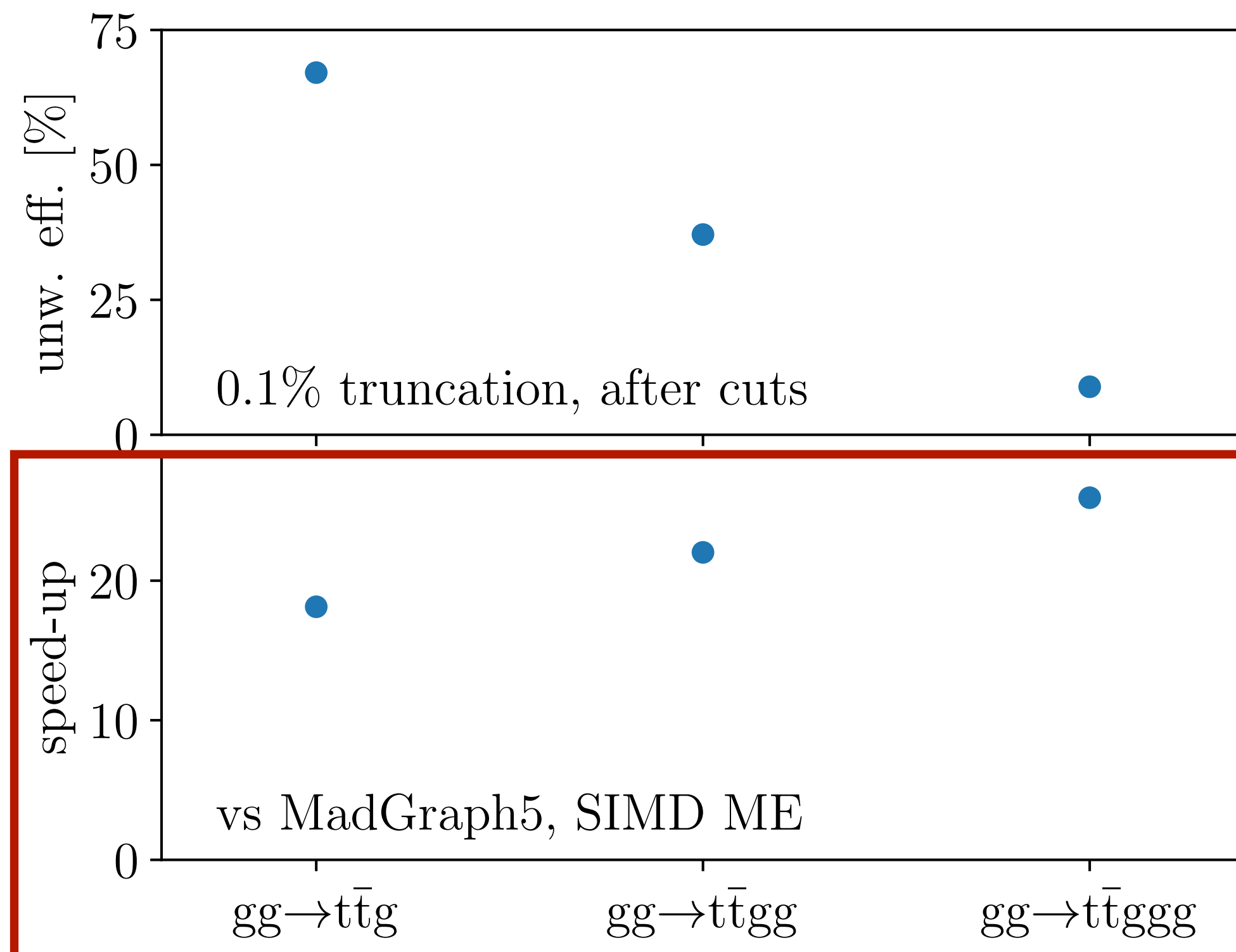
MadGraph: use amplitude structure, ...
Analytic mappings + ~~refine with Vegas~~

parametrize with *neural network*

refine with *Normalizing Flow*

MadgGraph + MadNIS at LO

Unweighting efficiency

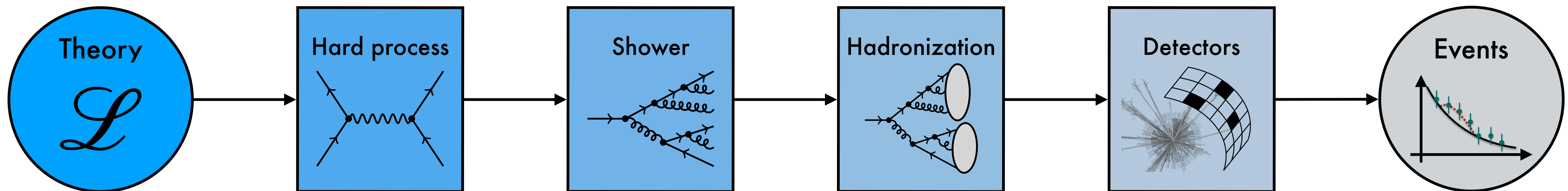


- improved unweighting efficiency
→ **large speed-up across processes**

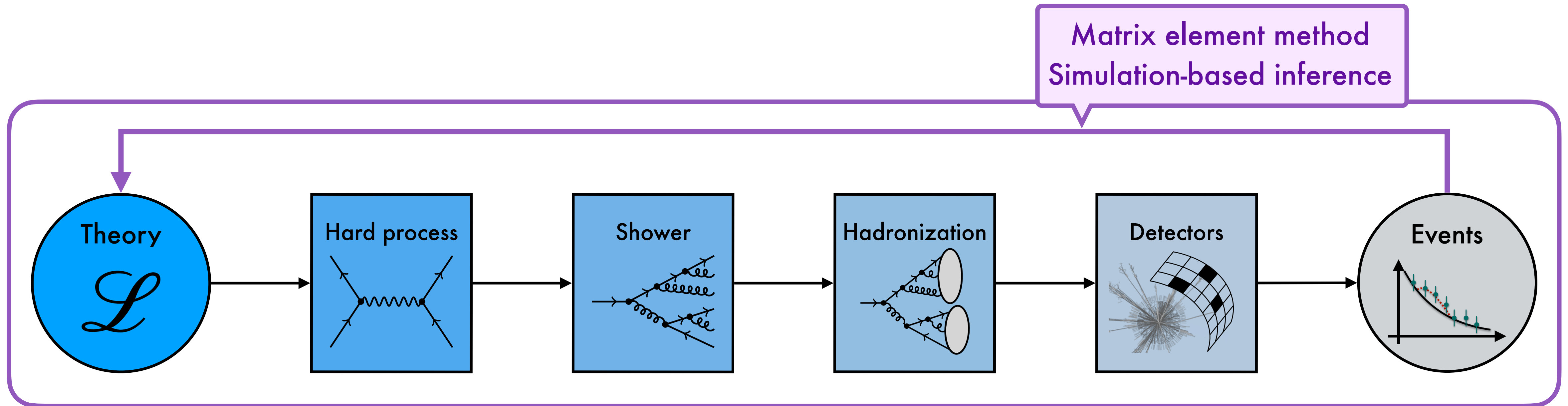
- will become default in many cases
→ starting at $2 \rightarrow 3$ or $2 \rightarrow 4$

(preliminary)

The LHC simulation chain



The LHC simulation chain + ML

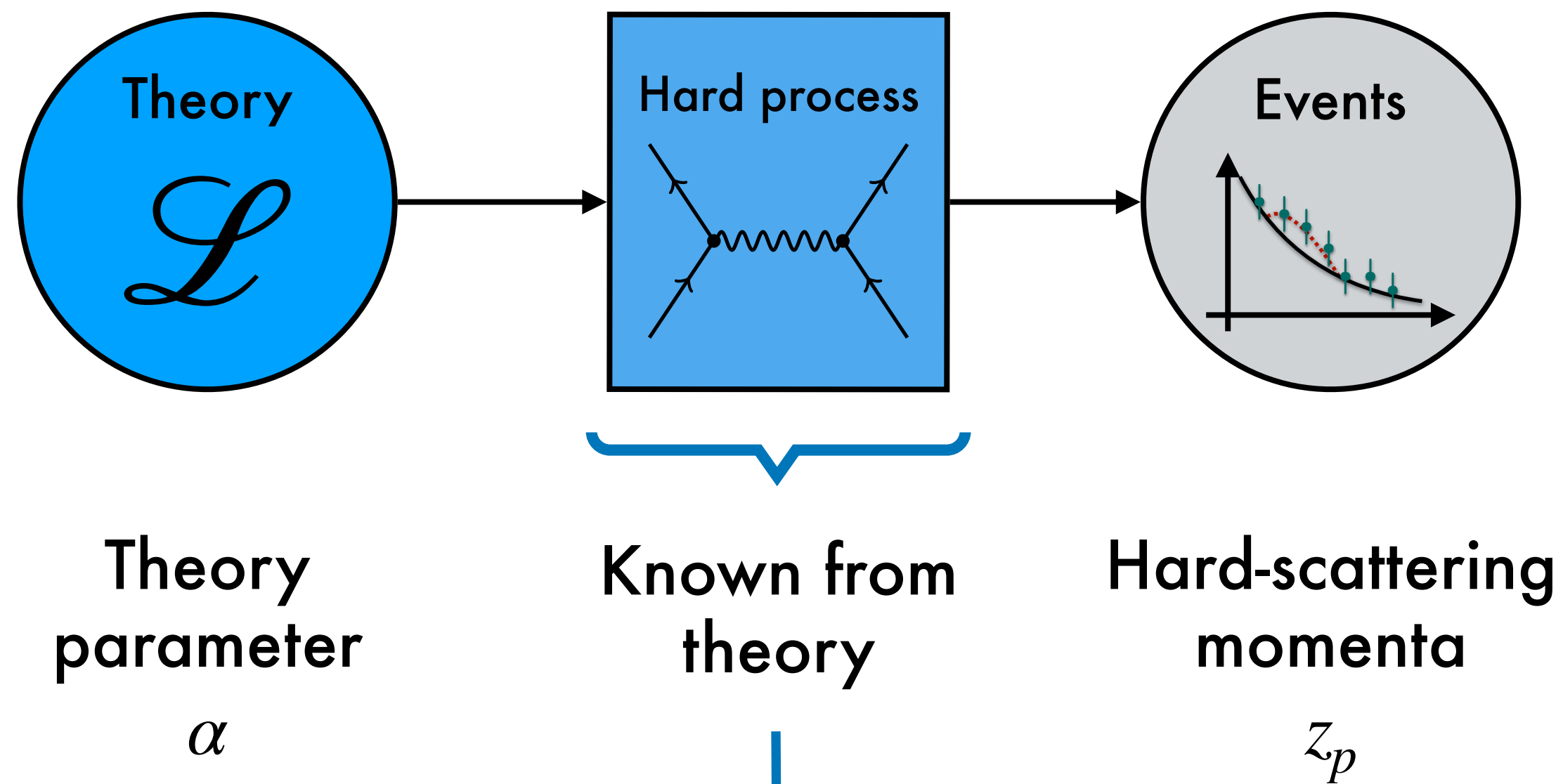


Example II

Matrix Element Method

Matrix Element Method — Theory

49



Likelihood from differential cross section

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

Classical analysis

- ⊖ hand-crafted observables
- ⊖ binned data
- not all information is used 😞

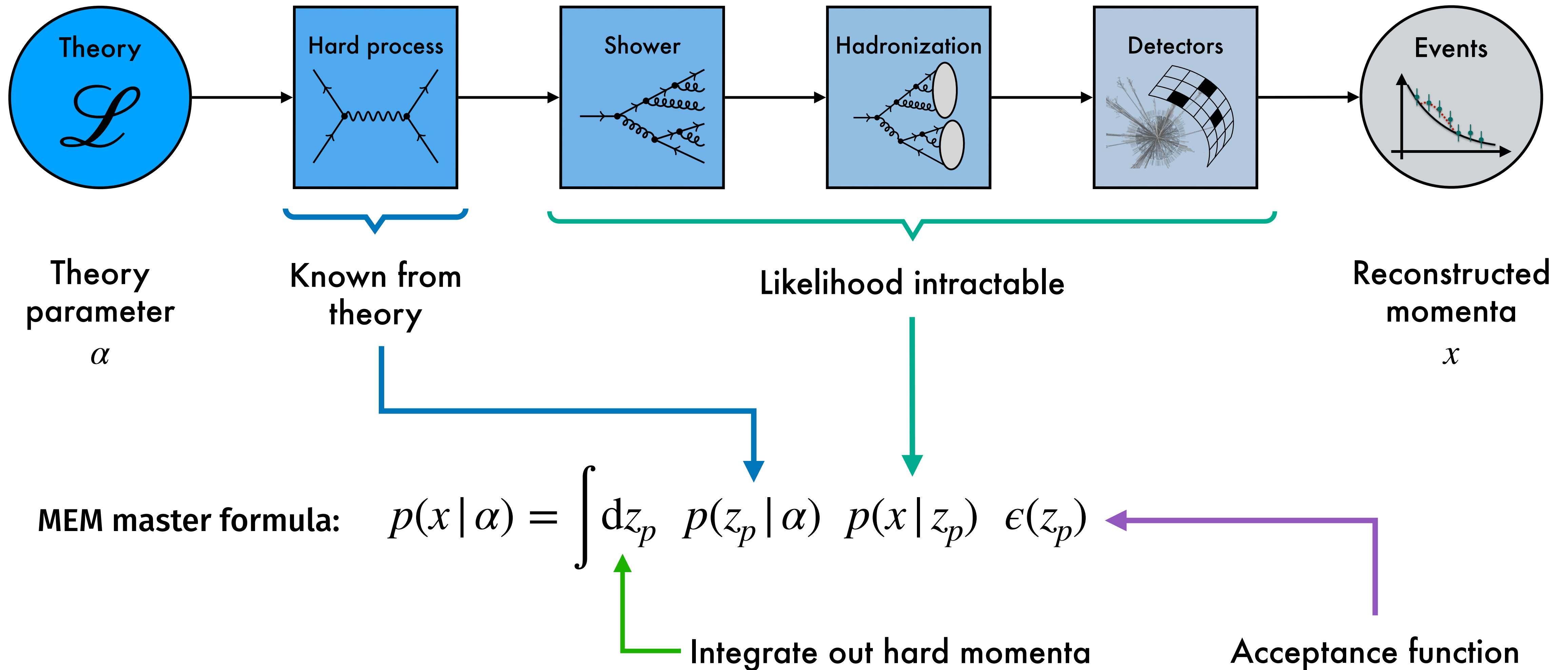


Matrix Element Method (MEM)

- ⊕ based on first principles
- ⊕ estimates uncertainties reliably
- ⊕ optimal use of information
- perfect for processes with few events 😊

Matrix Element Method — Reality

50



Matrix Element Method + ML

51

$$p(x | \alpha) = \int dz_p \quad p(z_p | \alpha) \quad p(x | z_p) \quad \epsilon(z_p)$$

Efficient MC integration

importance sampling:
Normalizing Flow

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

Theory knowledge

differential
cross-section

$$\frac{1}{\sigma(\alpha)} \frac{d\sigma(\alpha)}{dz_p}$$

Transfer function

density estimation:
Normalizing Flow
solve combinatorics:
Transformer

Transformer

Acceptance function

learn with simple
Classifier network

Single Higgs production

52

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



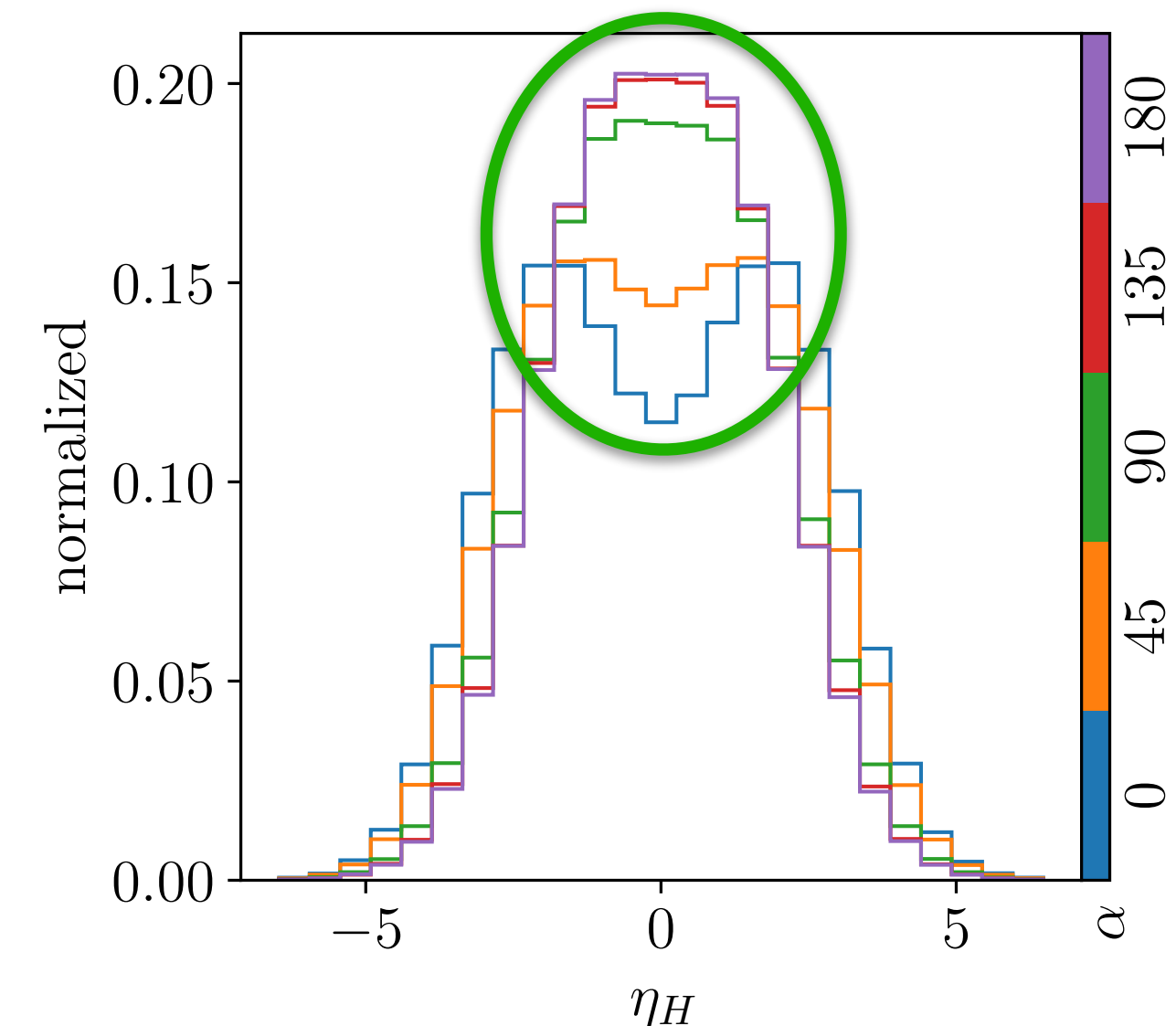
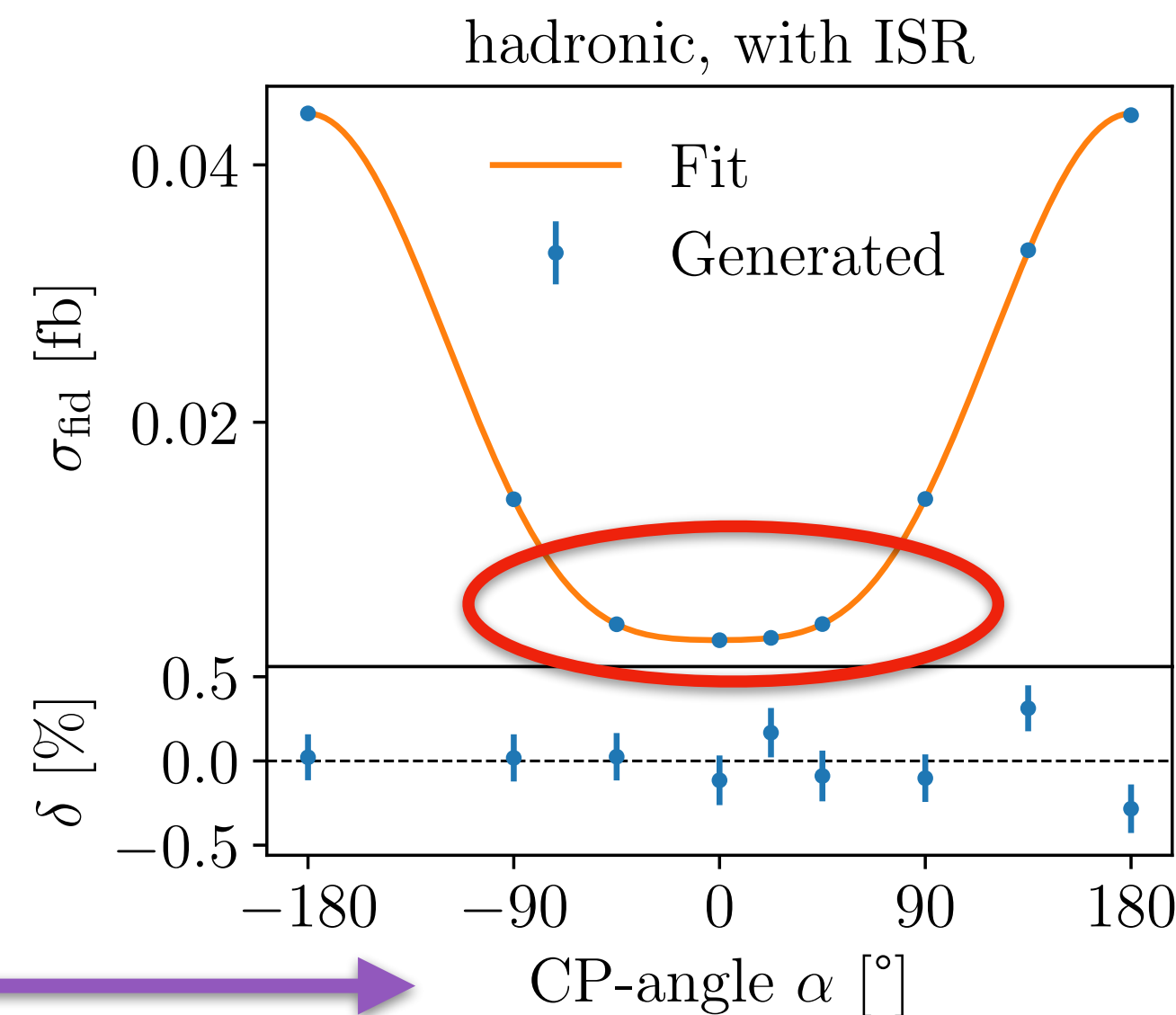
Hadronic decay of top + ISR

$tHq \rightarrow (bjj) (\gamma\gamma) j + \text{QCD jets}$



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

Anomalous coupling with **CP-angle** α



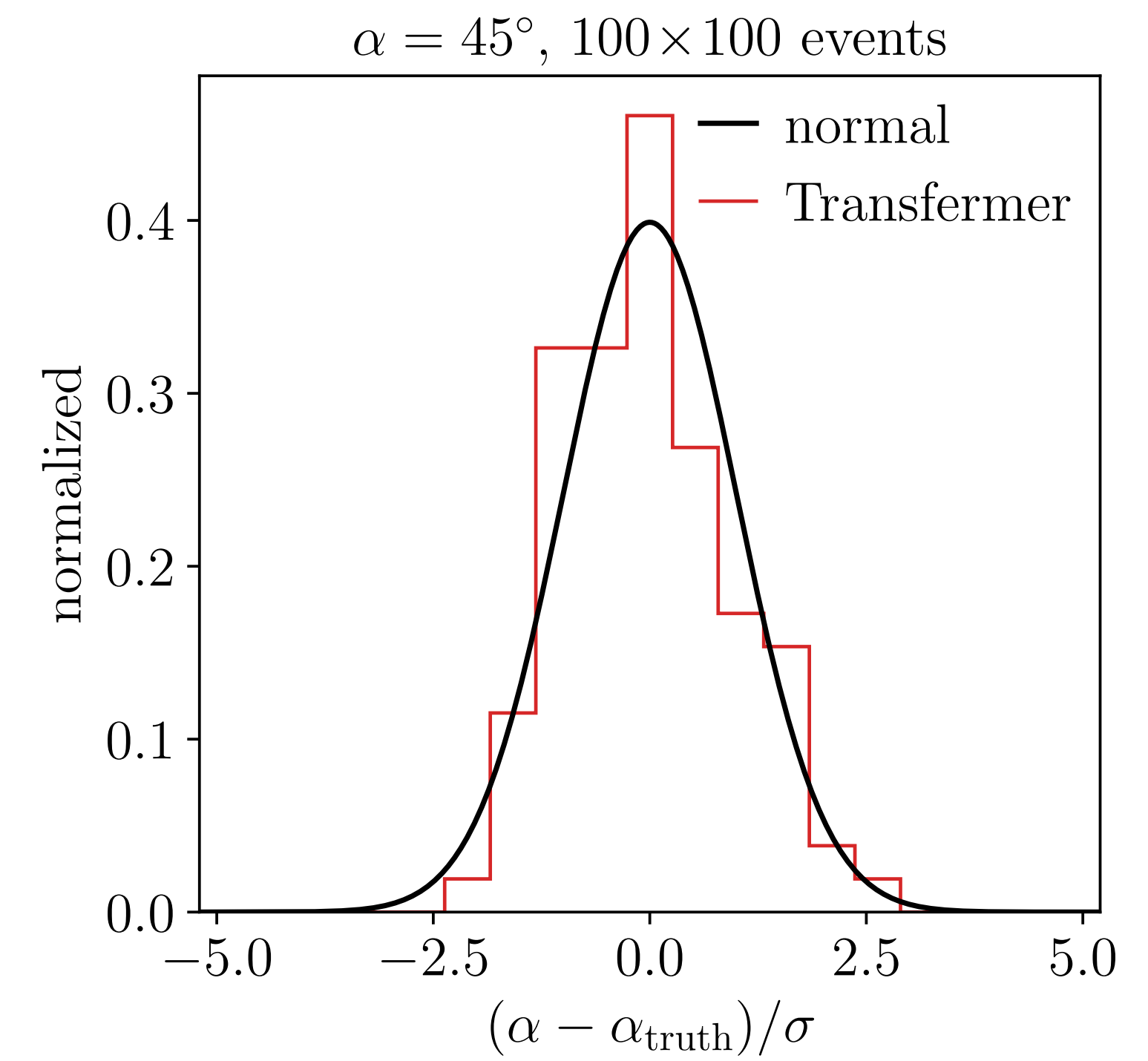
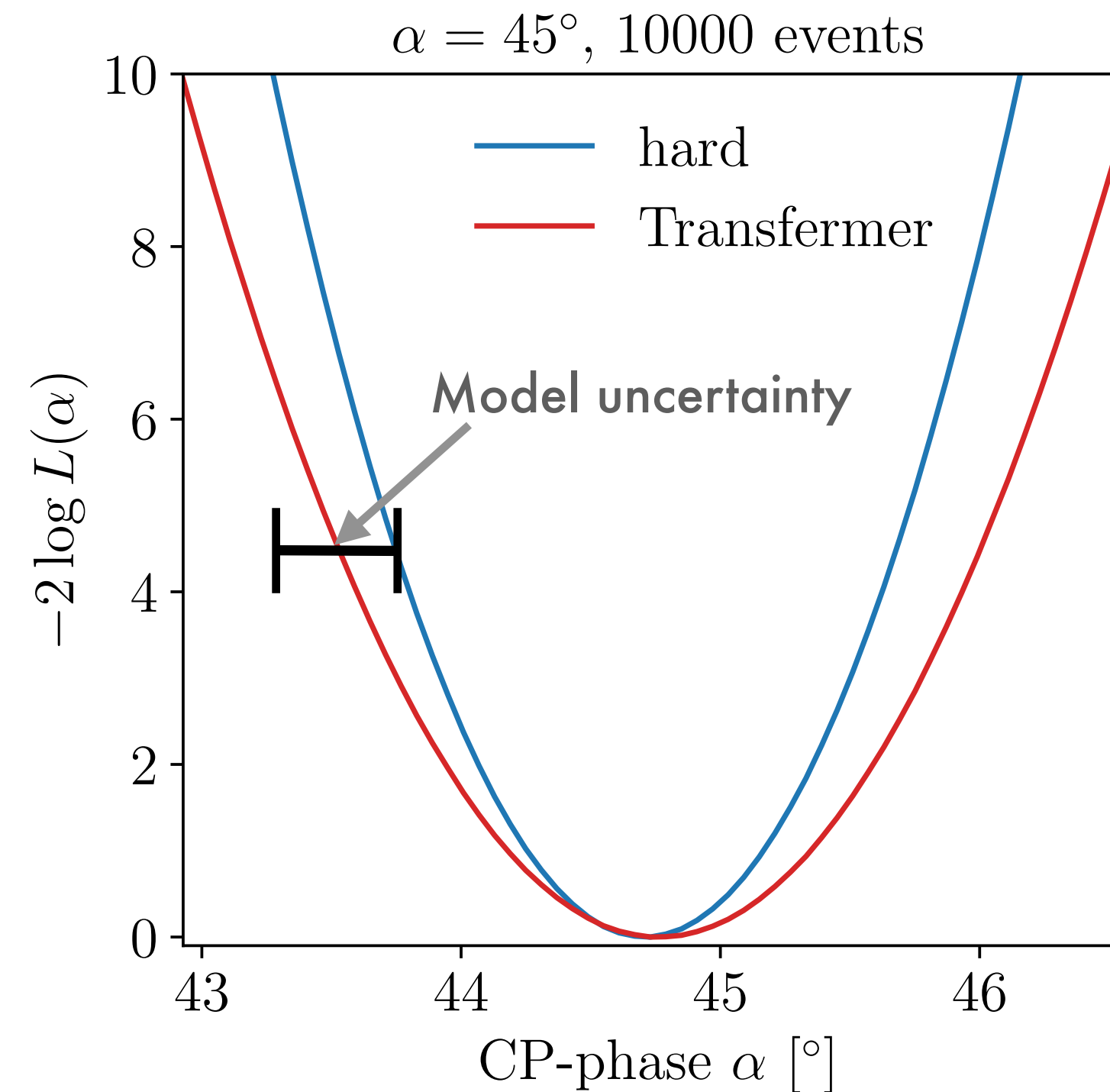
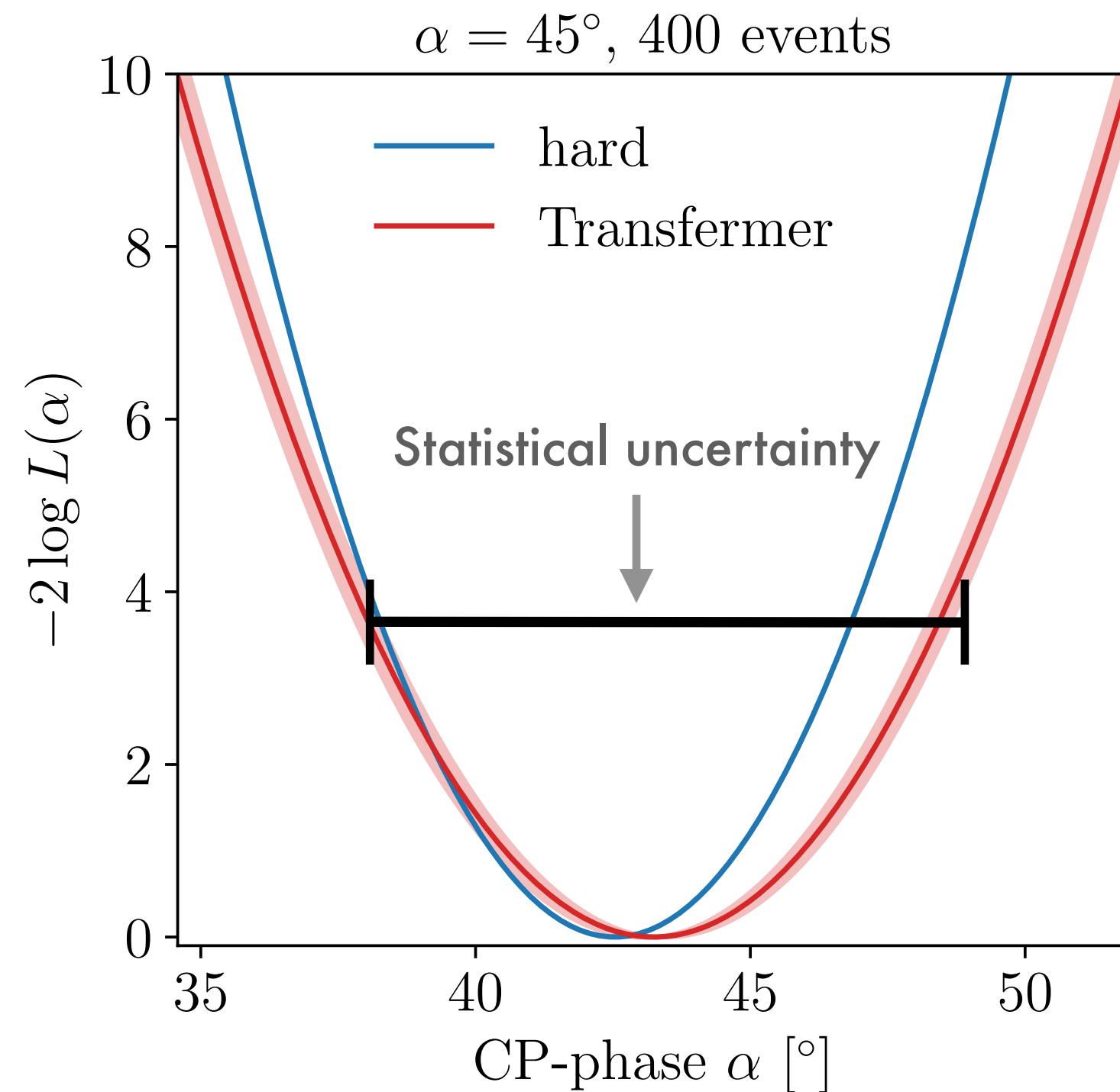
- ⊖ low total cross section
- ⊖ low variation of rate

⊕ kinematics sensitive

→ ideal use case for **MEM**

Inference results

53



- Uncertainty bands: **MC integration error** & **model uncertainty from limited training statistics**
- **well-calibrated likelihoods**, both for low and high event count