UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

ML 4 Jets 2023

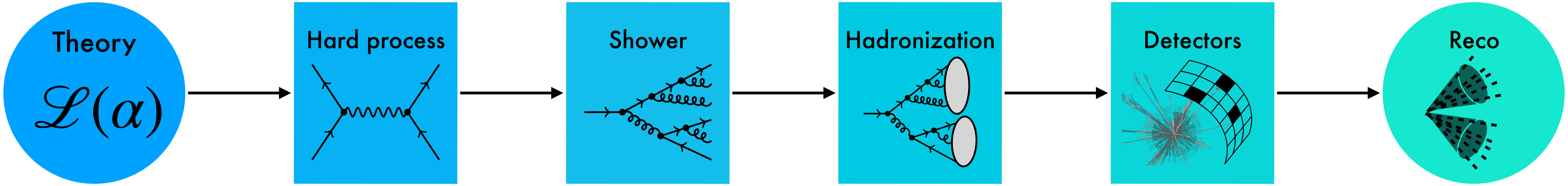
Precision-Machine Learning for the Matrix Element Method

T. Heimel, **N. Huetsch**, R. Winterhalder, T. Plehn, A. Butter
arXiv: 2310.07752

SPONSORED BY THE
Federal Ministry of Education and Research

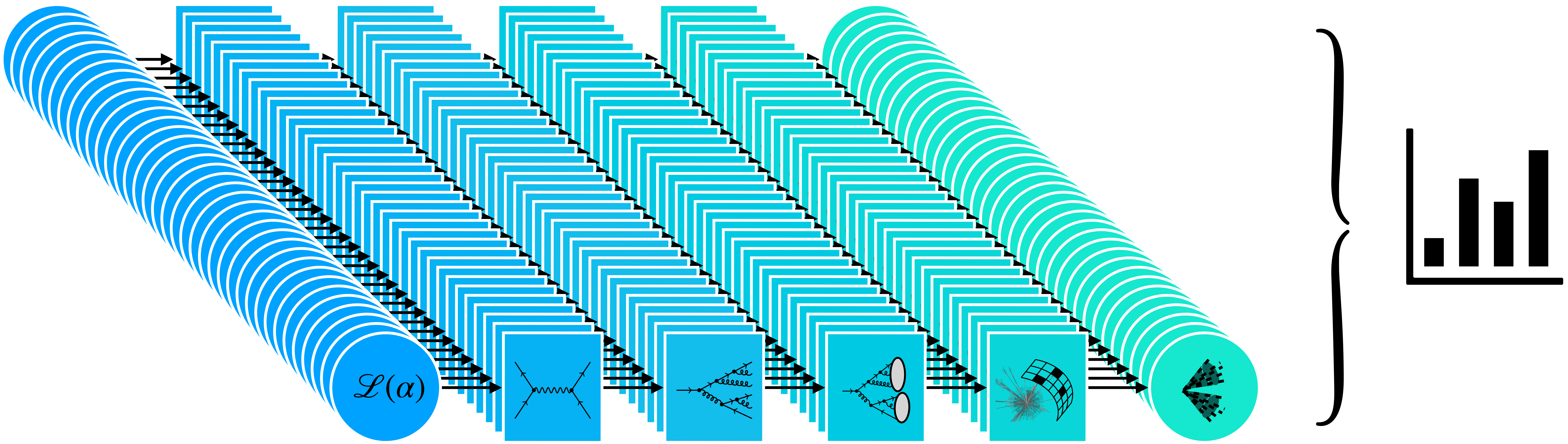
Building on: A. Butter, T. Heimel, T. Martini, S. Peitzsch, T. Plehn: arXiv:2210.00019

From Theory to Experiment in LHC Physics



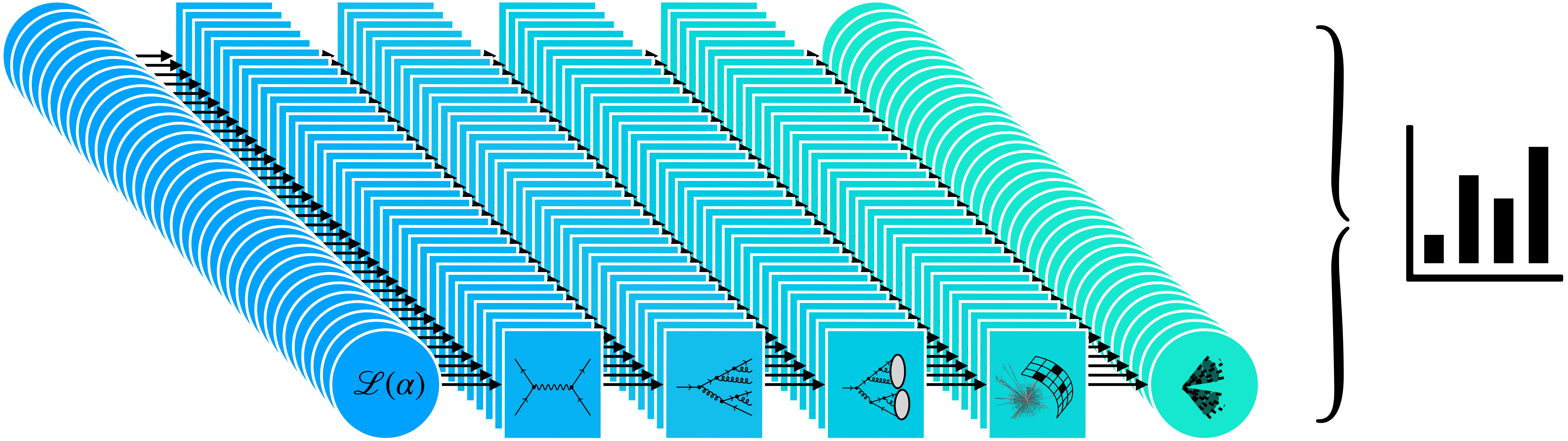
Each event undergoes reconstruction

From Theory to Experiment in LHC Physics



Event samples are combined into observable histogram

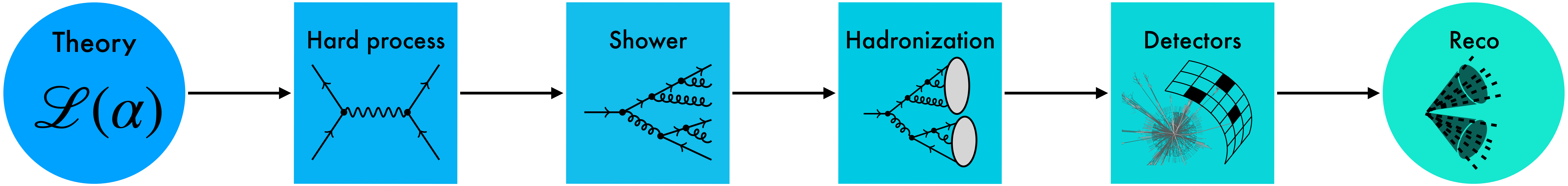
From Theory to Experiment in LHC Physics



We lose a lot of information doing this!

What if we could go for the likelihood instead?

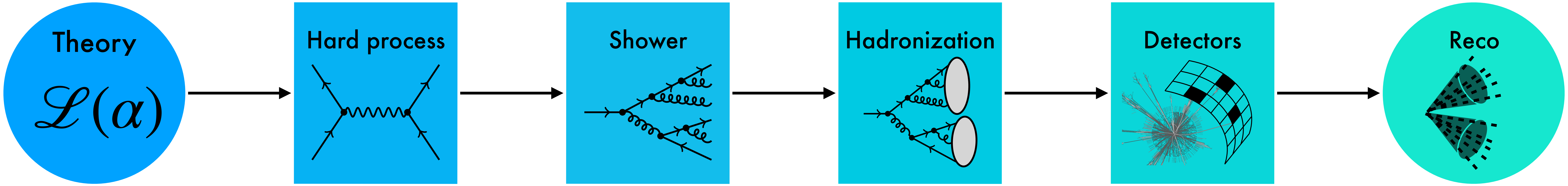
Going for the likelihood



$$p(x_{hard} | \alpha)$$

Hard-scattering level likelihood known via differential cross-section $p(x_{hard} | \alpha) = \dfrac{1}{\sigma(\alpha)} \dfrac{d\sigma(\alpha)}{dx_{hard}}$

Going for the likelihood

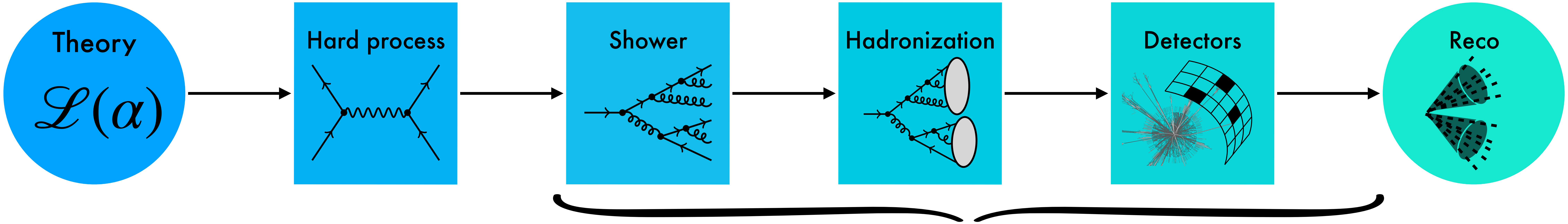


$p(x_{hard}|\alpha)$

$p(x_{reco}|\alpha)$

Need access to the likelihood at reconstruction level!

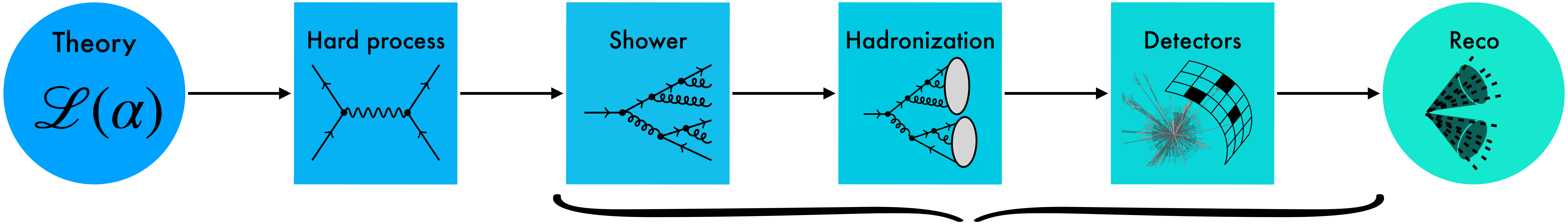
Going for the likelihood



$p(x_{hard} | \alpha)$ $\quad$ $p(x_{reco} | x_{hard}, \alpha)$ $\quad$ $p(x_{reco} | \alpha)$

Hard-scattering and reconstruction linked by forward transfer probability

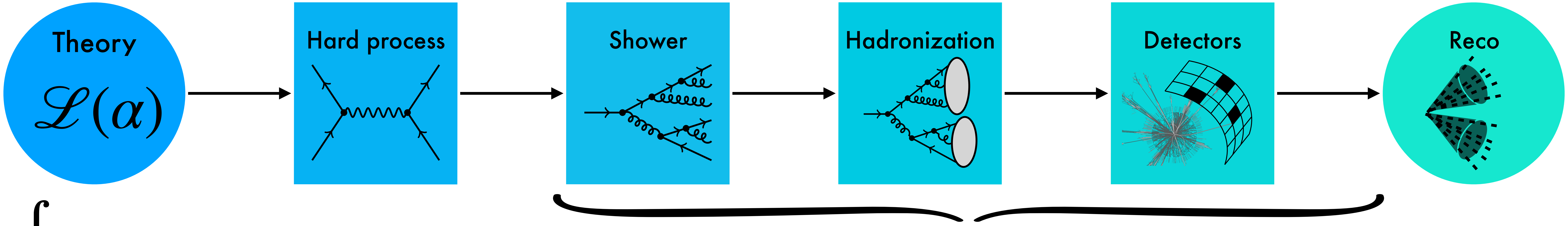
Going for the likelihood



$$p(x_{hard} | \alpha) \qquad\qquad p(x_{reco} | x_{hard}, \alpha)\, \epsilon(x_{hard}, \alpha) \; = \; p(x_{reco} | \alpha)$$

Include an efficiency term to account for acceptance of events

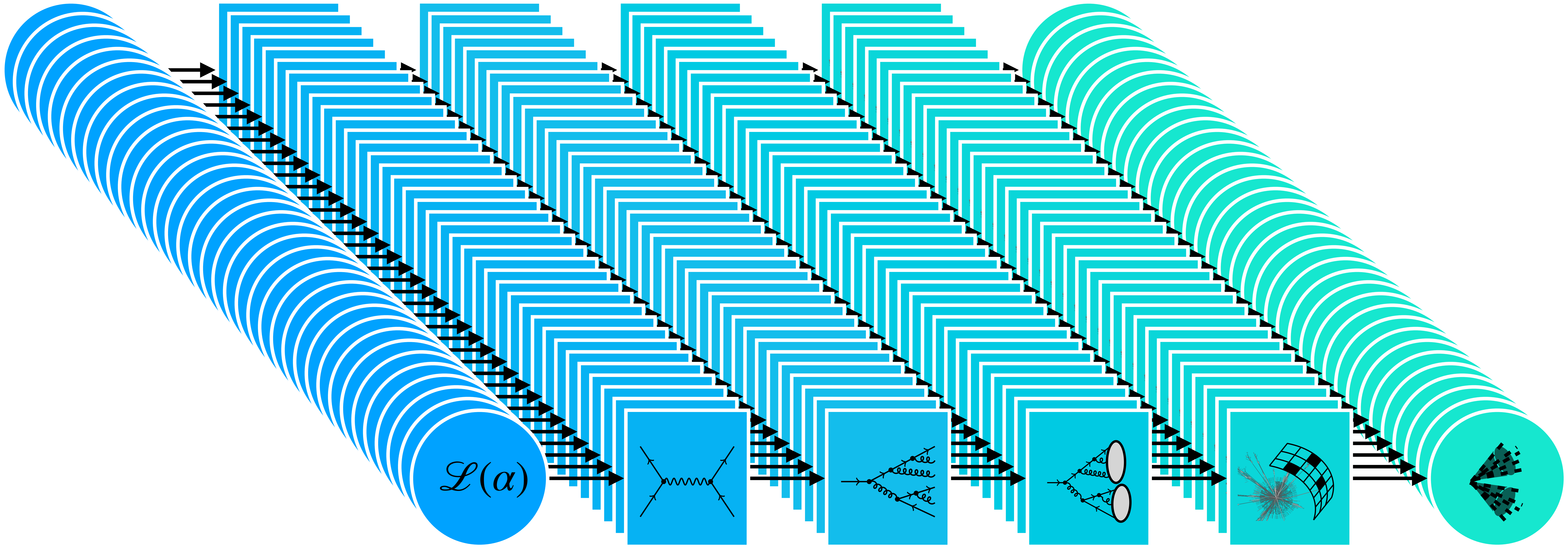
Going for the likelihood



$$\int dx_{hard} \quad p(x_{hard}\,|\,\alpha) \quad p(x_{reco}\,|\,x_{hard}, \alpha)\,\epsilon(x_{hard}, \alpha) \;=\; p(x_{reco}\,|\,\alpha)$$

Integrate over all possible hard-scattering configurations

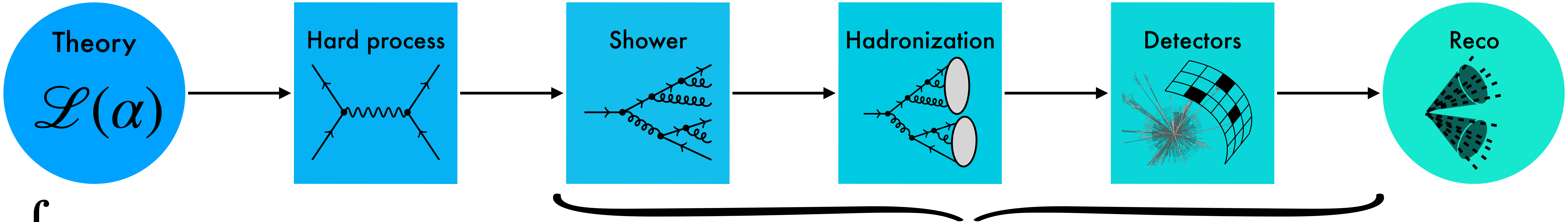
Going for the likelihood



$$\prod_{events\ i} p(x_{reco,i} | \alpha)$$

Event likelihoods are combined into sample likelihoods

The Matrix Element Method



$$\int dx_{hard} \quad p(x_{hard}|\alpha) \quad \underbrace{p(x_{reco}|x_{hard},\alpha)\,\epsilon(x_{hard},\alpha)} = p(x_{reco}|\alpha)$$

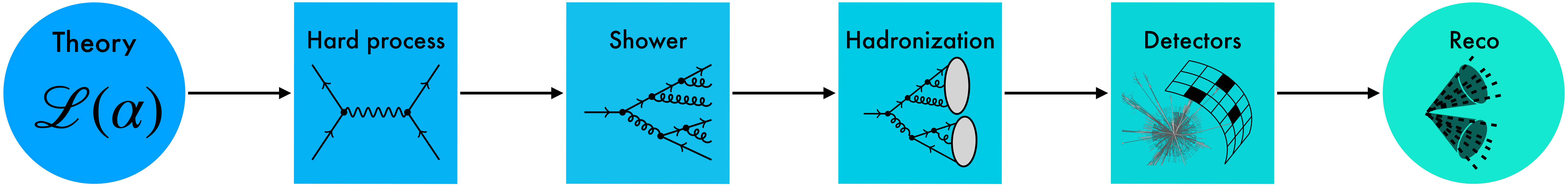
+++ Unbinned and multivariate by design

+++ Optimal use of information derived from Newman-Pearson lemma

— — Transfer probability and efficiency not known

— — Integral numerically very challenging

The Matrix Element Method



$$\int dx_{hard} \quad p(x_{hard}\,|\,\alpha) \quad p(x_{reco}\,|\,x_{hard},\alpha)\,\epsilon(x_{hard},\alpha) \;=\; p(x_{reco}\,|\,\alpha)$$

+++ Unbinned and multivariate by design

+++ Optimal use of information derived from Newman-Pearson lemma

— — ~~Transfer probability and efficiency not known~~ **USE MACHINE LEARNING**

— — ~~Integral numerically very challenging~~ **USE MACHINE LEARNING**

The Transfer Network

Intractable

$$\int dx_{hard}\; p(x_{hard} | \alpha)\; p(x_{reco} | x_{hard}, \alpha)\; \epsilon(x_{hard}, \alpha) \;=\; p(x_{reco} | \alpha)$$

The Transfer Network

Intractable

$$\int dx_{hard}\; p(x_{hard}|\alpha)\; p(x_{reco}|x_{hard})\; \epsilon(x_{hard}, \alpha) \;=\; p(x_{reco}|\alpha)$$

The Transfer Network

Intractable

$$\int dx_{hard}\; p(x_{hard}|\alpha)\; p(x_{reco}|x_{hard})\; \epsilon(x_{hard}, \alpha) \;=\; p(x_{reco}|\alpha)$$

Transfer probability is analytically intractable

Transfer can be simulated to generate paired data x_{hard} , x_{reco}

Generative Neural Network to encode the transfer probability

The Transfer Network

$$\int dx_{hard}\, p(x_{hard}|\alpha)\, p(x_{reco}|x_{hard})\, \epsilon(x_{hard}, \alpha) = p(x_{reco}|\alpha)$$

Intractable

x_{hard} —Simulation→ x_{reco} —Cuts→ ✅ / ❌

The transfer network is trained only on accepted pairs

➡ Bias in the learned transfer probability $p(x_{reco}|x_{hard})$ that we need to correct!

The Acceptance Network

Unknown

$$\int dx_{hard}\; p(x_{hard}|\alpha)\; p(x_{reco}|x_{hard})\; \epsilon(x_{hard}, \alpha) = p(x_{reco}|\alpha)$$

The Acceptance Network

Unknown

$$\int dx_{hard}\; p(x_{hard}|\alpha)\; p(x_{reco}|x_{hard})\; \boxed{\epsilon(x_{hard})} = p(x_{reco}|\alpha)$$

The Acceptance Network

$$\int dx_{hard}\; p(x_{hard}|\alpha)\; p(x_{reco}|x_{hard})\; \epsilon(x_{hard}) = p(x_{reco}|\alpha)$$

Unknown

Need to encode the efficiency at hard-scattering level

Transfer can be simulated to generate labeled data $x_{hard} \rightarrow x_{reco}(x_{hard}) \rightarrow$ ✅ / ❌

Classifier Neural Network to encode the acceptance probability

The Sampling Network

Challenging

$$\int dx_{hard}\ p(x_{hard}|\alpha)\ p(x_{reco}|x_{hard})\ \epsilon(x_{hard}) = p(x_{reco}|\alpha)$$

The Sampling Network

Challenging

$$\int dx_{hard}\ p(x_{hard}|\alpha)\ p(x_{reco}|x_{hard})\ \epsilon(x_{hard}) = p(x_{reco}|\alpha)$$

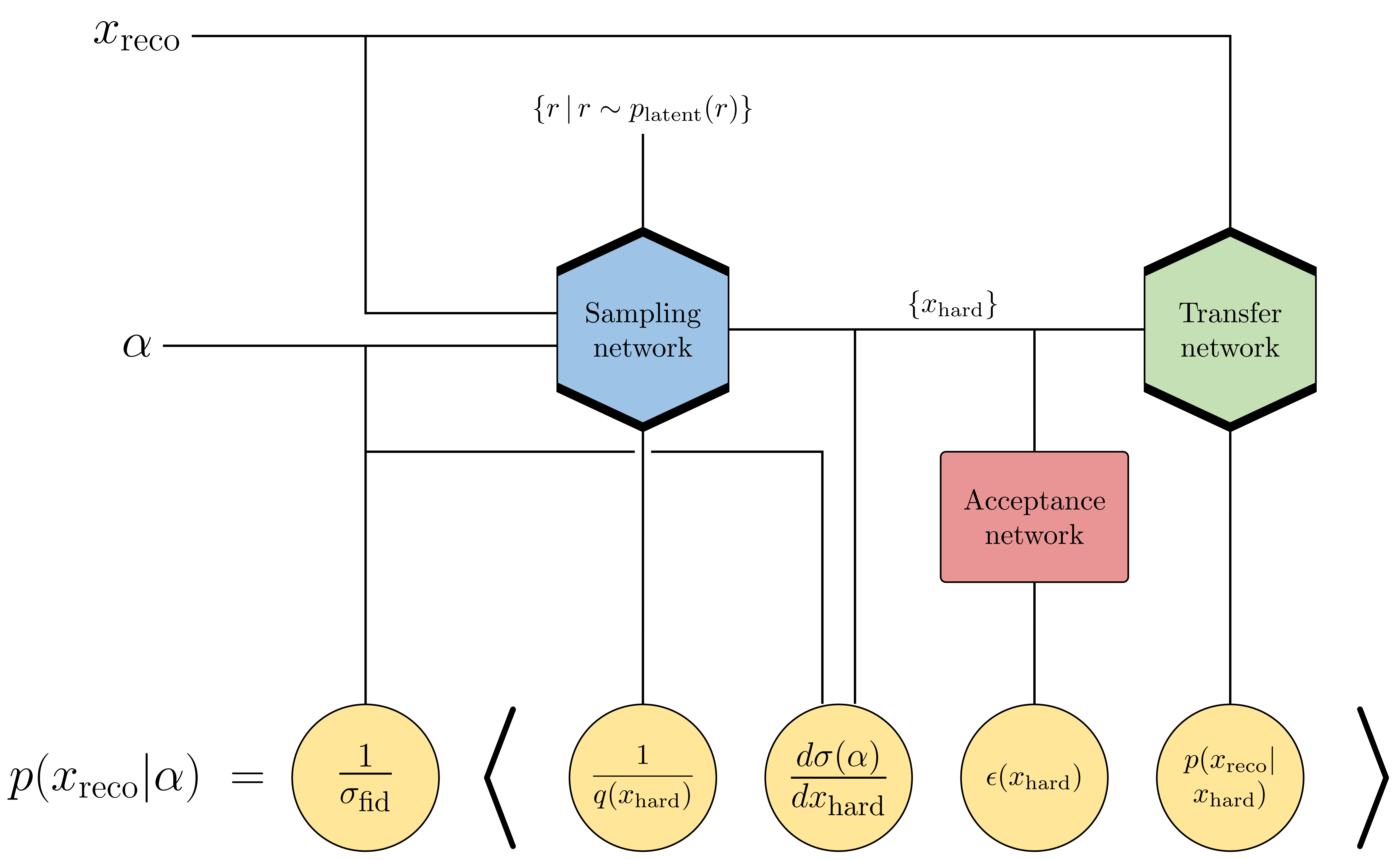
$$= \left\langle \frac{1}{q(x_{hard})} p(x_{hard}|\alpha)\ p(x_{reco}|x_{hard})\ \epsilon(x_{hard}) \right\rangle_{x_{hard}\sim q(x_{hard})}$$

Integral becomes trivial if : $q(x_{hard}) \propto p(x_{hard}|x_{reco},\alpha)\epsilon(x_{hard})$

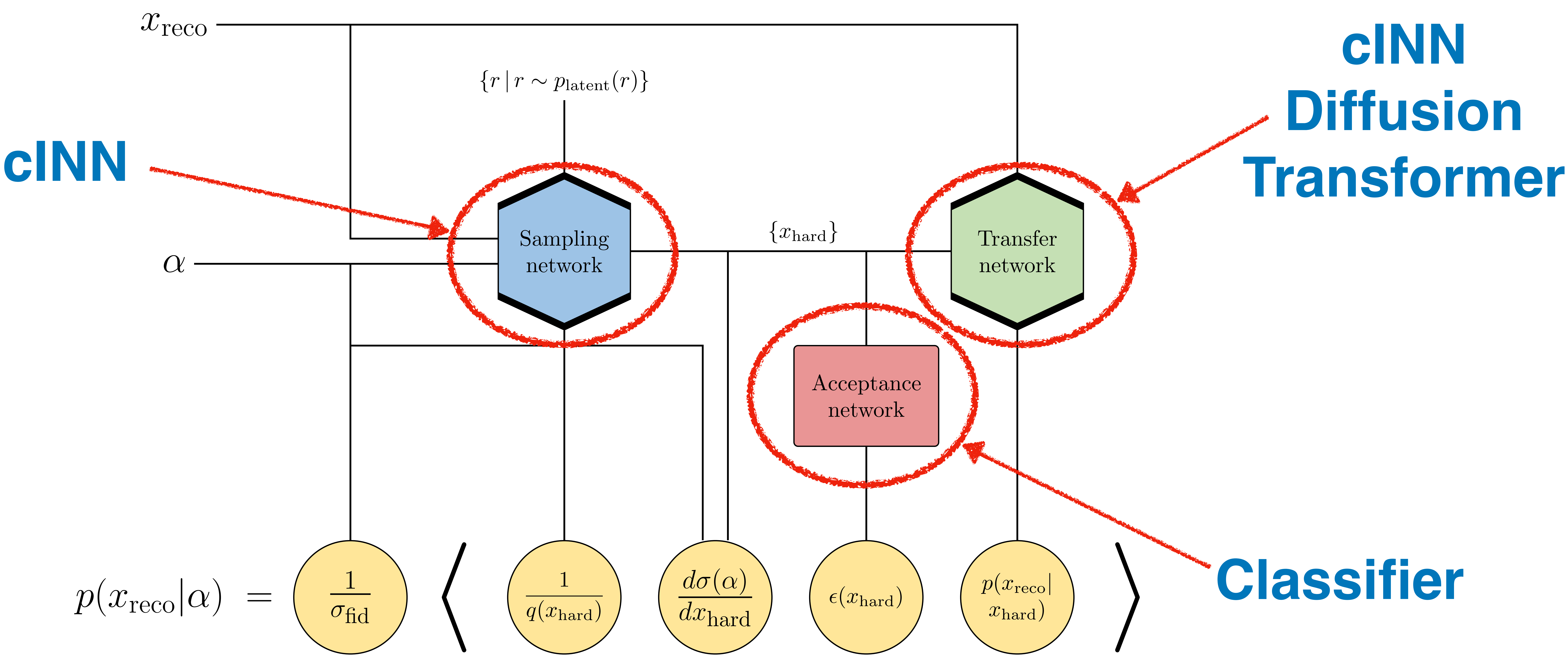
Generative Neural Network to encode sampling distribution

$$r \sim p_{latent}(r) \longleftrightarrow x_{hard}(r) \sim q_{\phi}(x_{hard})$$

Machine-learned MEM



Machine-learned MEM



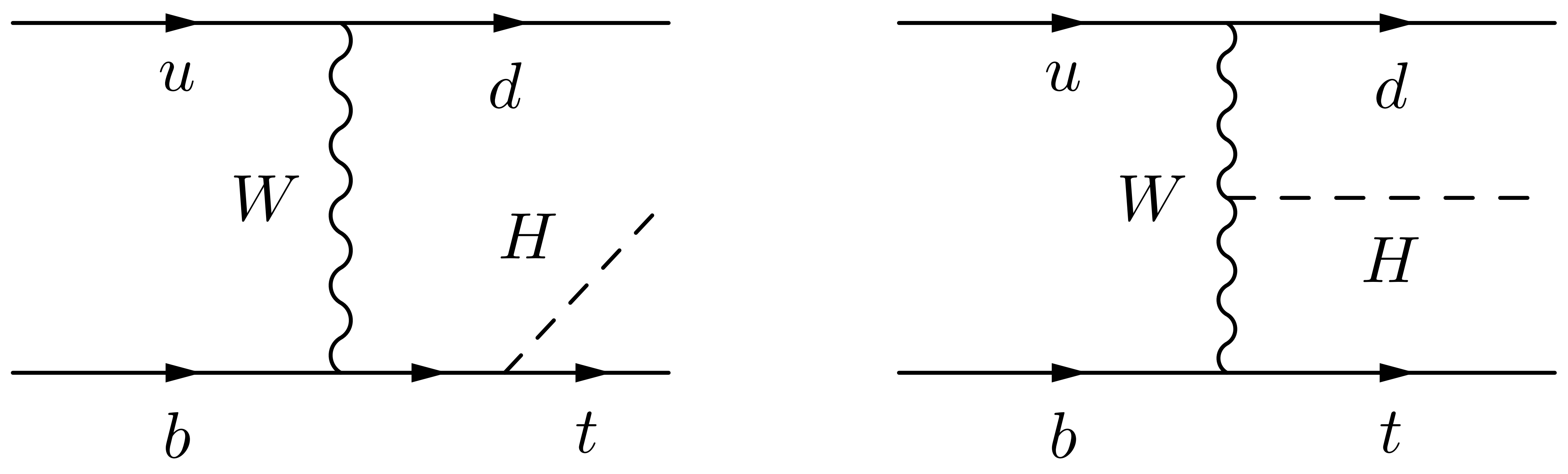
The physics example

The problem: Measuring a CP-phase in the top Yukawa coupling

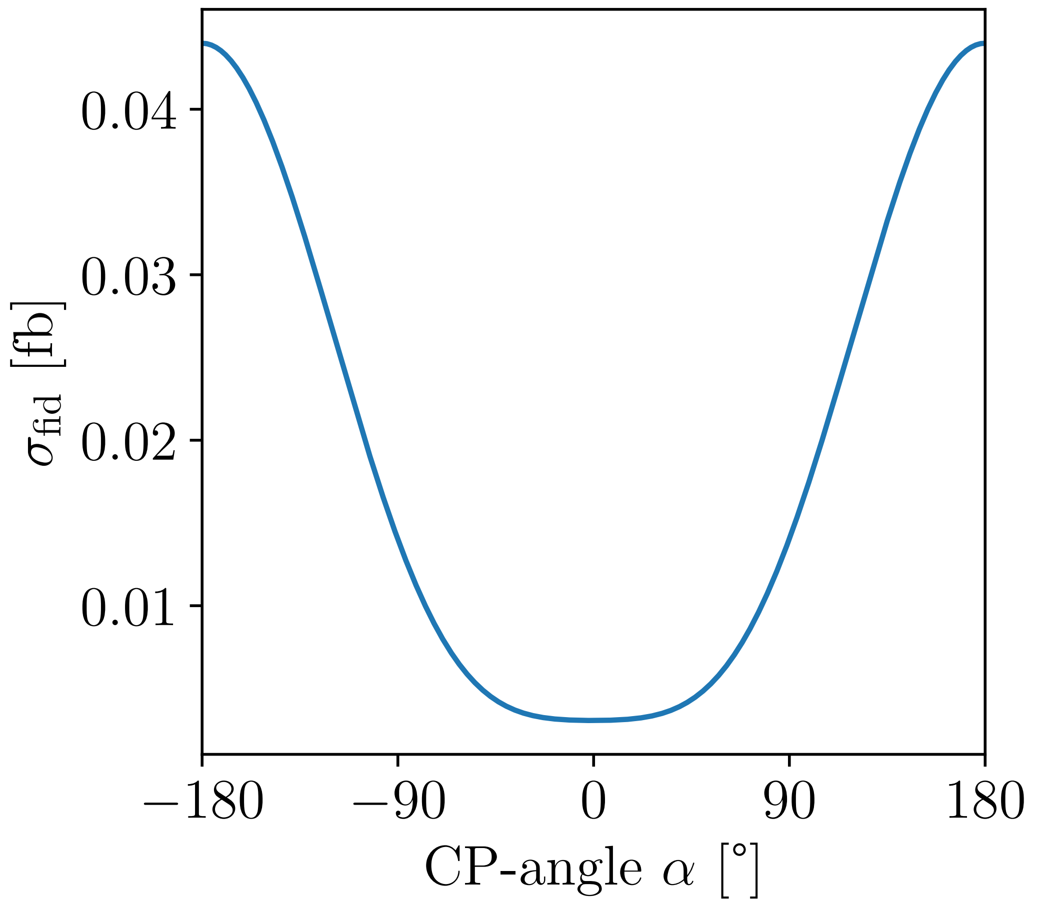
$$\mathscr{L}_{t\bar{t}H} = -\frac{y_t}{\sqrt{2}}\left[\cos\alpha\bar{t}t + \frac{2}{3}i\sin\alpha\bar{t}\gamma_5 t\right]H$$

The process: Associated single-top and Higgs production

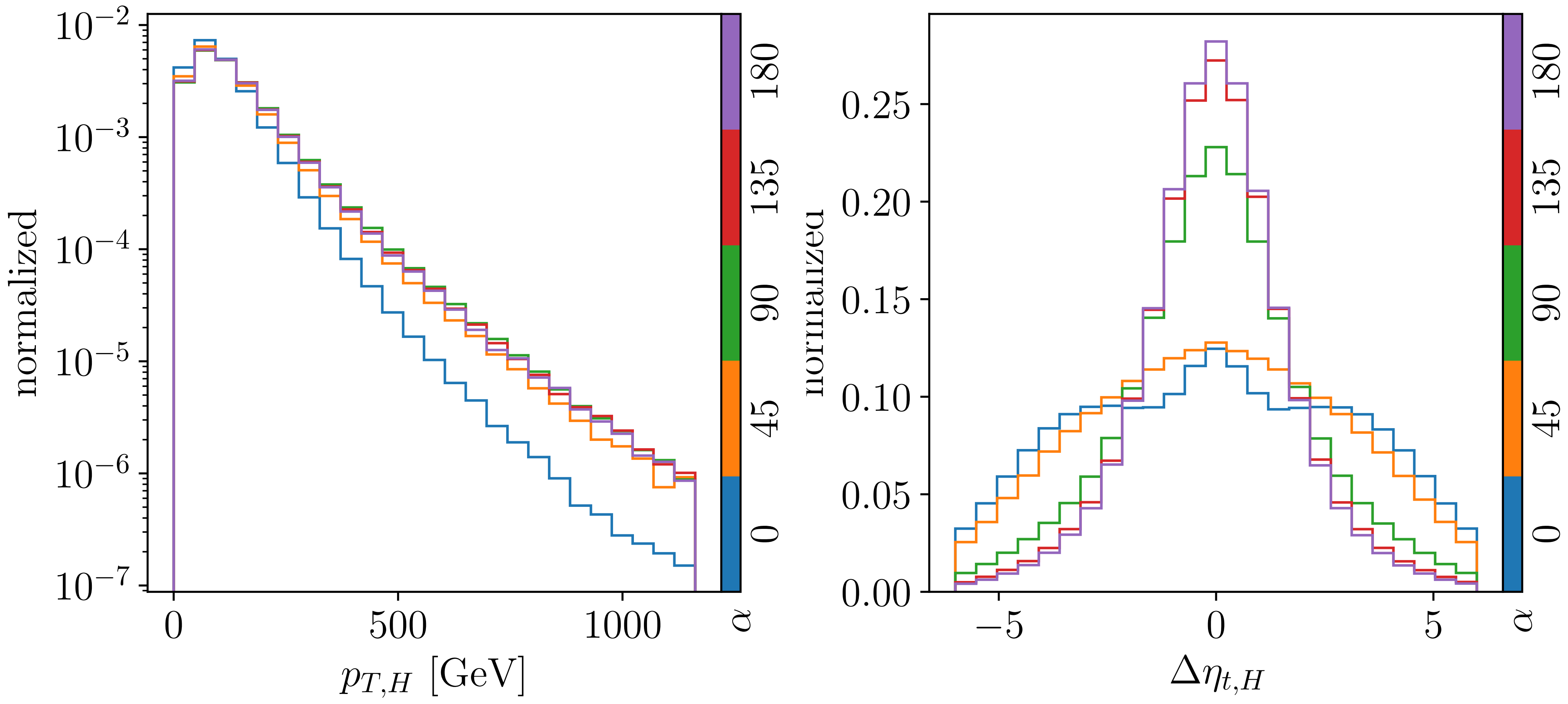
$$pp \to tHj \to (bjj)\,(\gamma\gamma)\,j \;+\; \text{ISR Jets}$$



Why MEM here?



Cross-section very small and insensitive to variations in α

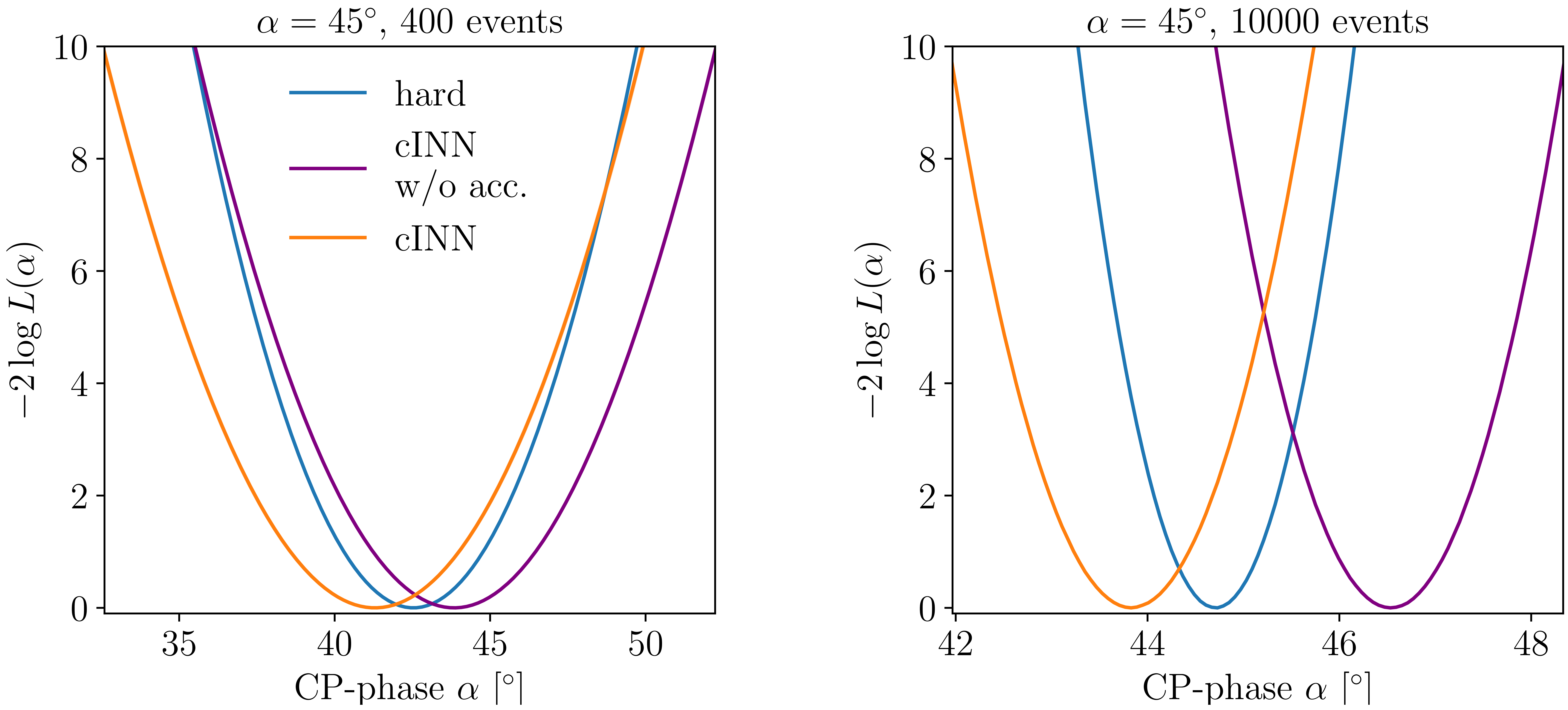


Hard-scattering $p(x_{hard} \,|\, \alpha)$ kinematics are sensitive

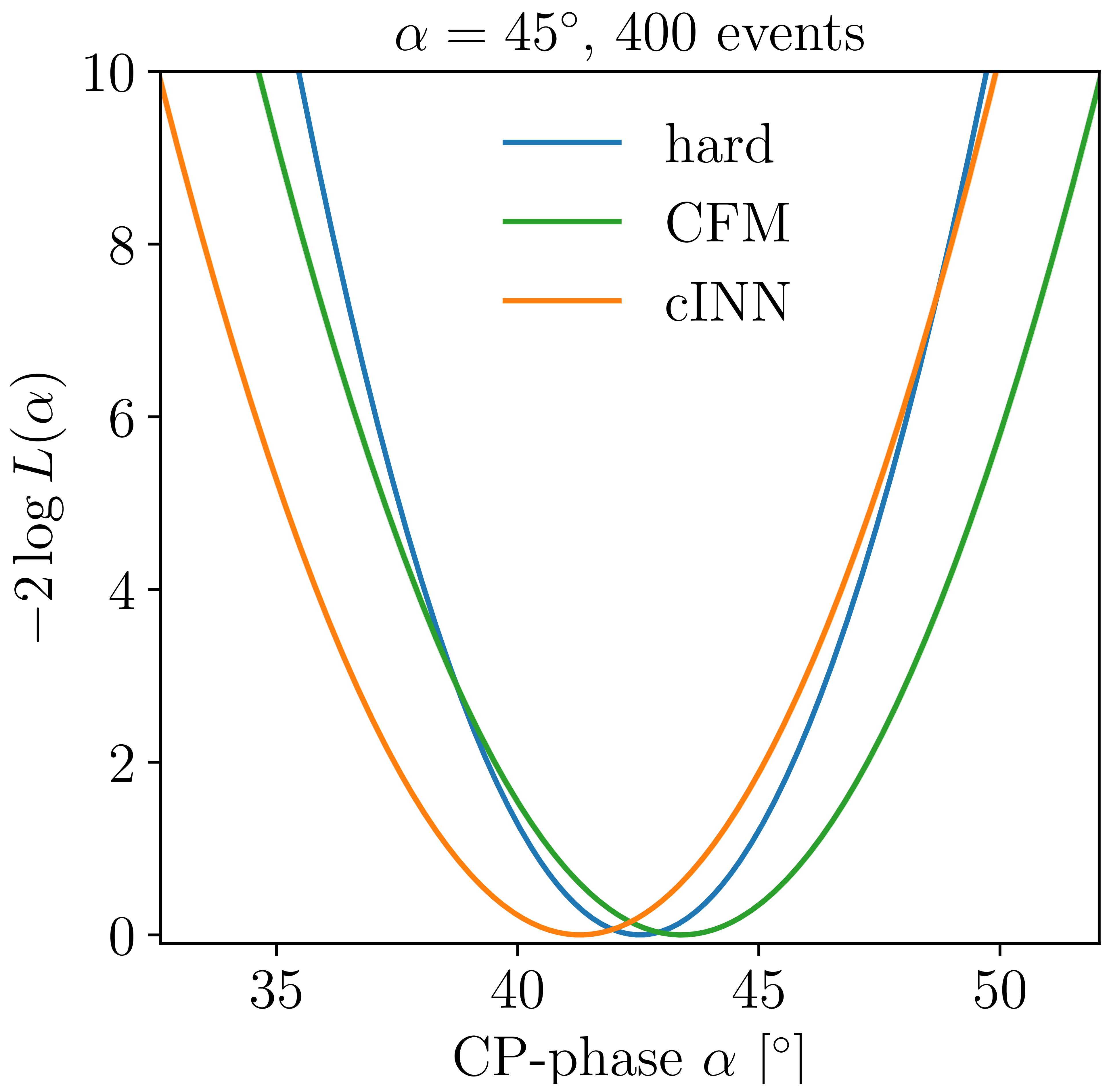
Ideal use case for the Matrix Element Method

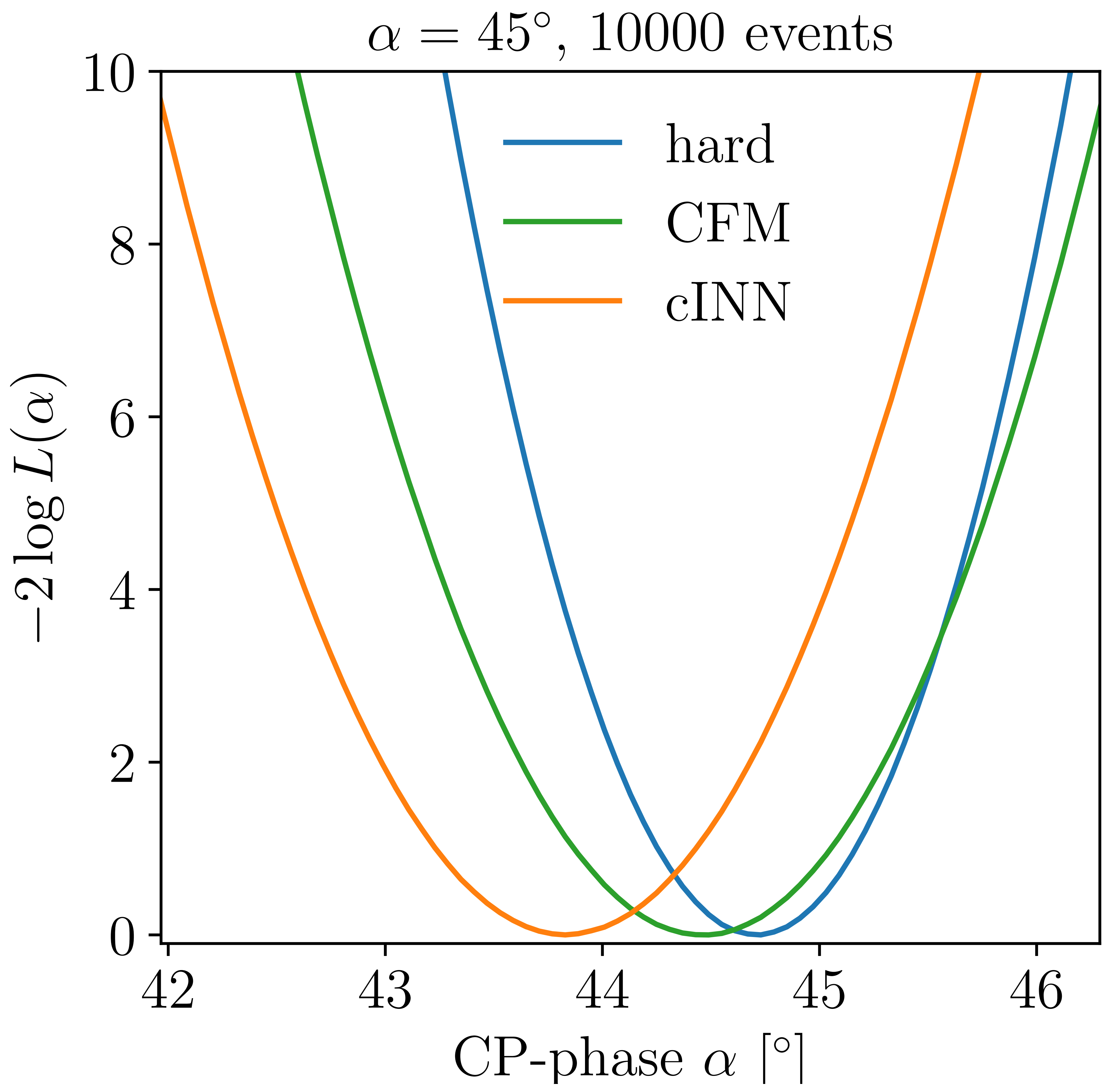
[Figures taken from Butter et al arXiv:2210.00019]

Baseline Results

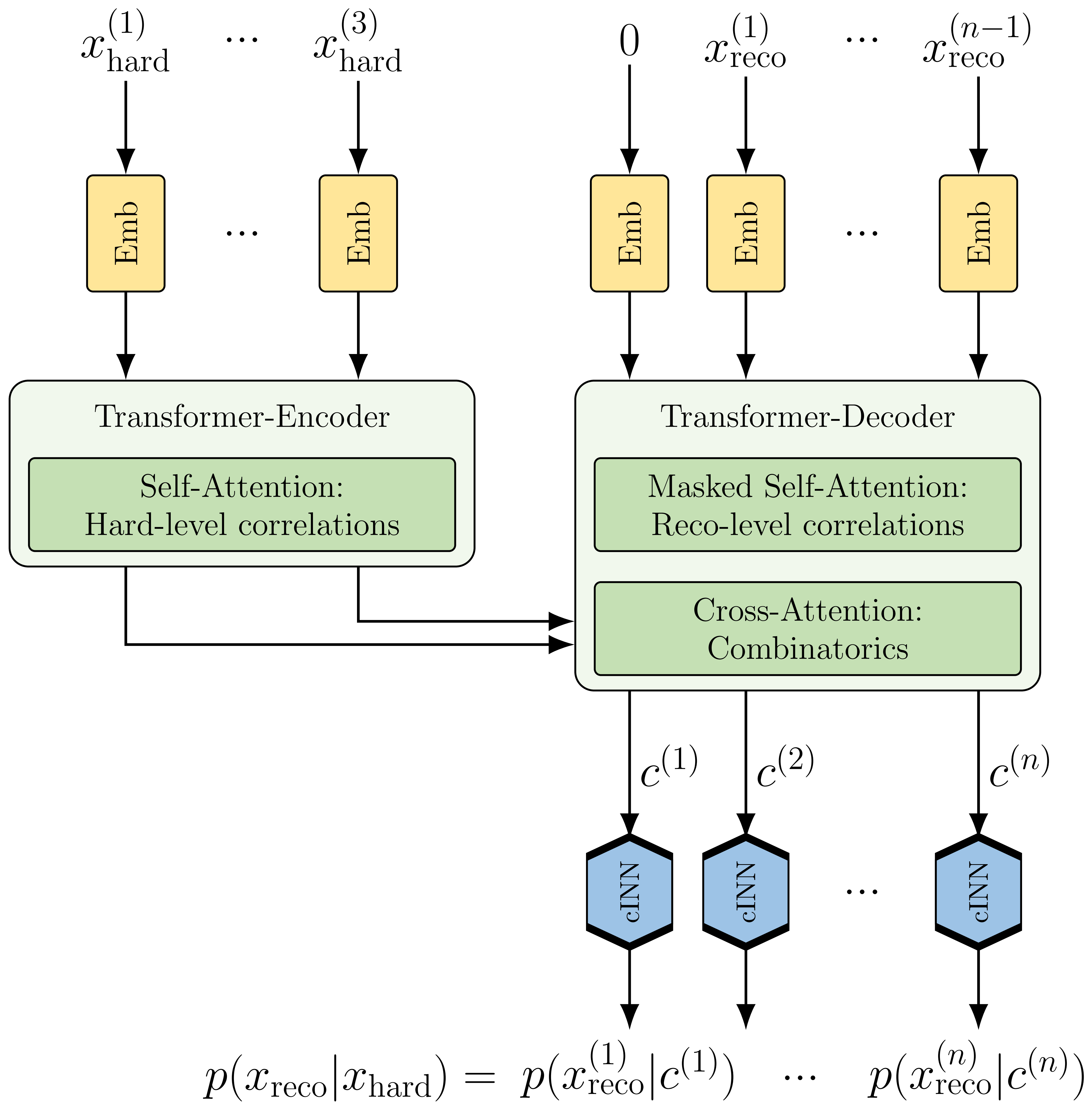


Transfer-Diffusion Results

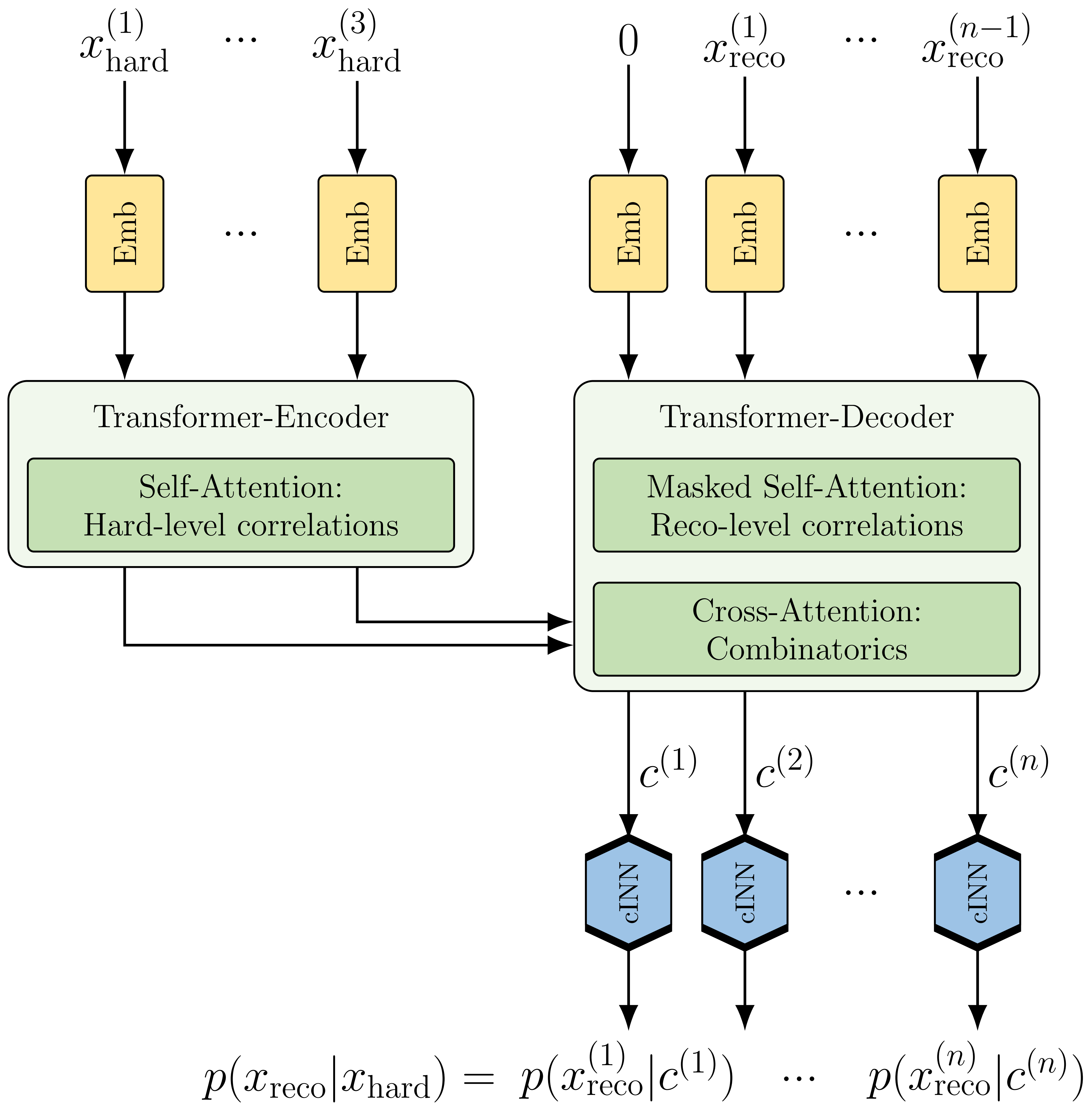




The Transfermer (Transfer-Transformer)



The Transfermer (Transfer-Transformer)



Why a Transformer here?

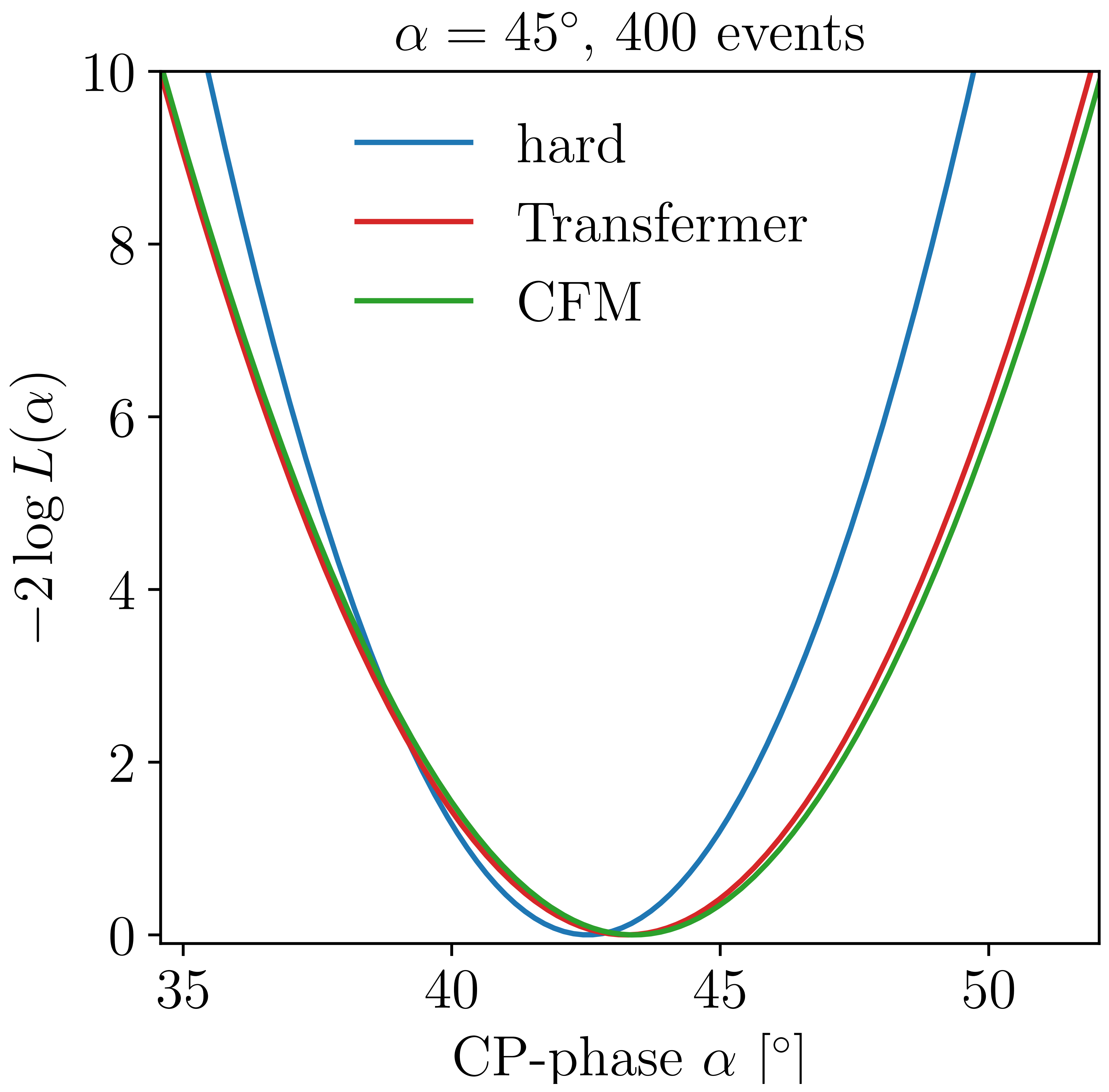
1. Jet combinatorics

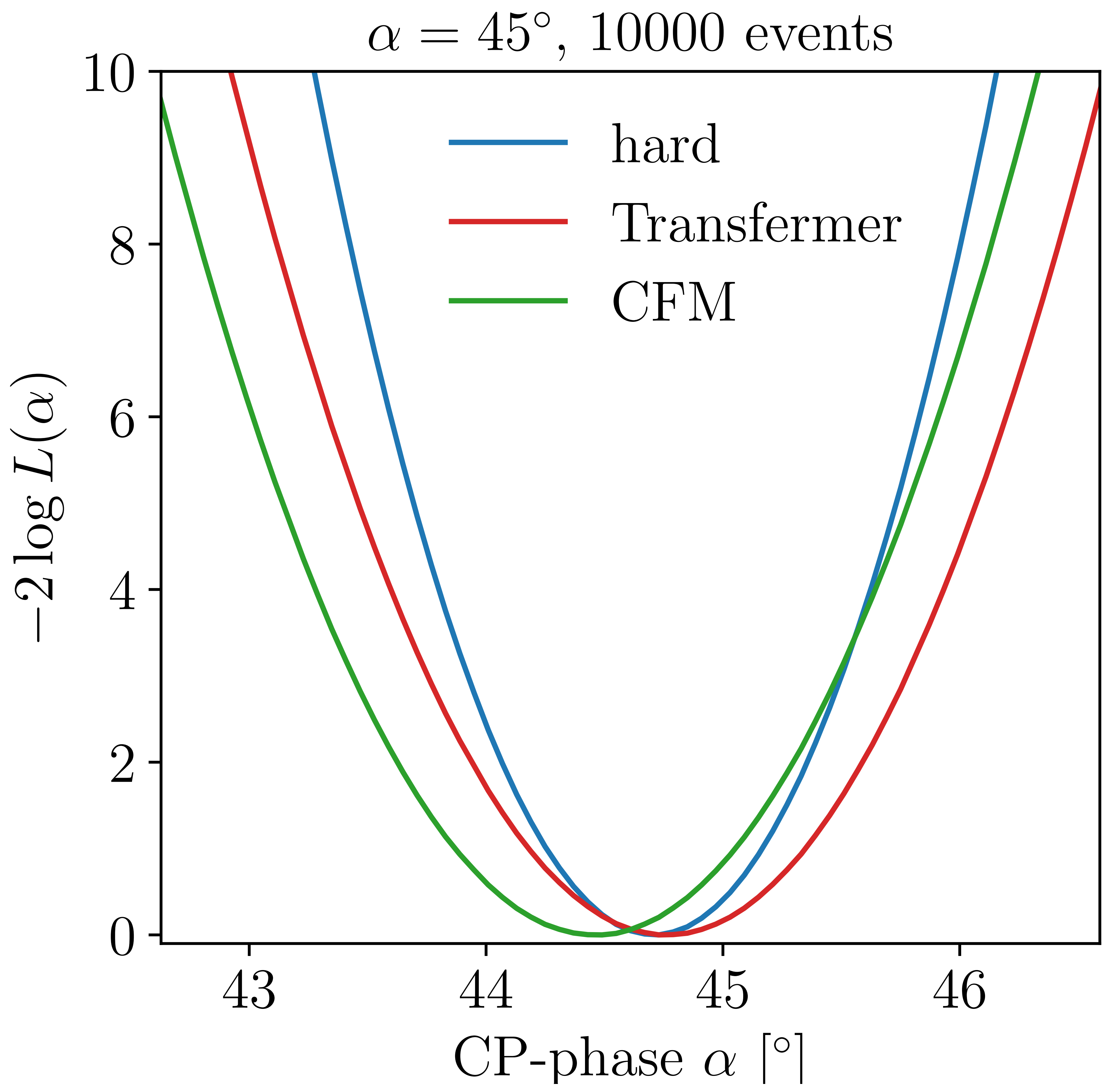
$$pp \to tHj \to (bjj)\ (\gamma\gamma)\ j\ + \text{ISR Jets}$$

2. Can handle varying particle multiplicity

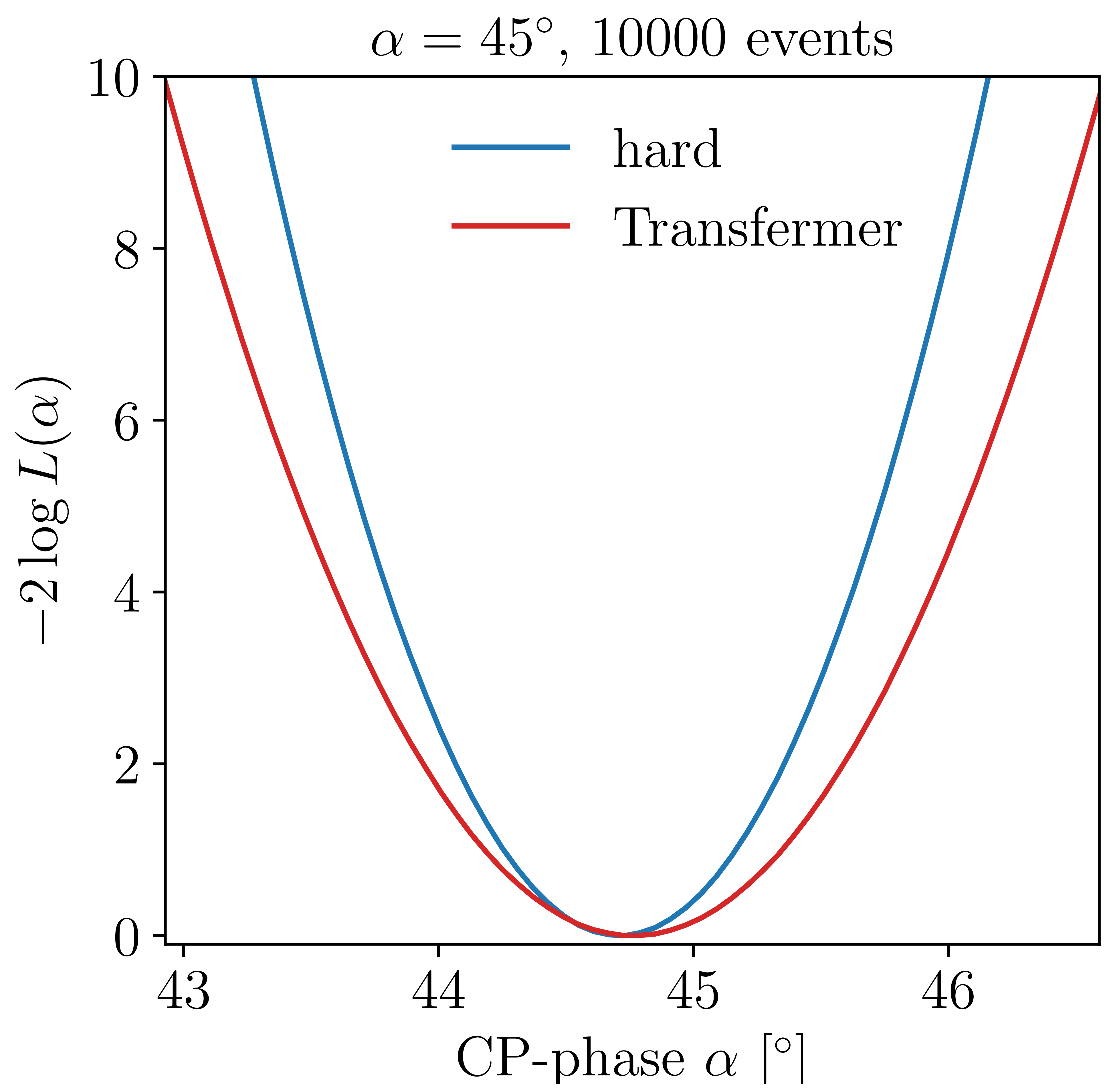
⟶ NLO

Transfermer Results





Estimating Uncertainty

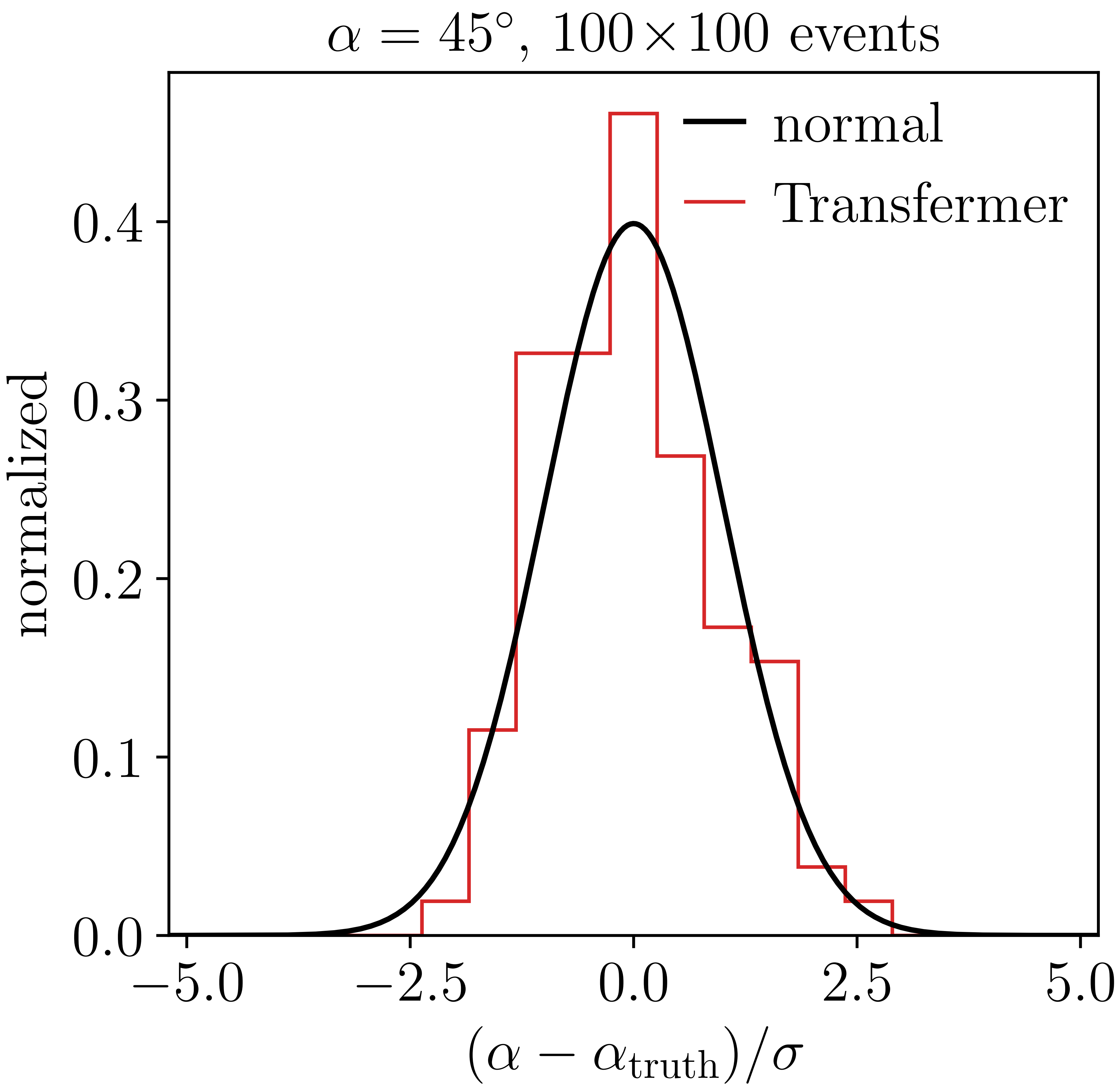


Estimate uncertainty using replicas:

Integration uncertainty:
Resample the MC points using bootstrapping

Network uncertainty:
Use Bayesian NN and resample for each replica

Estimating Calibration



Divide the 10k events into 100 samples of 100 events

Look at this distribution of the minima around the true α

Summary and Outlook

We present a three network setup to make the **MEM tractable** and **precise**

1) **Transfer-Network** encoding the transfer probability $p(x_{reco} \,|\, x_{hard})$
2) **Acceptance-Network** encoding the efficiency $\epsilon(x_{hard})$
3) **Sampling-Network** encoding the proposal distribution $q(x_{hard})$

For the **Transfer-Network** going from cINNs to **Diffusion/Transformer** models improves precision
For the **Unfolding-Network cINNs** with RQS-splines are still **state-of-the-art**

Bootstrapping and Bayesian NNs allow us to control the **uncertainties**

Extend our formalism to **NLO**

Test our setup on an **actual analysis** and/or more challenging processes

Improving integration convergence

$$\left\langle \frac{1}{q(x_{hard})} p(x_{hard}|\alpha)\, p(x_{reco}|x_{hard})\, \epsilon(x_{hard}) \right\rangle_{x_{hard} \sim q(x_{hard})}$$

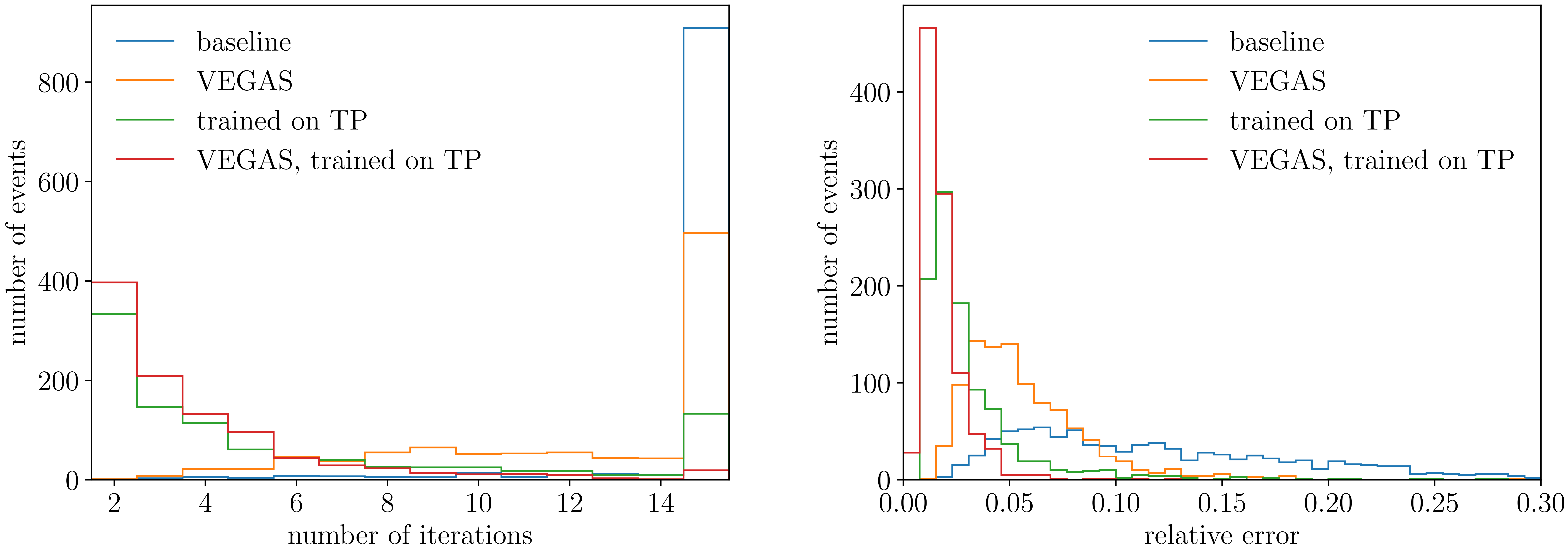
1 In practice $p_\theta(x_{reco}|x_{hard}) \neq p(x_{reco}|x_{hard})$

→ Train Sampling Network on trained Transfer Probability

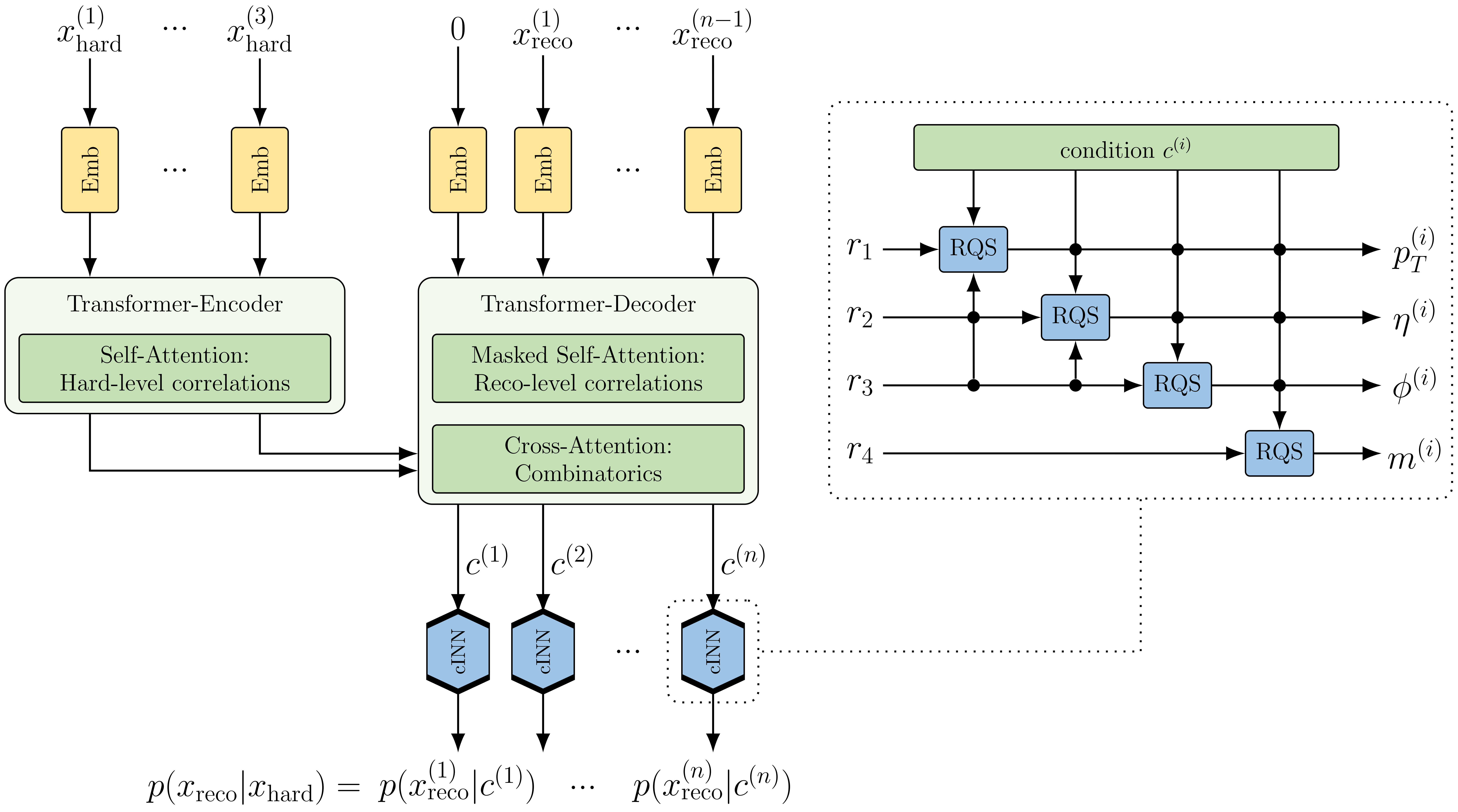
2 In practice $q_\theta(x_{hard}) \neq q_{ideal}(x_{hard})$

→ Further refine the cINN latent space with a VEGAS grid

Improving integration convergence



The Transfermer (Transfer-Transformer)



Transfer-Network observables