Beyond Images: Leveraging Stable Diffusion Techniques for Particle Physics Simulations

Dr. Nilotpal Kakati

Nathalie Soybelman

Eilam Gross

Dmitrii Kobylianskii

Dr. Etienne Dreyer

Eddie Shields

CERN

WEIZMANN INSTITUTE OF SCIENCE

24/11/2025

ORSAY

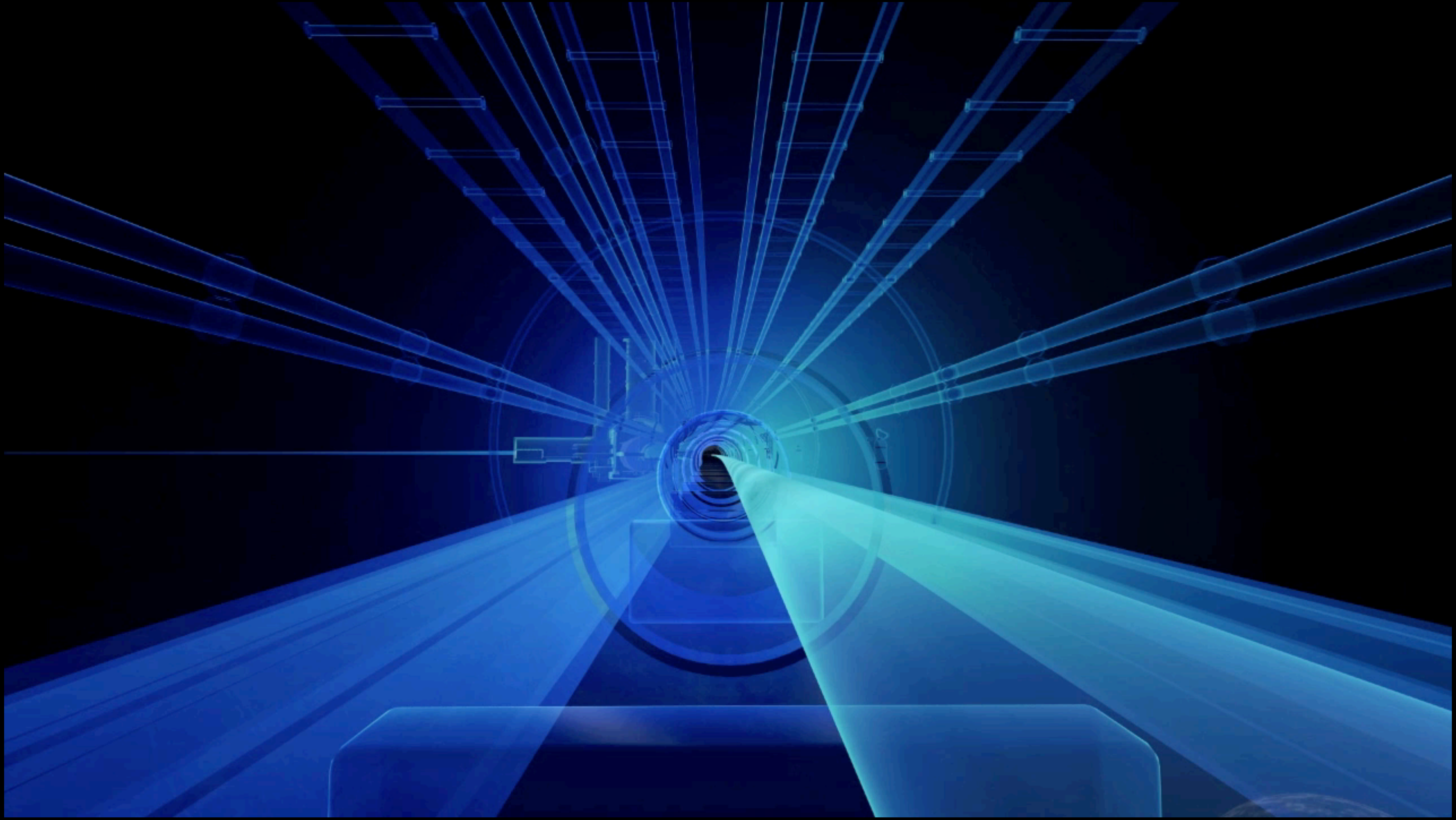
Outline

- **Introduction:**
  Particle detectors and particle-flow objects; simulation challenges

- **Analogy with Stable Diffusion 3.x:**
  Generative models in AI, diffusion vs flow matching, conceptual parallels to PF simulation

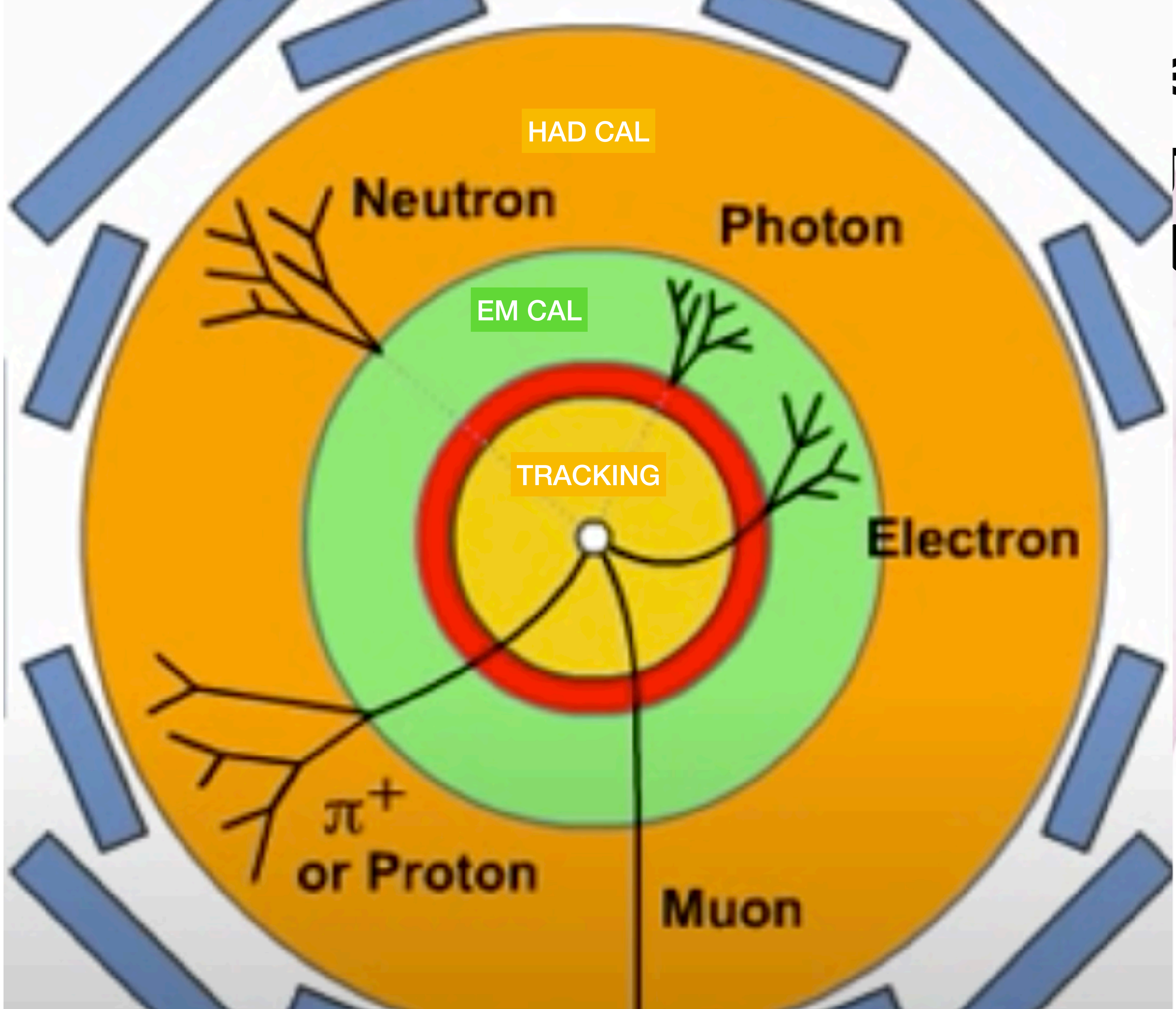
- **Method & Results:**
  The set-to-set generative model (Parnassus), architecture (transformers, flow matching), experiments and performance (accuracy, speedup, generalization)

- **Conclusion:**
  Impact and integration into HEP workflows



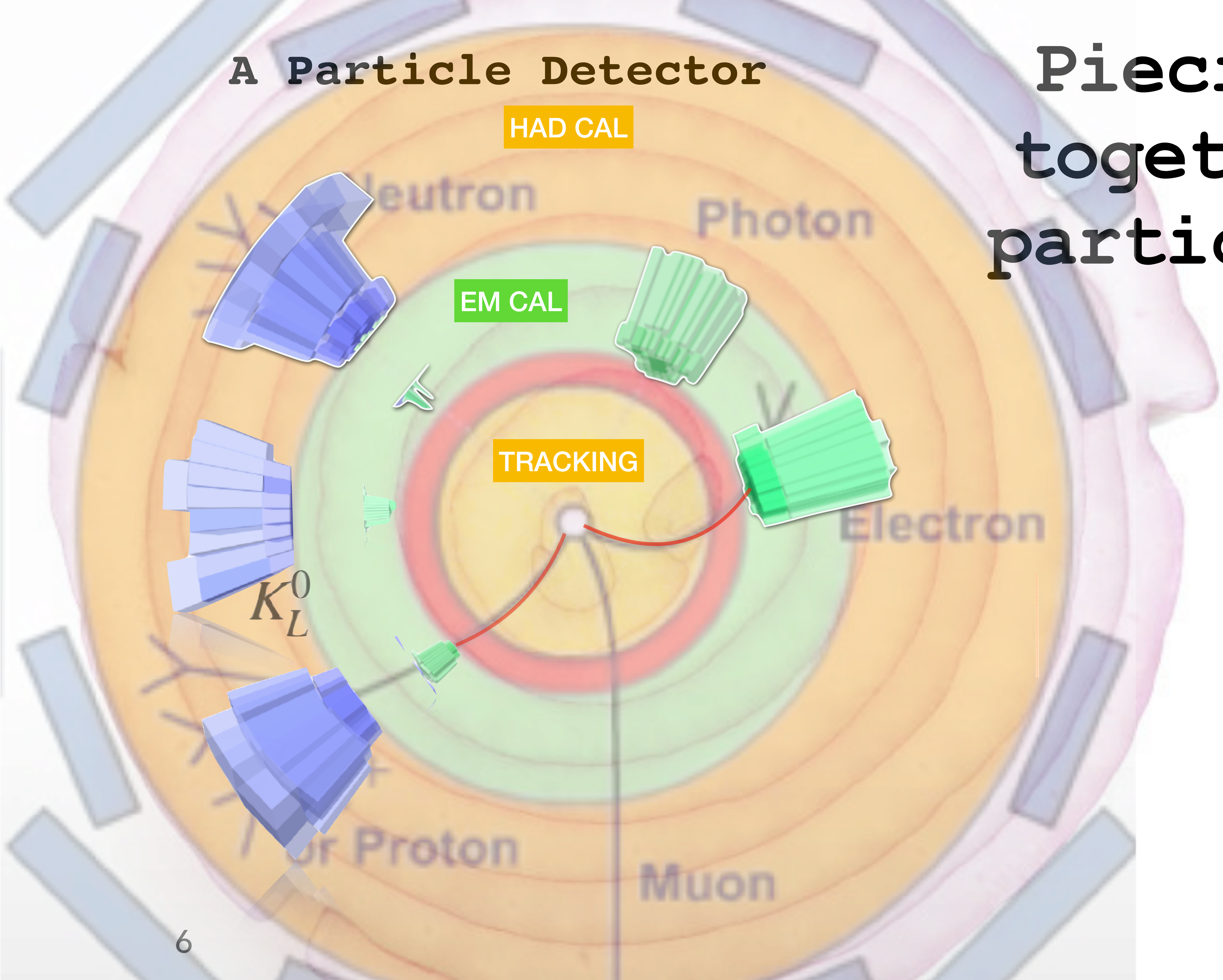
A Particle Detector

Piecing together particles



ecing
ether
ticles

Piecing together particles



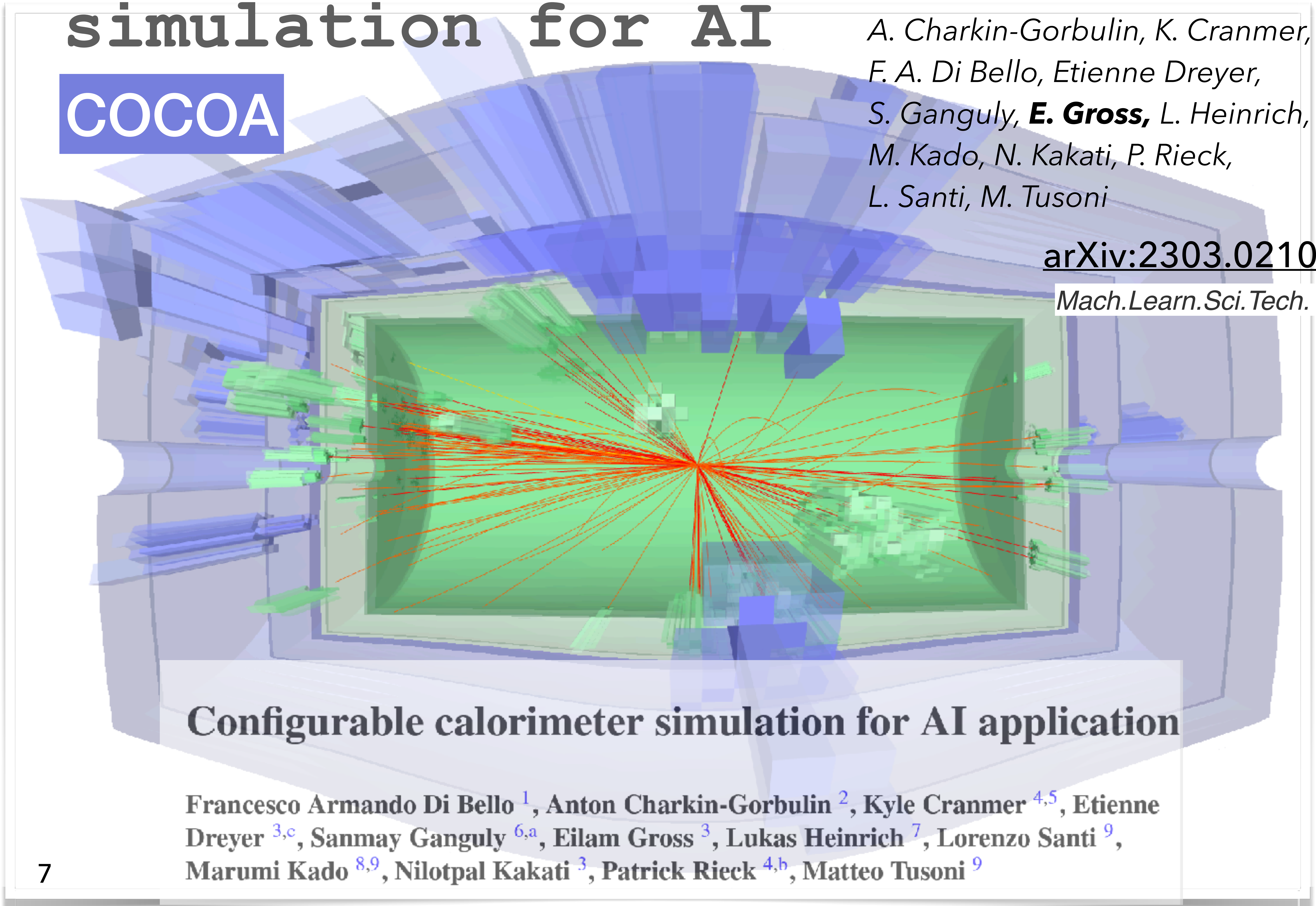
Configurable Calorimeter simulation for AI

COCOA

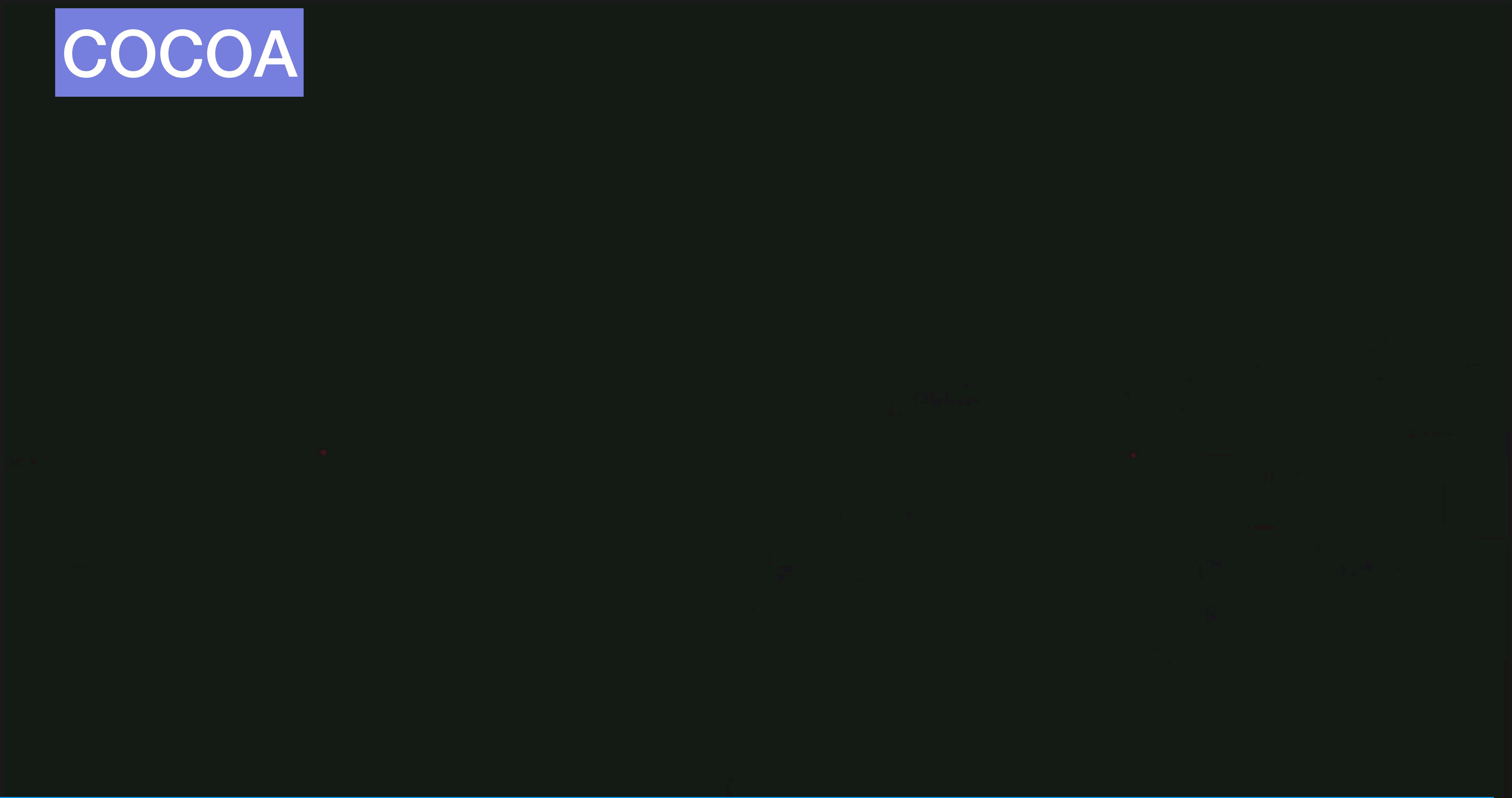
*A. Charkin-Gorbulin, K. Cranmer, F. A. Di Bello, Etienne Dreyer, S. Ganguly, **E. Gross,** L. Heinrich, M. Kado, N. Kakati, P. Rieck, L. Santi, M. Tusoni*

arXiv:2303.02101

Mach.Learn.Sci.Tech. 4 (2023) 3, 035042



COCOA Event Display

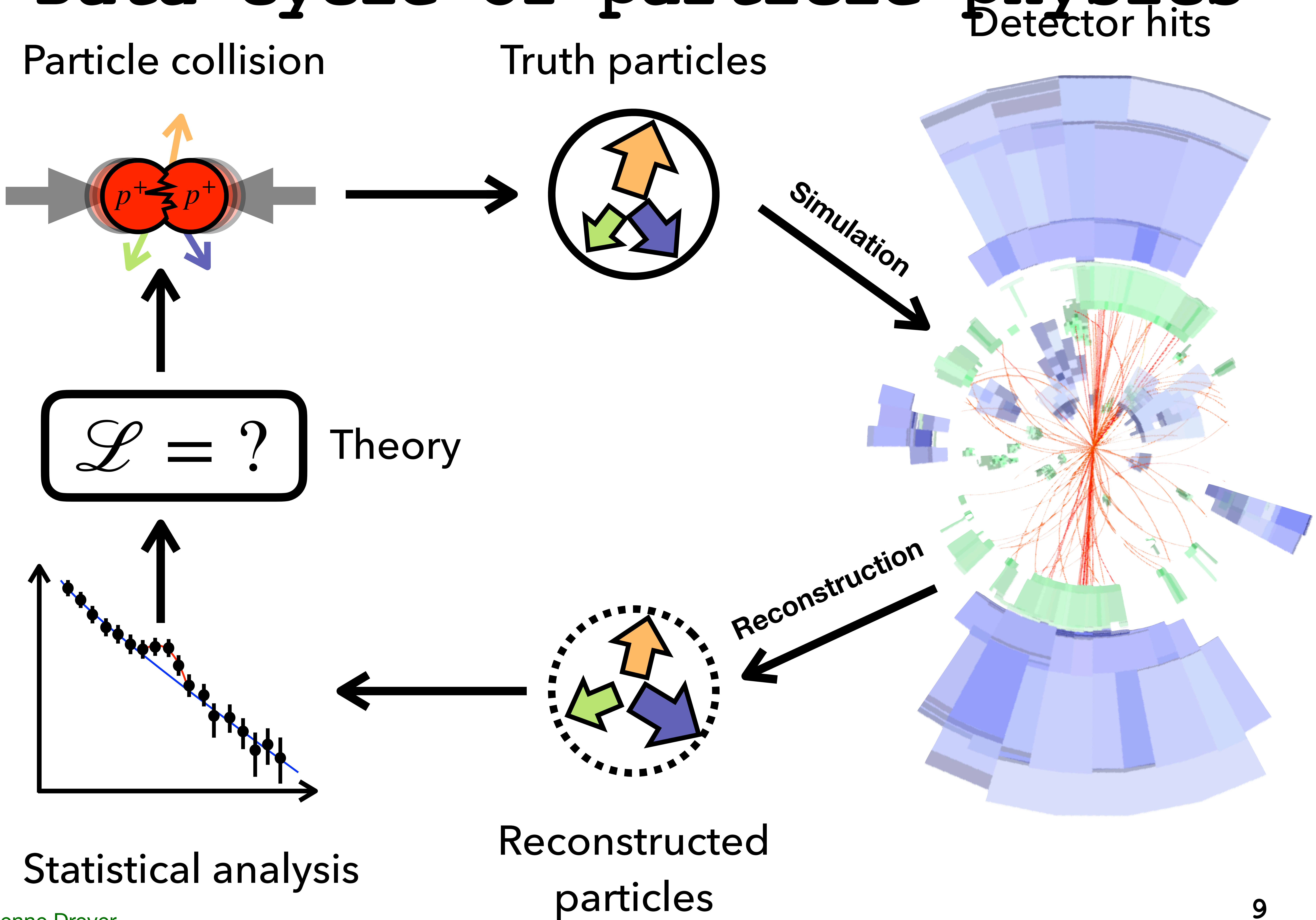


Challenge 1: Reconstruct the Detector responses into Particle Flow Candidates

Challenge 2: Fast Simulation of Detector Response / Particle Flow Candidates

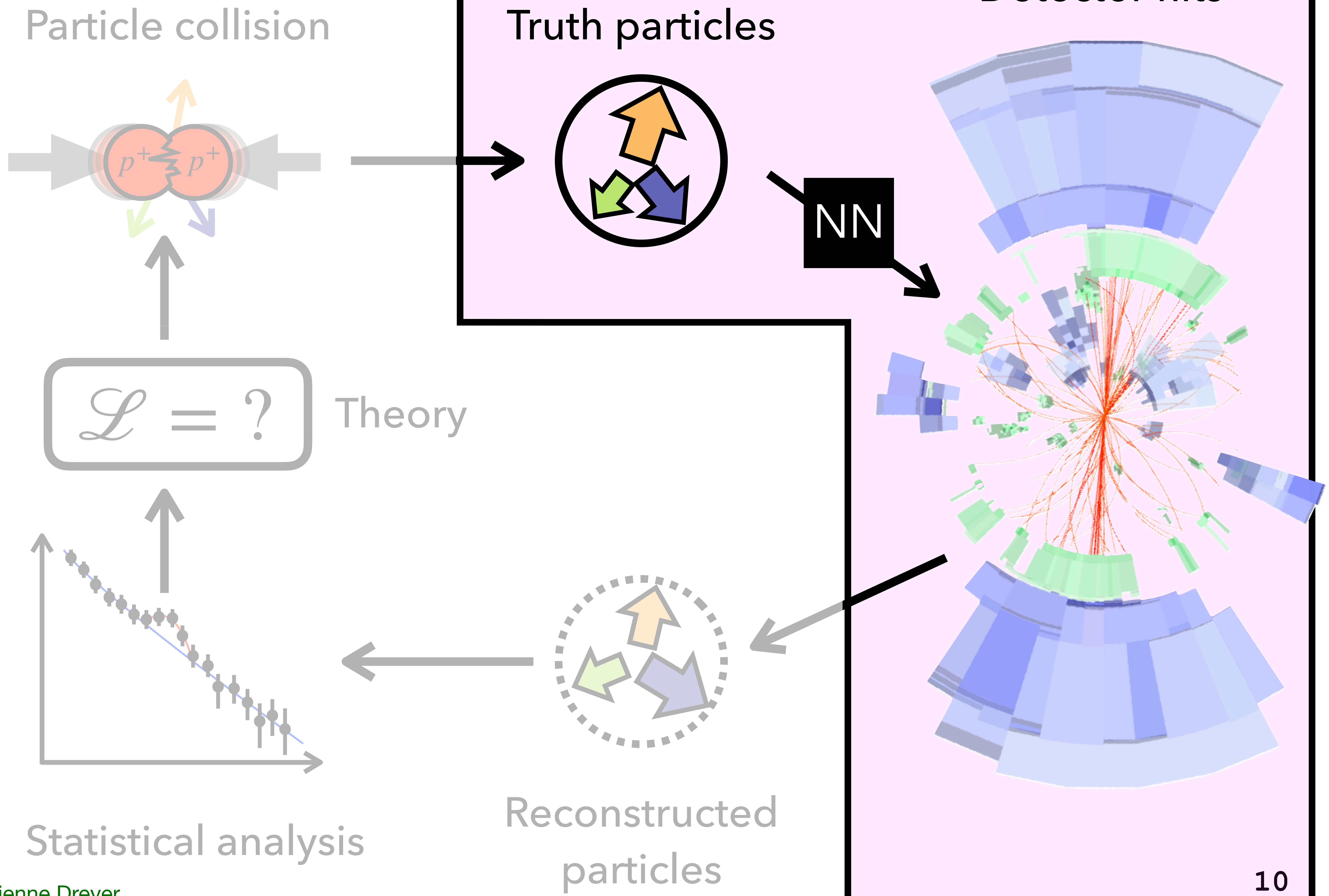
Animation by Nilopal Kakati

Data cycle of particle physics



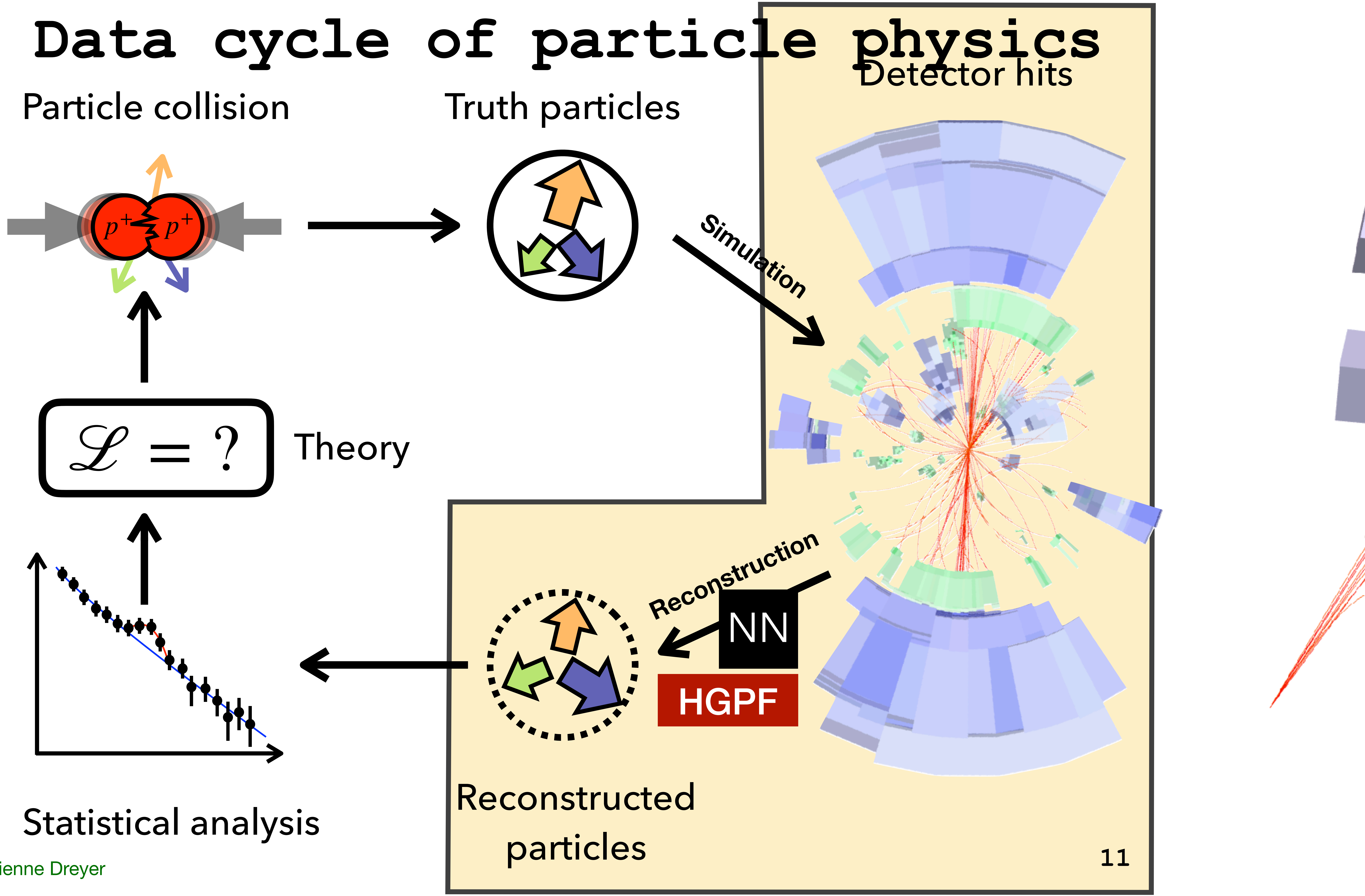
Slide from Etienne Dreyer

Data cycle of particle physics



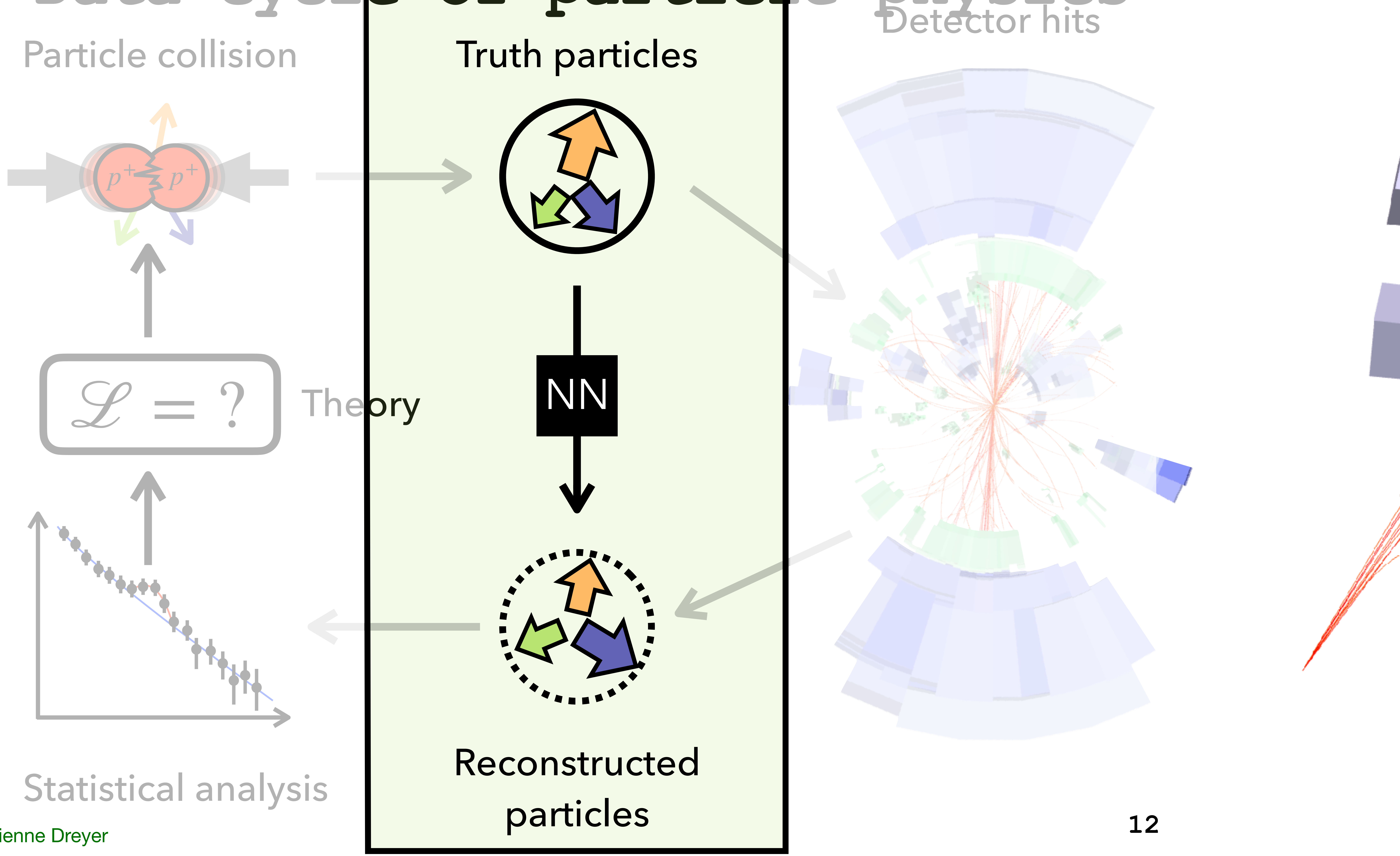
Slide from Etienne Dreyer

Data cycle of particle physics



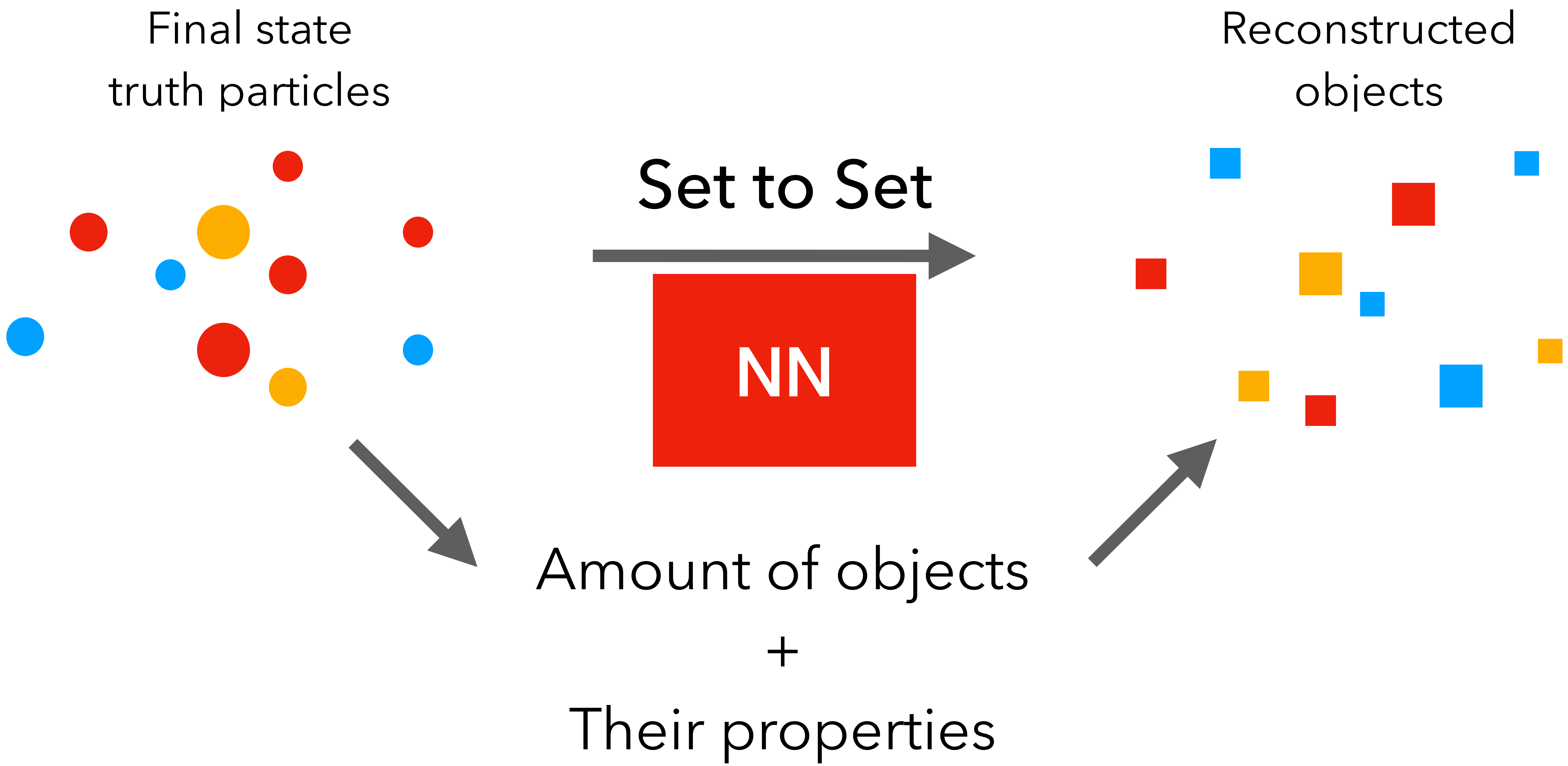
Slide from Etienne Dreyer

Data cycle of particle physics



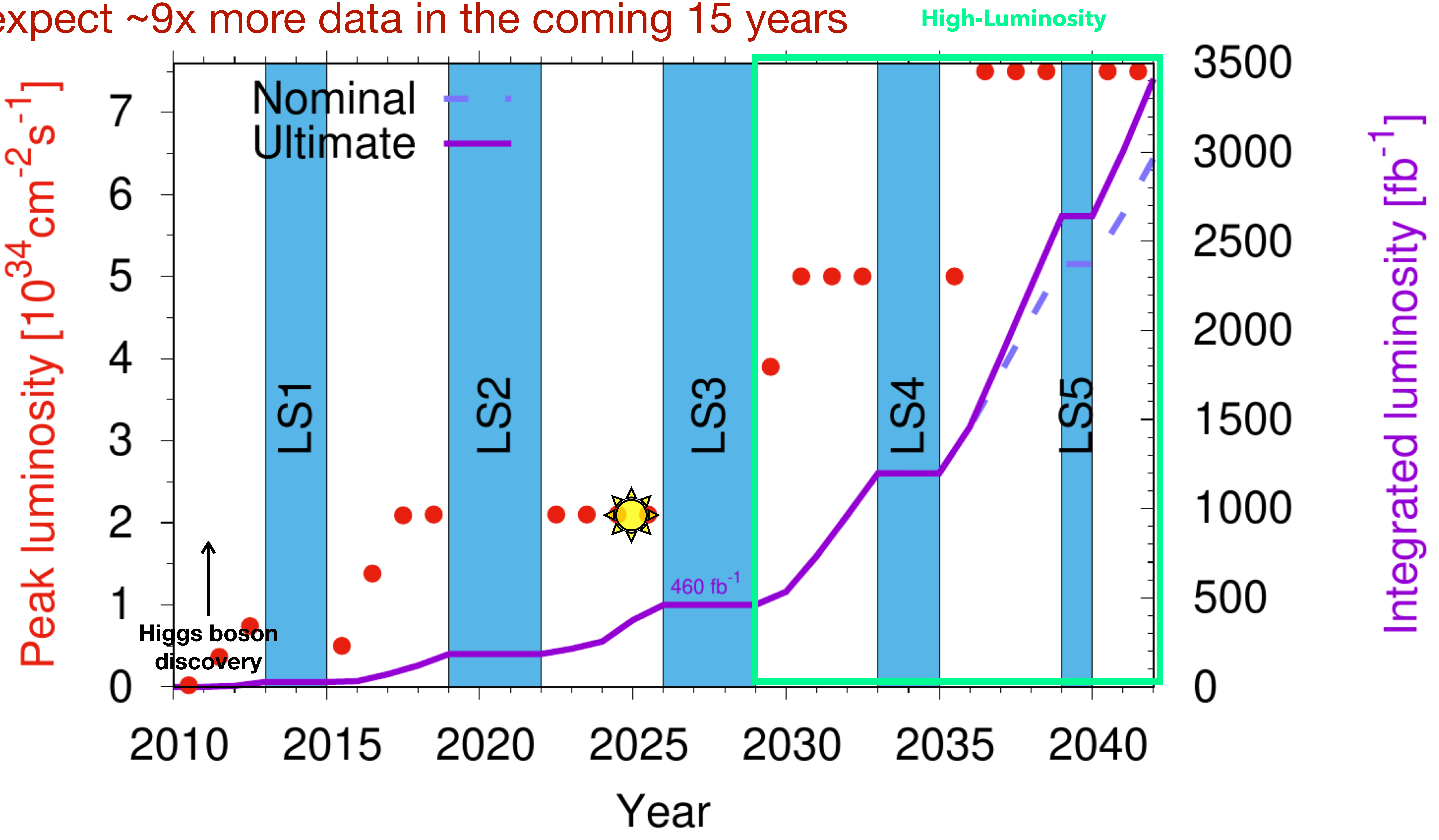
Slide from Etienne Dreyer

Problem to Solve



Why Fast Simulation

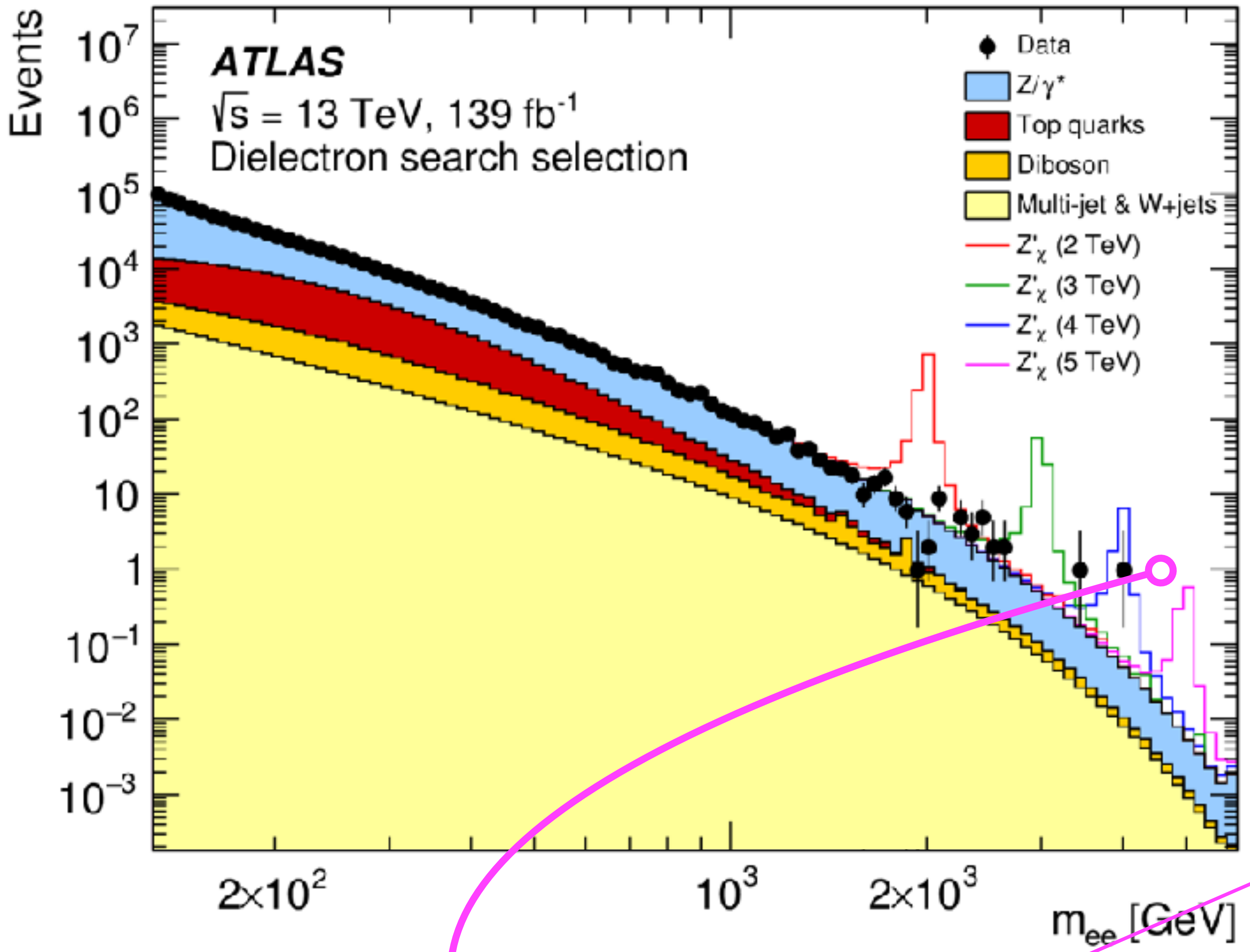
- We expect ~9x more data in the coming 15 years



Source: http://lhc-commissioning.web.cern.ch/schedule/images/2024/rampup_2023_YETS15weeks_NoIon_MDs_ULT.png

Slide from Etienne Dreyer

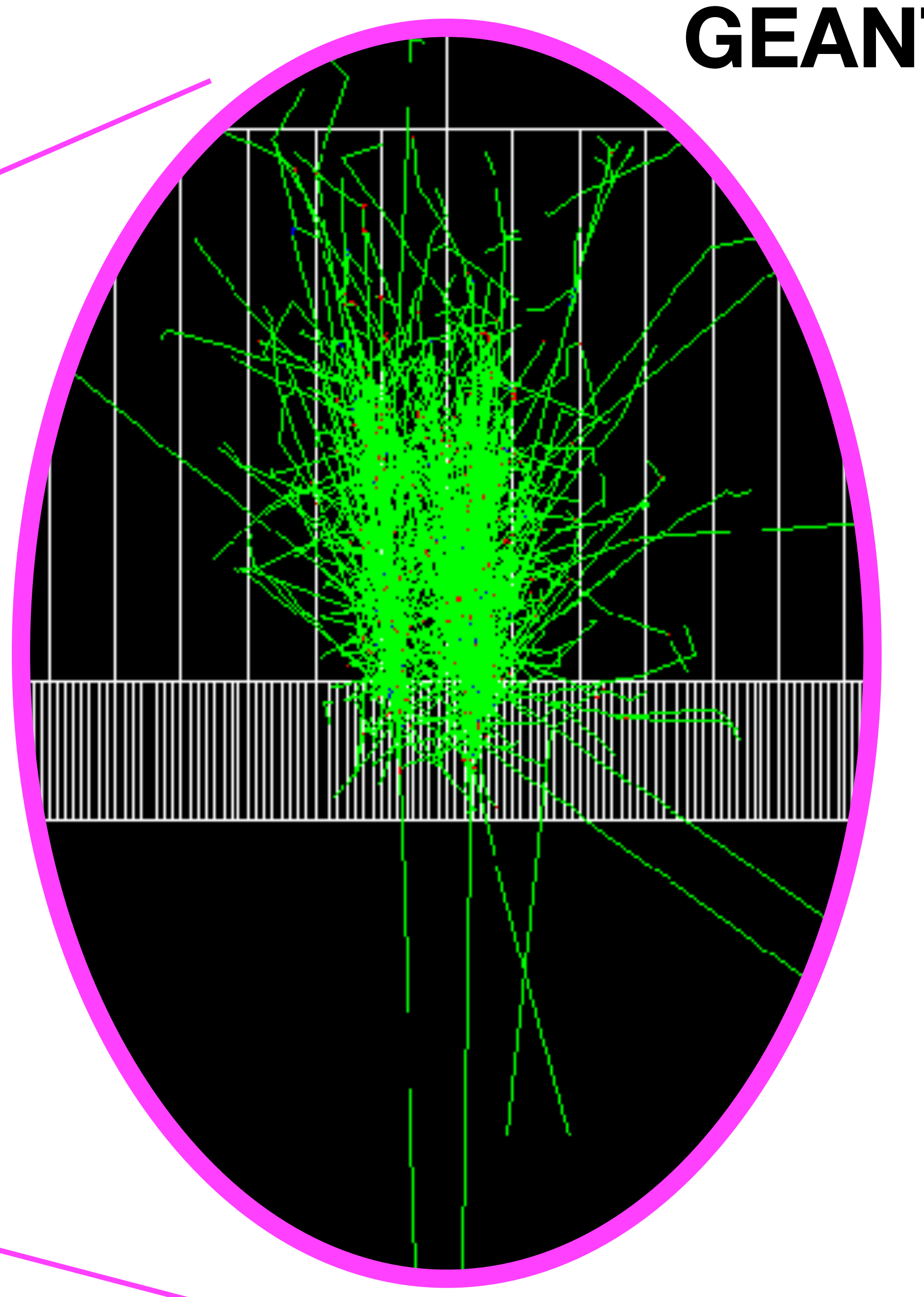
Phys. Lett. B 796 (2019) 68

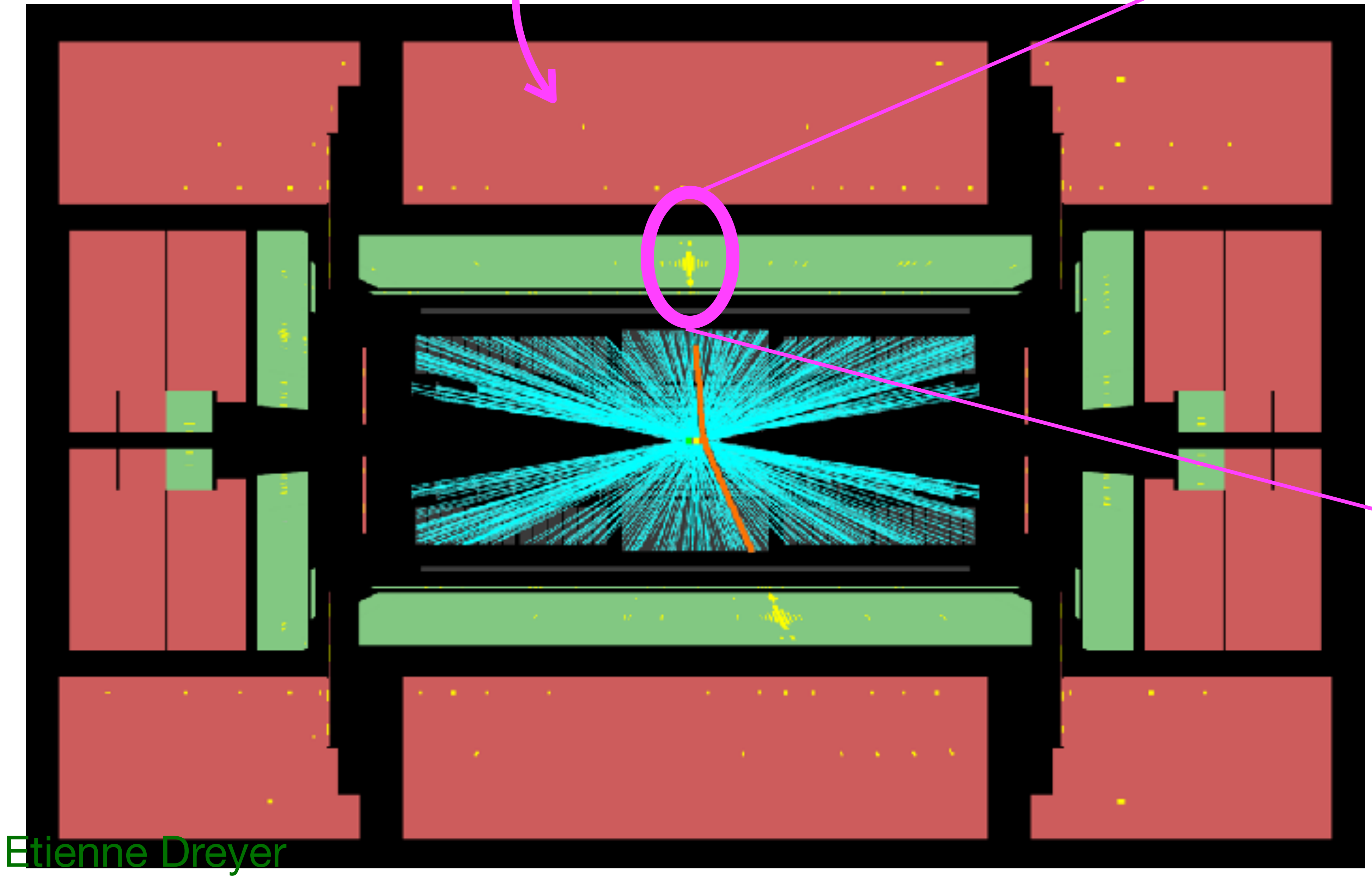


The simulation challenge

Testing hypotheses requires large amounts of simulated reference data (ideally 25-100x more than recorded data)

GEANT4 simulation





Fully simulating 1 event ~ $\mathcal{O}$(minutes)

⇒ $\mathcal{O}$(100M) events ~ O(1000) CPU years

Credit:
Michael Pitt

Slide from Etienne Dreyer

Why Fast Simulation

- **High Luminosity LHC**—>Orders of magnitude **more simulated events**
- **Future Colliders** will operate at higher energies and luminosities—>**1M Gbytes data/day**
- **Full Detector simulations** (GEANT) consumes >50% High Energy Phyiscs. They do not scale and **power and budget are limited**
- **High granularity detectors** will cost even more CPU power—>**Computing Bottleneck**
- **Precise physics** background modelling required for rare physics events, **current fast simulations cannot cope with** that
- **Designing future acceleerators** require **fast detector simulation response**
- GENERATIVE AI is fast and promising as we will see

Why Fast Simulation

- Fast simulations are not just a convenience-they are essential for the survival of HEP in the HL-LHC era and beyond.

- There is a fast simulation which also produces Particle Flow Candidates
Delphes (an hardcoded smearing based fast simulation):
Fast but less precise

DELPHES is our BASELINE

DELPHES 3, A modular framework for fast simulation of a generic collider experiment

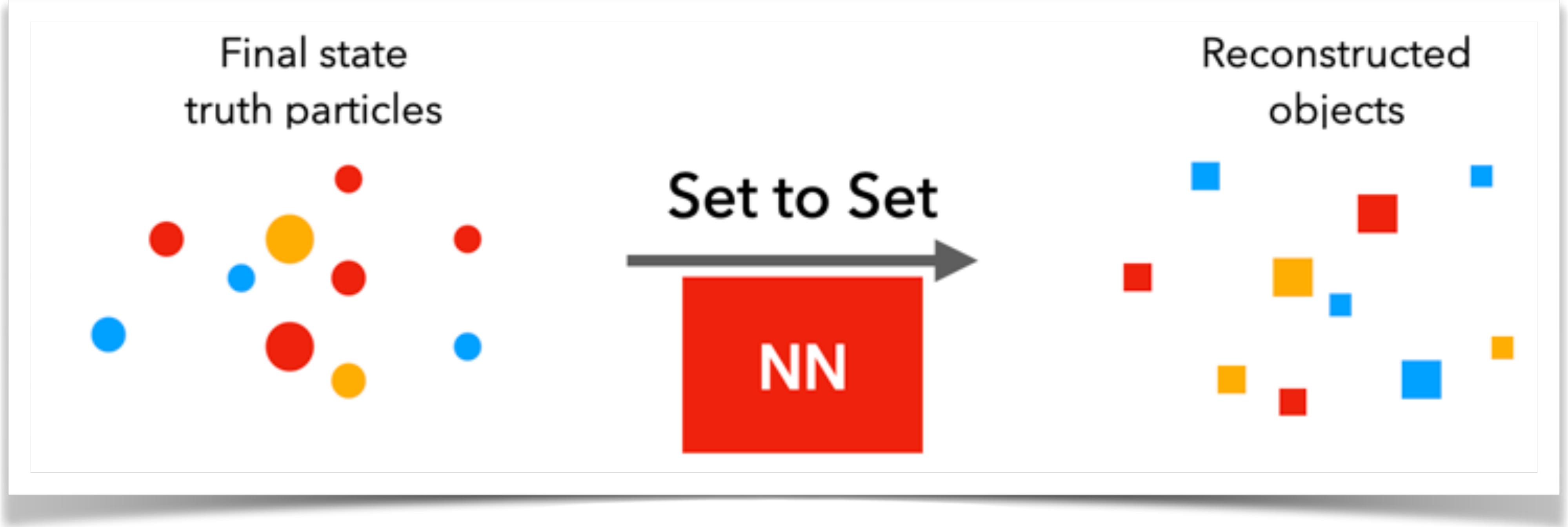
J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, M. Selvaggi

JHEP 02 (2014) 057. 3200 citations

- –>Need **accurate** and fast AI/ML and perhaps Quantum Ready Tools to overcome computational limits, unlock new physics and design detectors for 100-TeV frontier.

Introducing Parnassus

- Set to Set learned mapping:
  GEN truth particles → PF candidates (PFC)



- Trained on CMS full simulation data
- Two models: Diffusion (D) and **Flow Matching (F)**
- Outputs PF Candidate sets with kinematic features and class types

Parnassus

Particle-flow Neural Assisted Simulations

Dmitrii Kobylianskii, Vinicius Mikuni,
Benjamin Nachman, Nathalie Soybelman,
Nilotpal Kakati, Etienne Dreyer, Eilam Gross

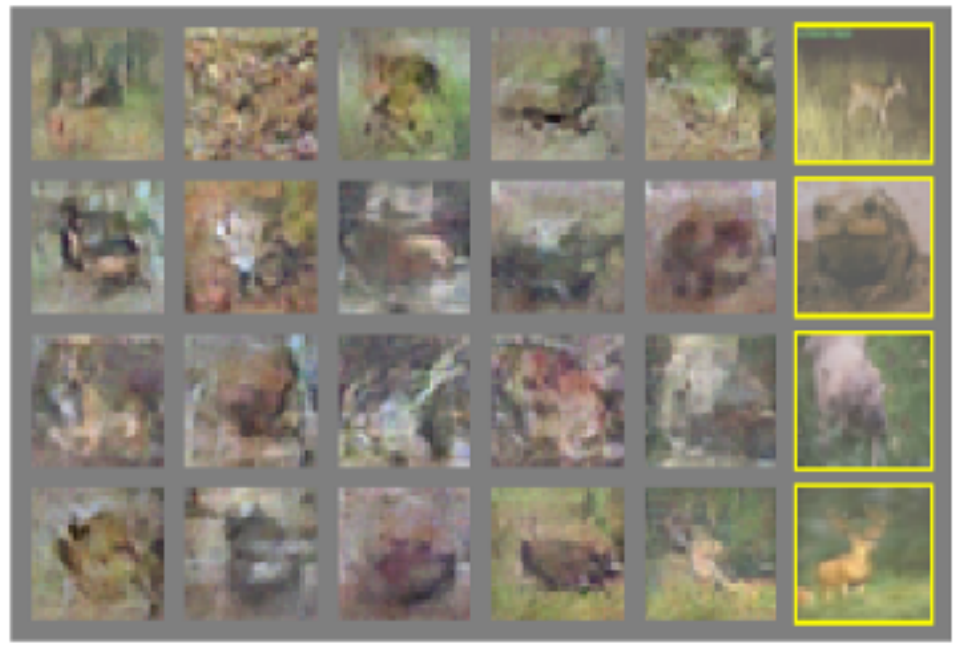
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

BERKELEY LAB

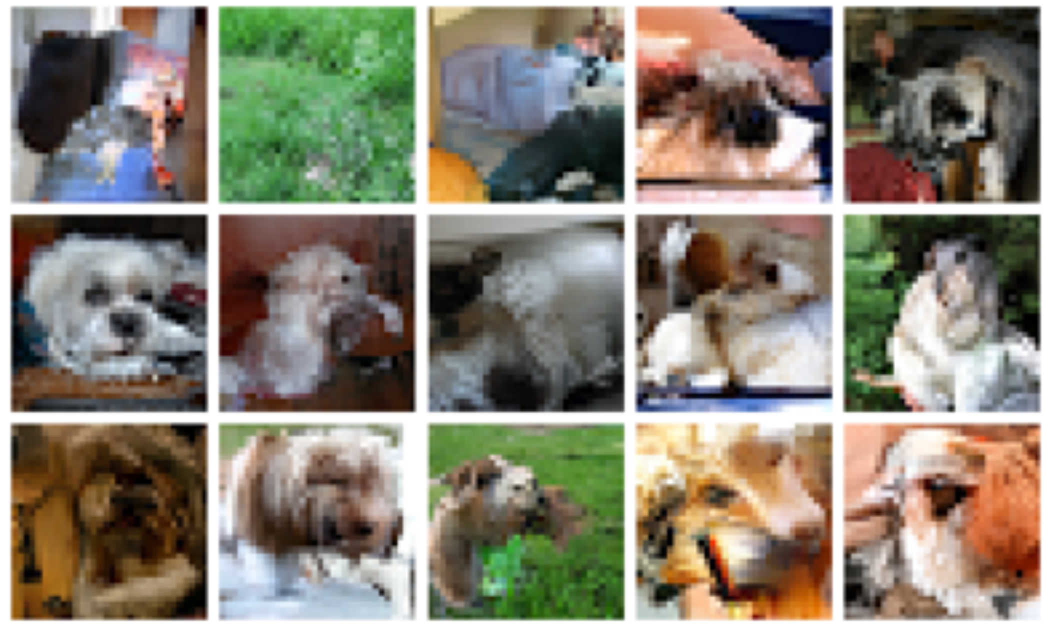
HOW?

VAE (2013)

GAN (2014)



PixelCNN (2016)



BigGAN (2018)

Imagen (2022)

Diffusion

Stable Diffusion 3 (2024)

Flow Matching

GPT 4o (2025)

Auto-Regressive

Stable Diffusion 3

Scaling Rectified Flow Transformers for High-Resolution Image Synthesis

Patrick Esser * Sumith Kulal Andreas Blattmann Rahim Entezari Jonas Müller Harry Saini Yam Levi
Dominik Lorenz Axel Sauer Frederic Boesel Dustin Podell Tim Dockhorn Zion English
Kyle Lacey Alex Goodwin Yannik Marek Robin Rombach *
Stability AI

arXiv:2403.03206v1 [cs.CV] 5 Mar 2024

Figure 1. High-resolution samples from our 8B rectified flow model, showcasing its capabilities in typography, precise prompt following and spatial reasoning, attention to fine details, and high image quality across a wide variety of styles.

Abstract

Diffusion models create data from noise by inverting the forward paths of data towards noise and have emerged as a powerful generative modeling technique for high-dimensional, perceptual data such as images and videos. Rectified flow is a ra- ... strate the superior performance of this approach compared to established diffusion formulations for high-resolution text-to-image synthesis. Additionally, we present a novel transformer-based architecture for text-to-image generation that uses separate weights for the two modalities and en-

Flow Matching

Text prompt:

But how?

NN

A bird standing upon the waters

Stable Diffusion 3.5

Conditional generation

Text prompt:

```
A bird standing
upon the waters
```

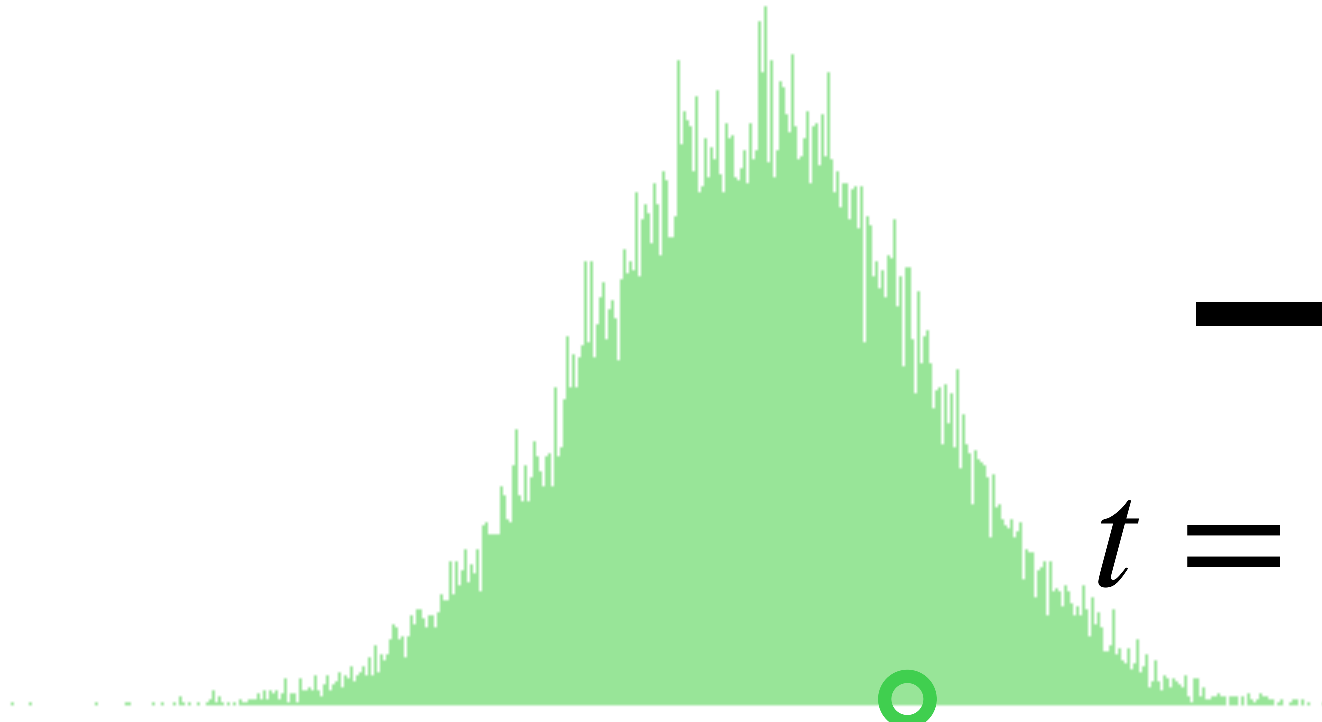
(Gaussian Noise)

$x_0 \sim q = \mathcal{N}(0,1)$

c

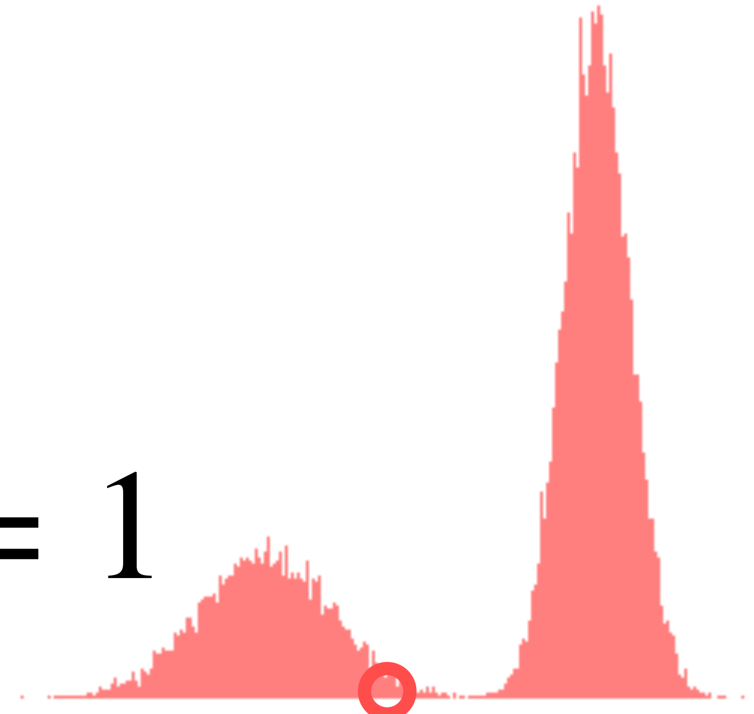
(Data)

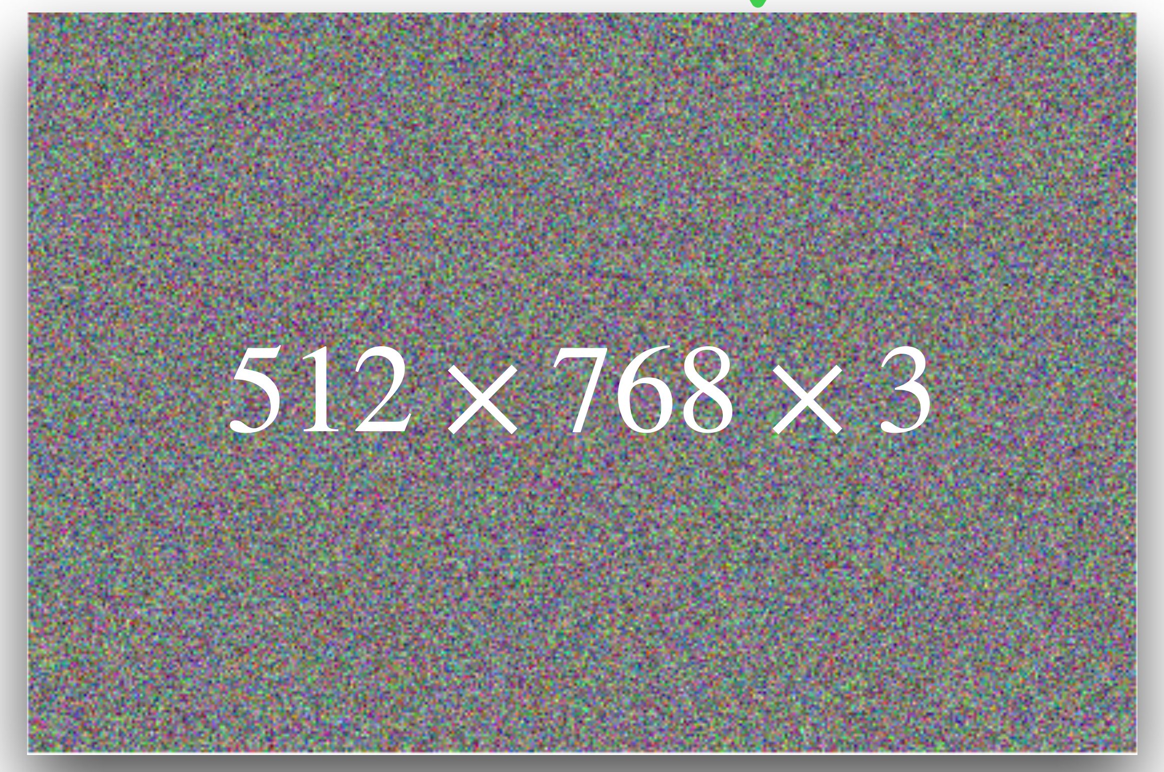
$x_1 \sim p_{\text{data}}$



NN $\longrightarrow \cdots \longrightarrow$

$t = 0$ $\quad t + dt$ $\quad t = 1$





Slide from Etienne Dreyer

But how?

"out of distribution"

NN

Stable Diffusion 3.5

Text prompt:

```
A small, exquisite bird
with vibrant pink
feathers, adorned with
intricate, sparkling
jewels and pearls,
Perched on a branch of a
cherry tree, wearing an
ornate golden crown
encrusted with a
kaleidoscope of colorful
gemstones, is set against
a dreamy, blurred
background of soft
purples and blues,
reminiscent of a
fantastical sky.
including a "cute king"
label at the top, with a
touch of ethereal soft
focus and heavenly
sunshine, as if divine
beams are illuminating
the clouds.
```

Leveraging Stable Diffusion Techniques?

Feature/Principle	SD3.x	Parnassus
Output	Images (pixels)	Particle Flow Candidates - PFC
Conditioning	Text Prompts	Generator-level-particles (truth particles)
Generation	text—>image	2 stages: Event, PFC set (truth)—>set (PFC)
Generative Principle	Flow Matching (Rectified Flow)	Flow Matching (Rectified Flow)
Probability Path	Rectified Flow, straight trajectory	Rectified Flow, straight trajectory
Sampling Efficiency	ODE based	ODE based
Architecture	Transformer based	Transformer based
Generalization	Unseen Text/Image Prompts	Unseen Physics Processes

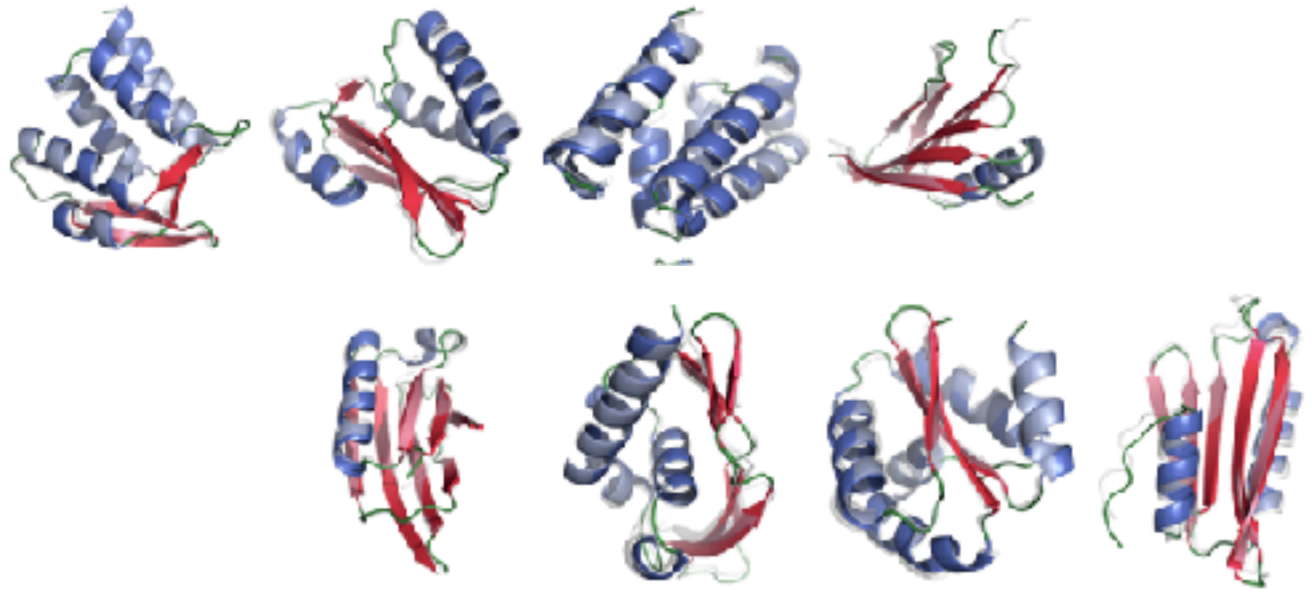
Leveraging Stable Diffusion Techniques?

Feature/Principle	SD3.x	Parnassus
Output	Images (pixels)	Particle Flow Candidates - PFC
Conditioning	Text Prompts	Generator-level-particles (truth particles)
Generation	text—>image	truth—>event Level truth+event—>PFC
[illegible]	[illegible]	[illegible]
Probability Path	Rectified Flow, straight trajectory	Rectified Flow, straight trajectory
Sampling Efficiency	ODE based	ODE based
Architecture	Transformer based	Transformer based
Generalization	Unseen Text/Image Prompts	Unseen Phyiscs Processes

Flow Matching

Flow Matching at SCALE

Text-2-Video
MovieGen, Meta



Protein Generation
Huguet el a. 24

Text-2-Image
Stable Diffusion 3

(c) Yaron Lipman, Meta, WIS

WHAT IS FLOW MATCHING?

A scalable method to train flow generative models.

HOW DOES IT WORK?

Train by regressing a velocity, sample by following the velocity

slide by Yaron Lipman

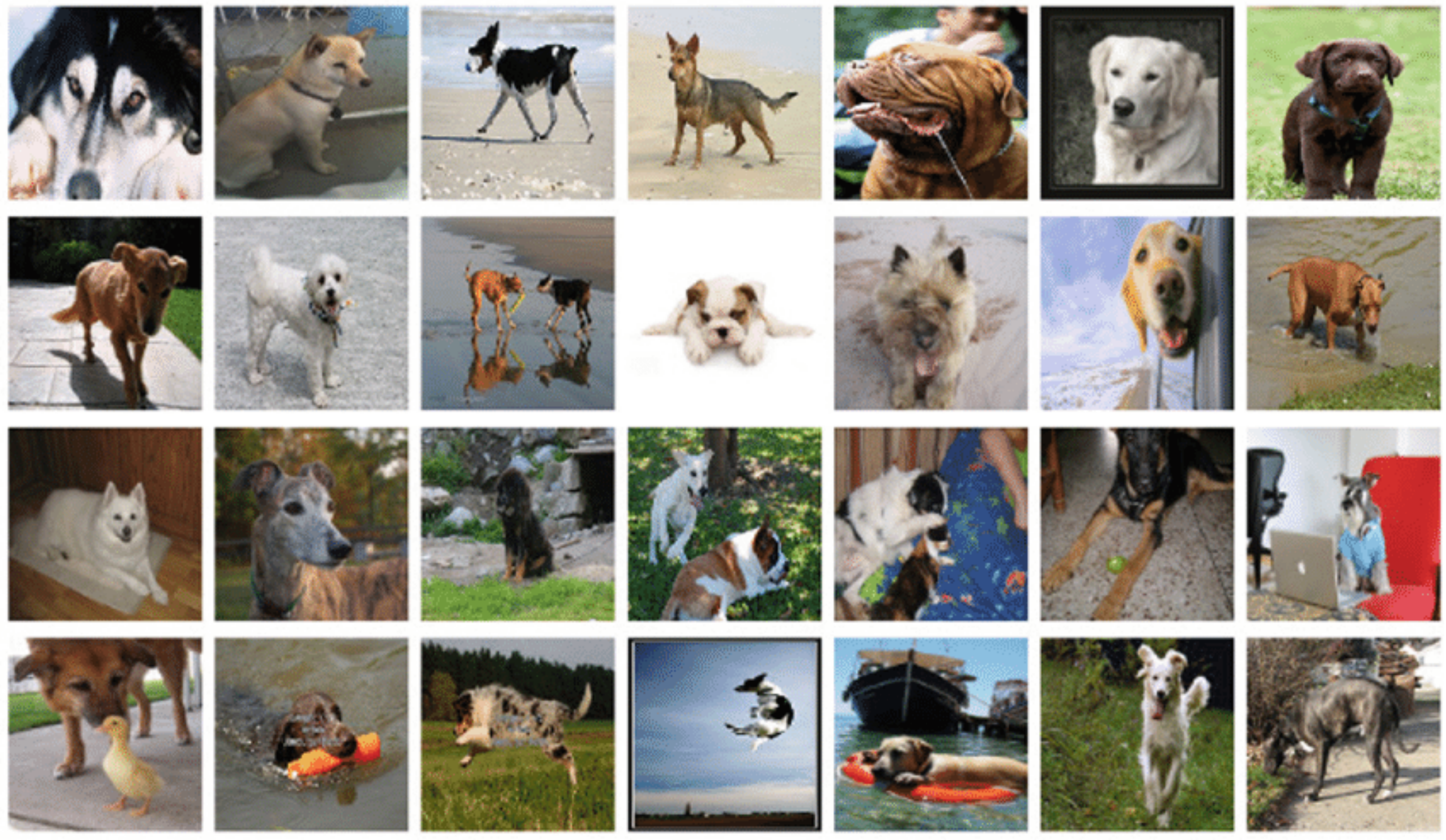
The Generative Modeling Problem

$\mathbb{R}^d$

The distribution of dogs p_{data} is unknown
but we have training data from which we can sample

$x \in \mathbb{R}^{HxWx3}$

x



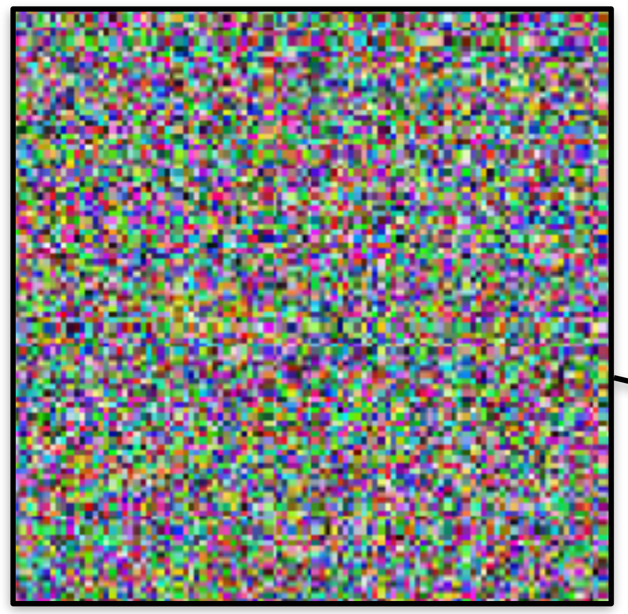
Figures by Yaron Lipman

The Generative Modeling Problem

$\mathbb{R}^d$

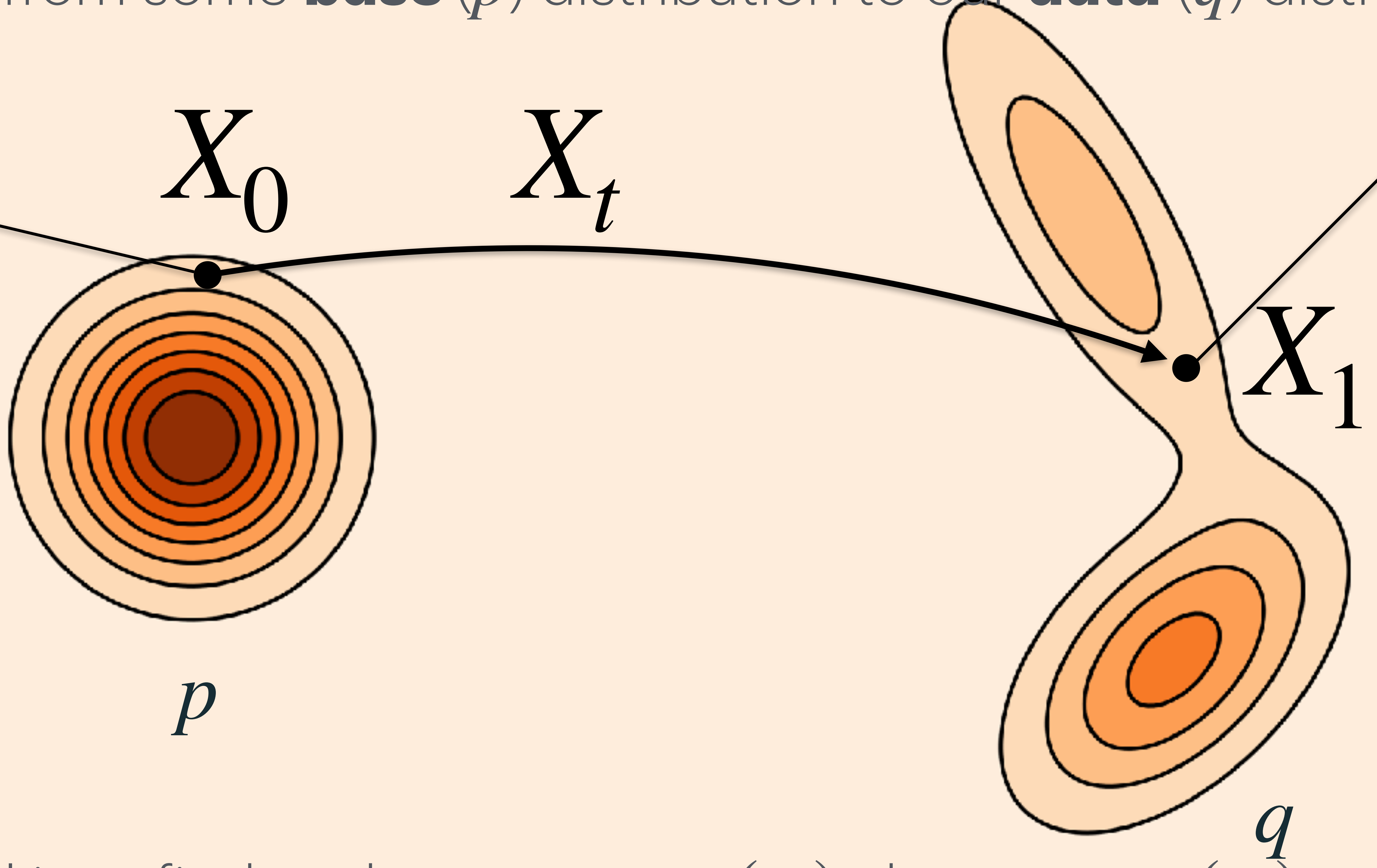
A generative model converts samples from an initial distribution into samples from an unknown Data distribution

Transfer from some **base** (p) distribution to our **data** (q) distribution



$X_0 \sim p$

Noise~Gaussian



$X_1 \sim q$

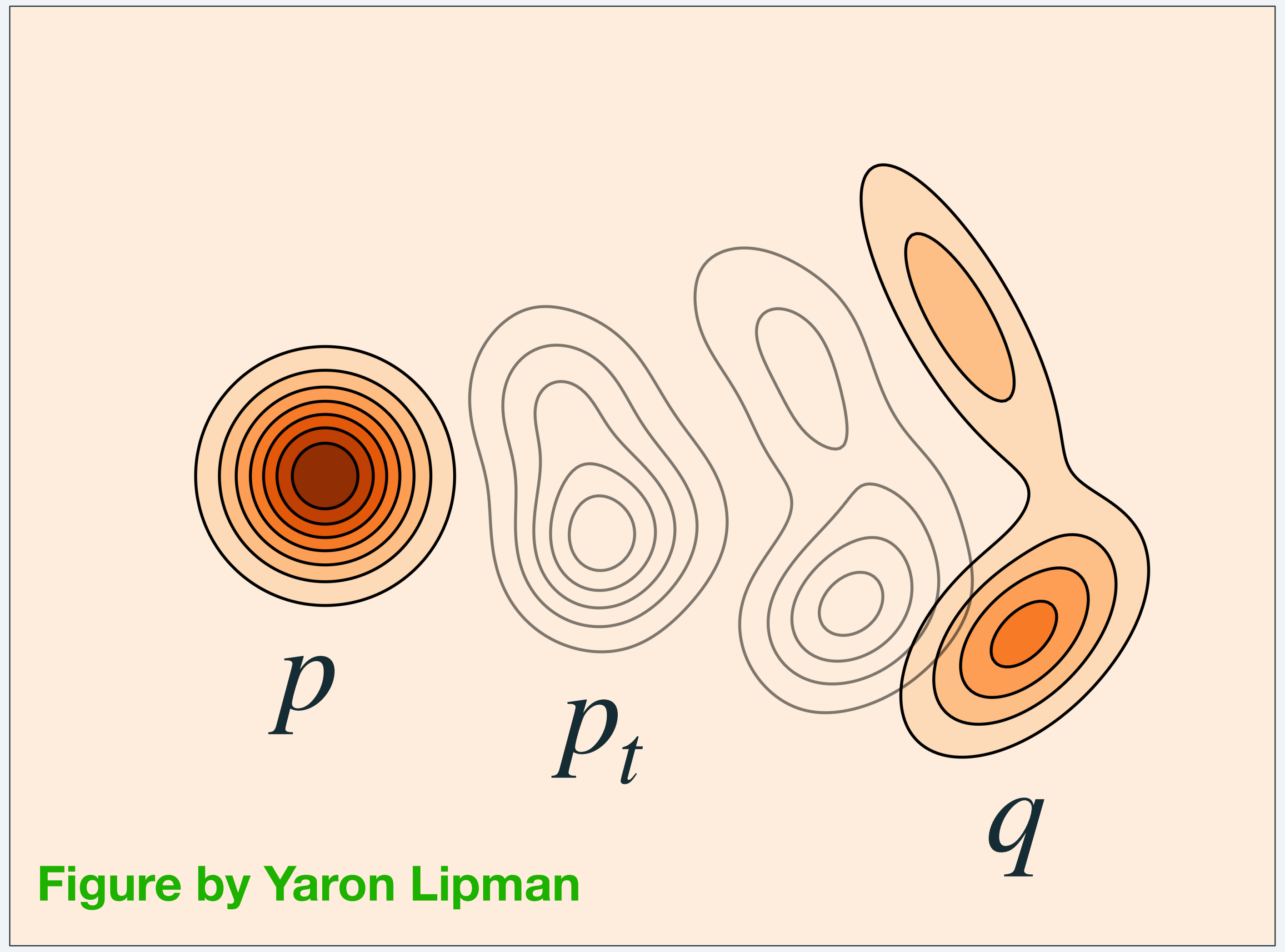
The goal is to find such ψ_t : $x_t = \psi_t(x_0)$, that $x_1 = \psi_1(x_0) \sim q$

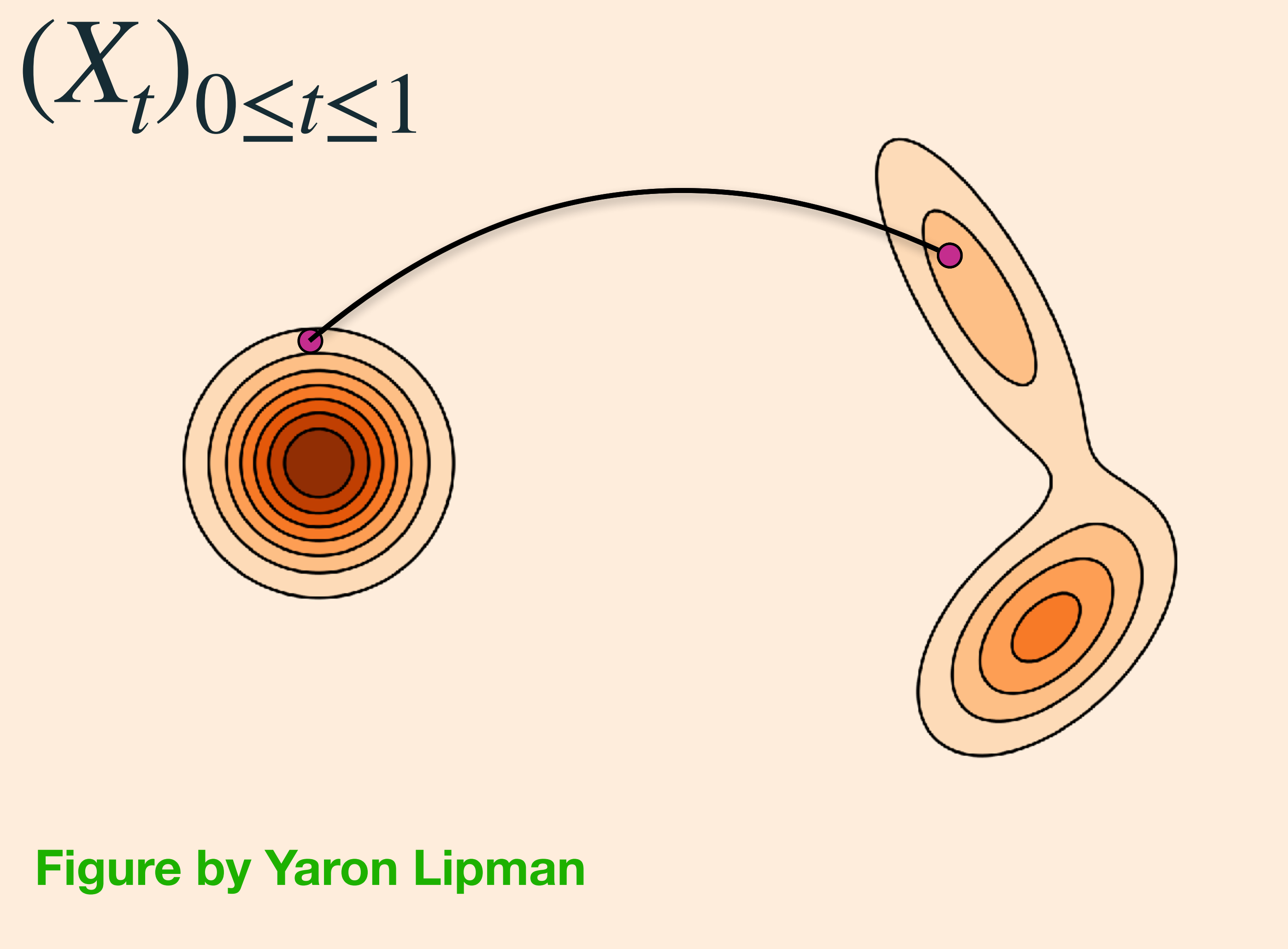
This ψ_t we call **flow**

Figures by Yaron Lipman

Marginal probability path and Flows

$$X_t \sim p_t$$

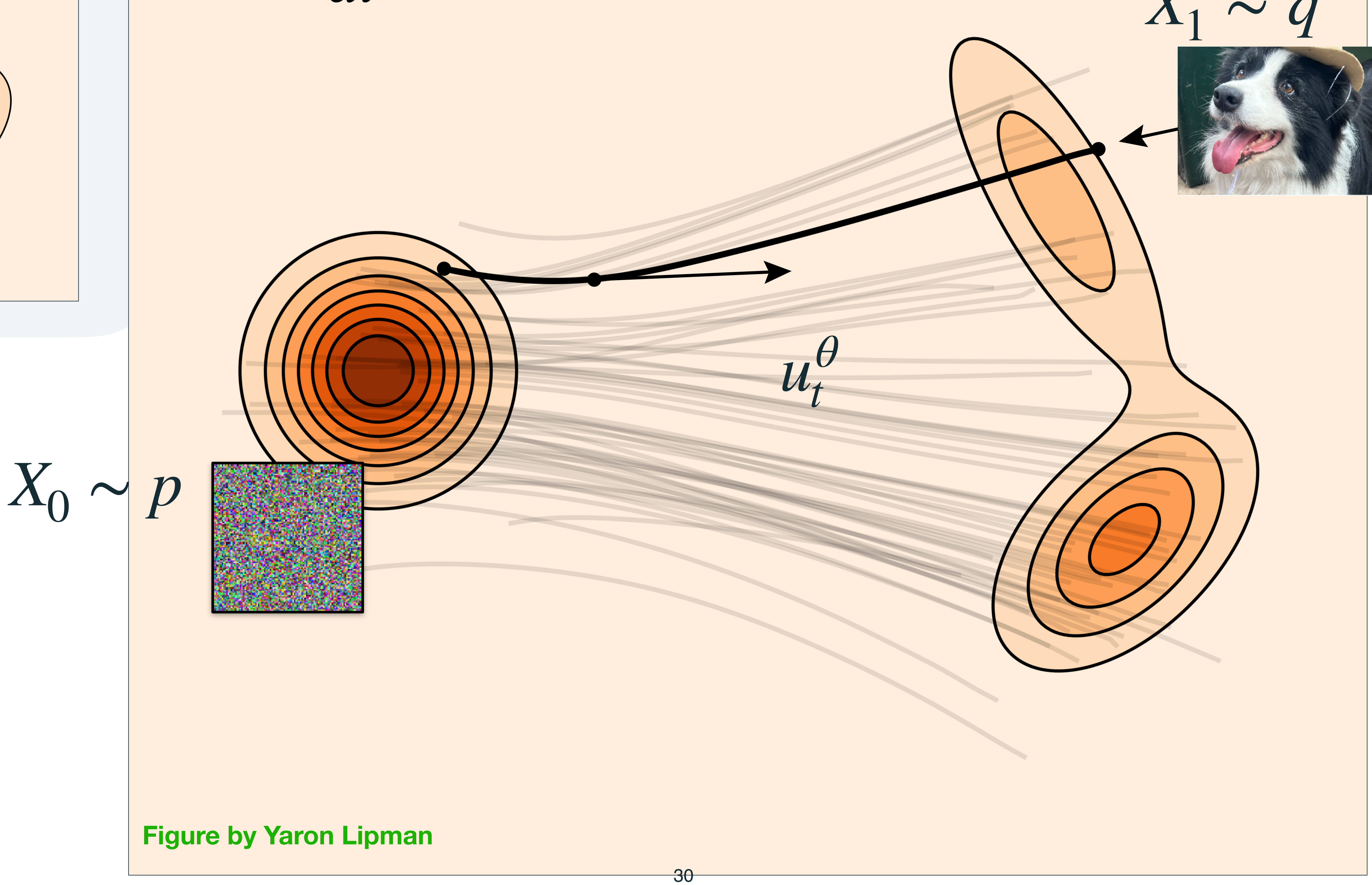




Flow

The flow defines a trajectory $\psi_t(x_0)$ that solves a differential equation

$$\frac{d}{dt} x_t = u_t(x_t)$$



Marginal probability path and Flows

$$X_t \sim p_t$$

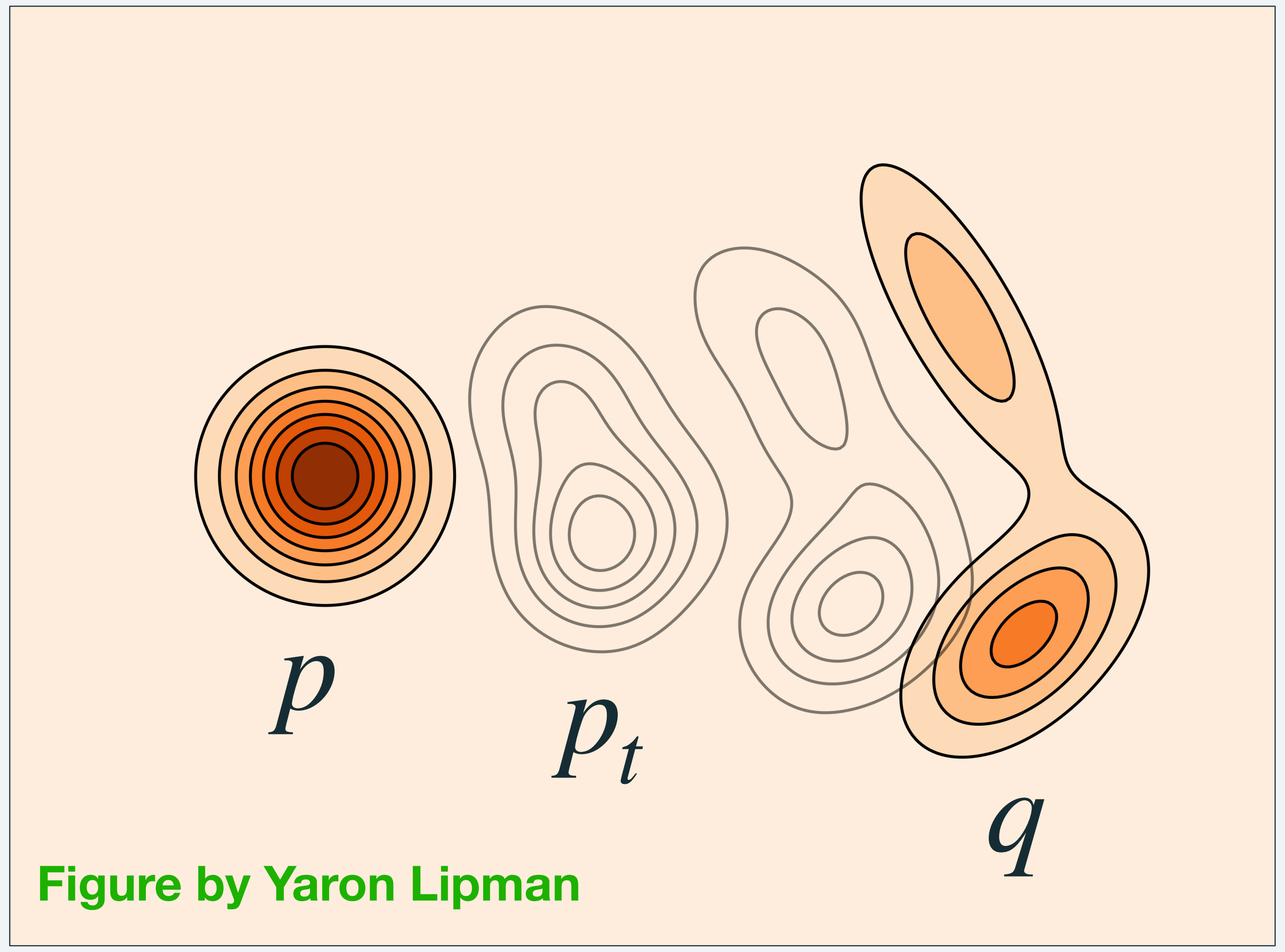


Figure by Yaron Lipman

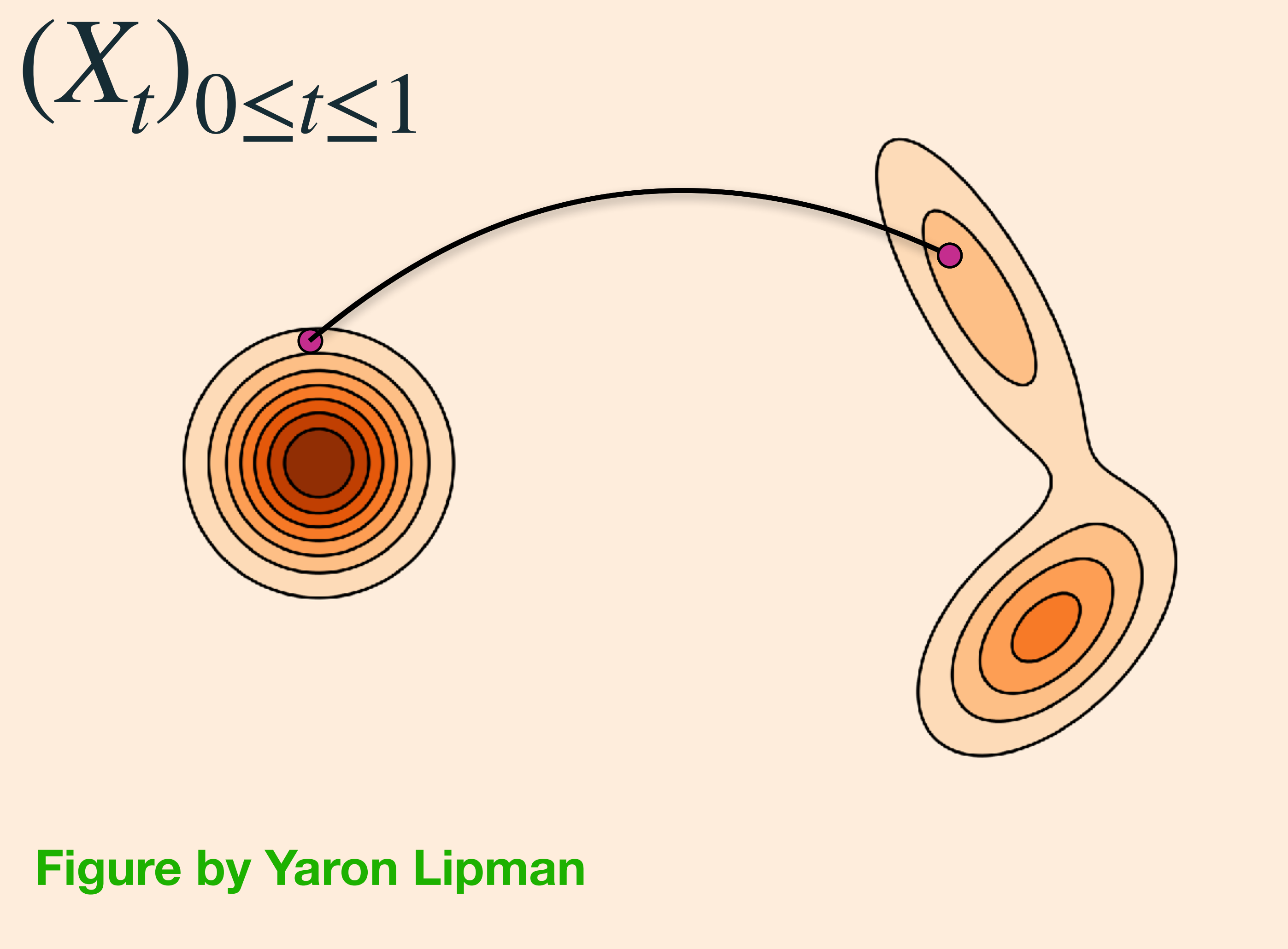
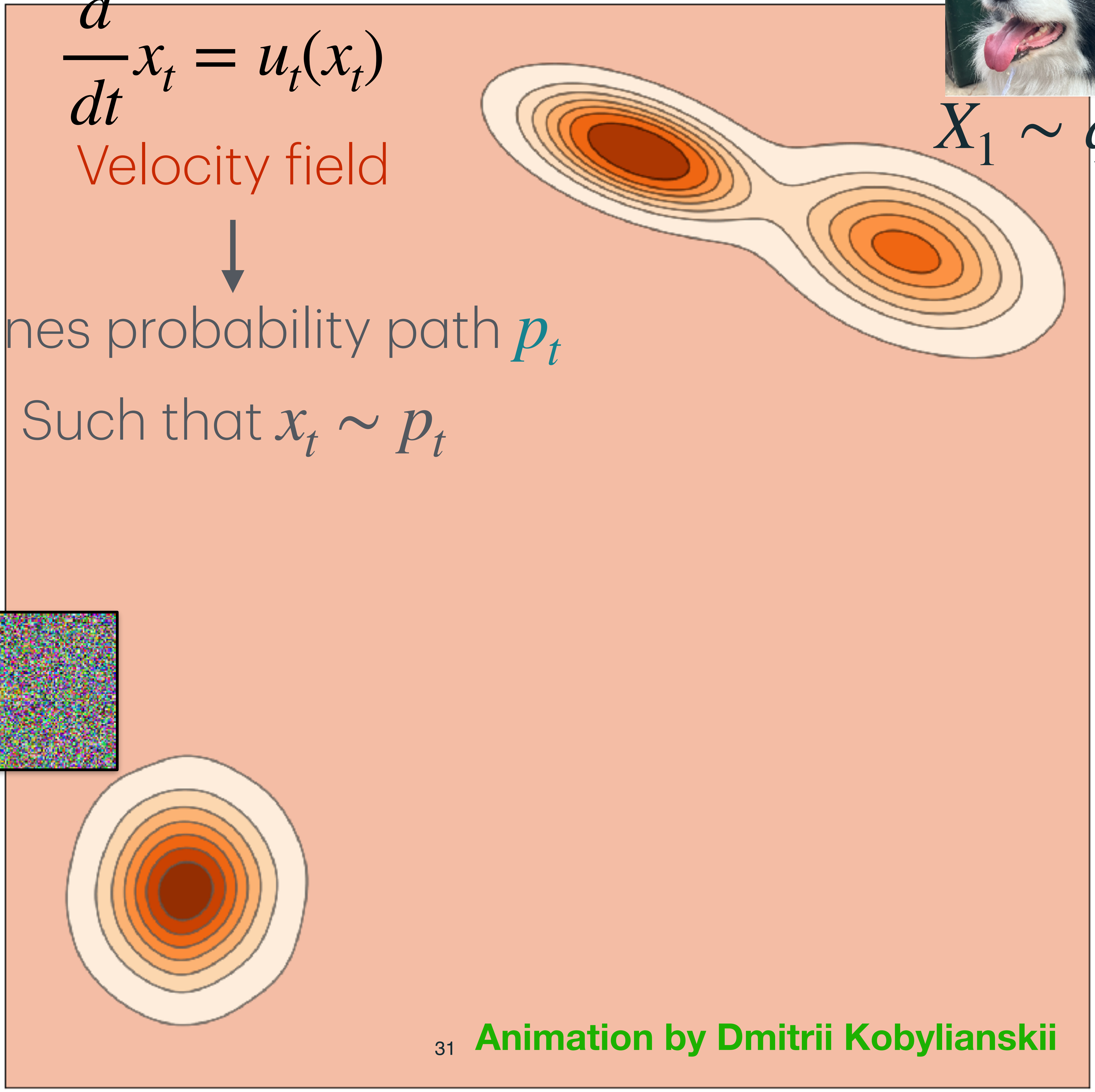


Figure by Yaron Lipman

Flow

The flow defines a trajectory $\psi_t(x_t)$ that solves a differential equation



$$\frac{d}{dt} x_t = u_t(x_t)$$

Velocity field

↓

Defines probability path p_t

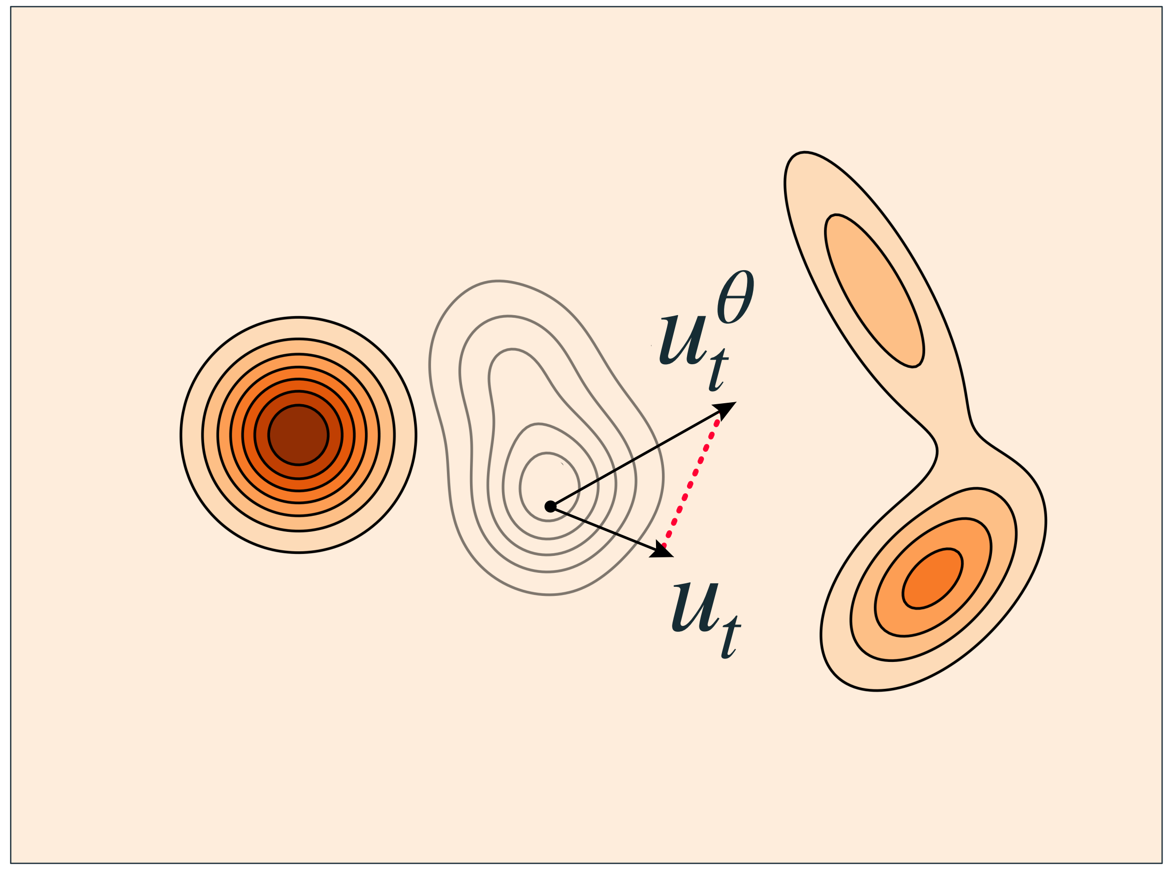
Such that $x_t \sim p_t$

$X_1 \sim q$

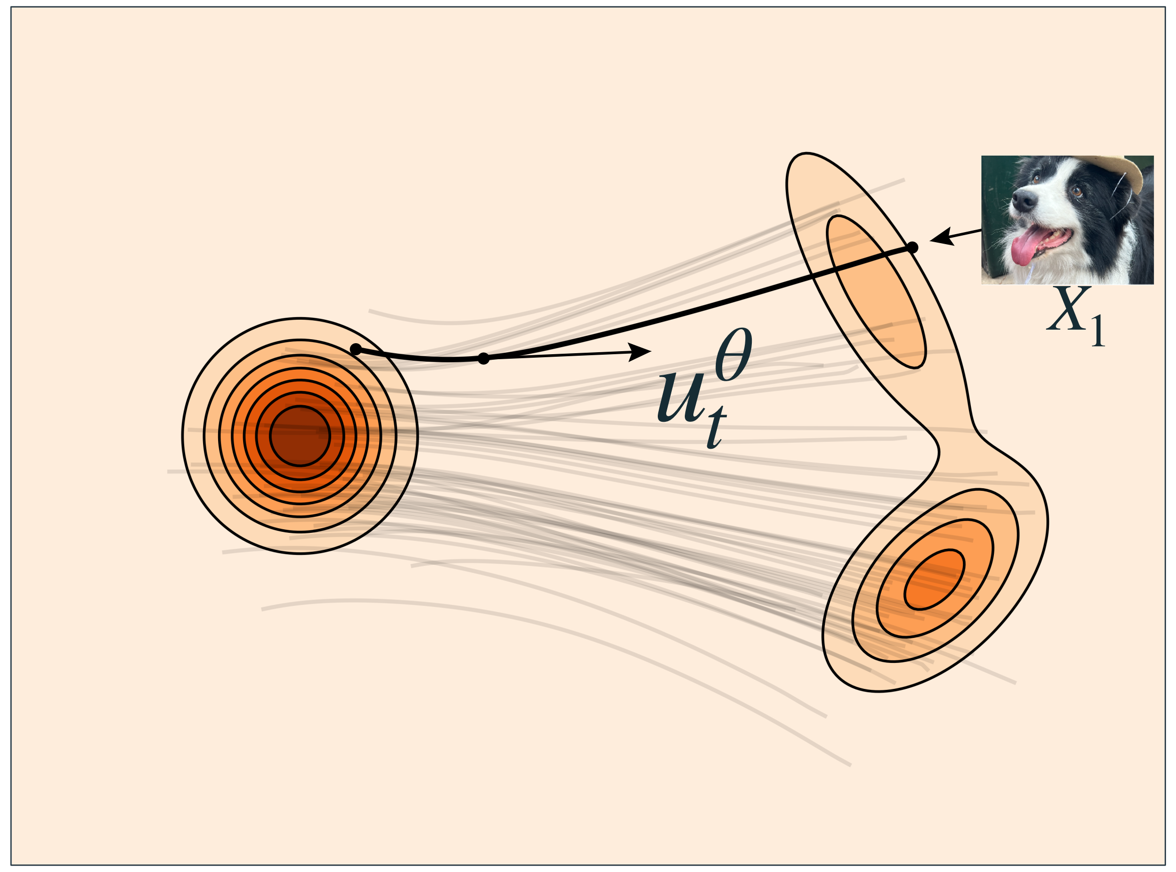
$X_0 \sim p_0$

Animation by Dmitrii Kobylianskii

Flow Matching



Train a velocity
generating p_t with
$p_0 = \mathcal{N}(0,1)$ and $p_1 = p_{data} = q$



Sample
from $X_0 \sim \mathcal{N}(0,1)$

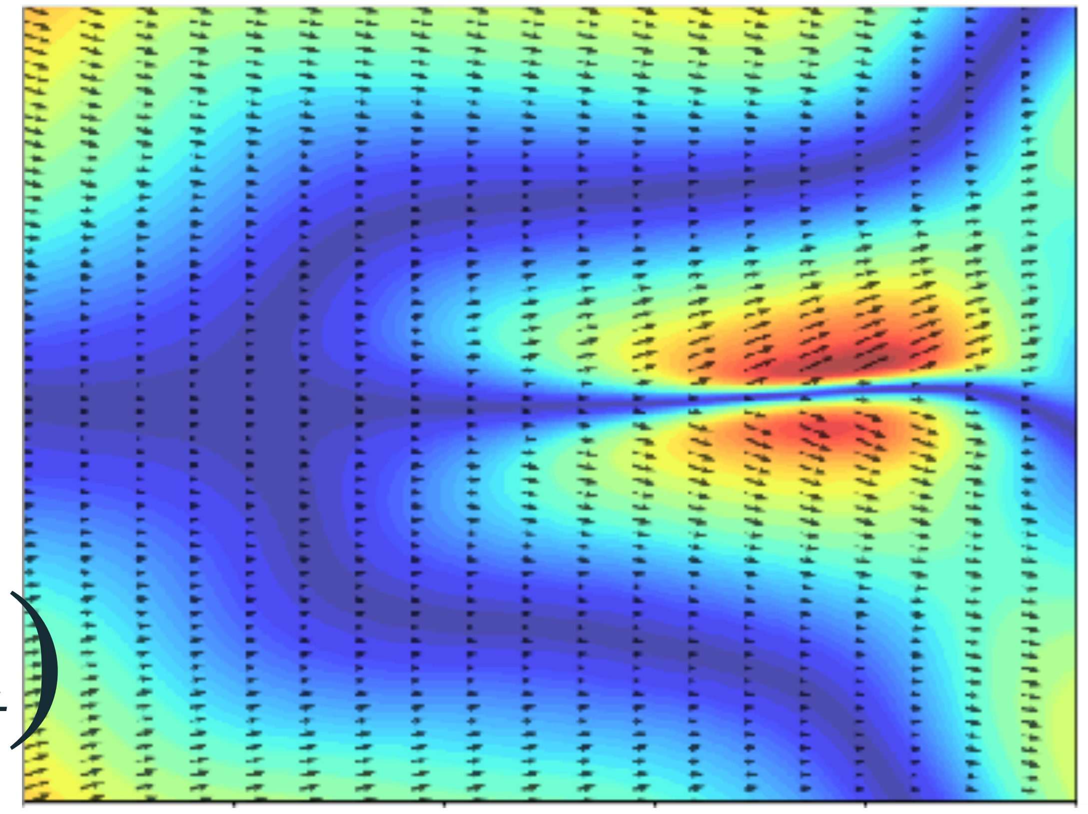
slide by Yaron Lipman

Flow Match Loss

- We need to minimize the marginal velocity loss $u_t(X_t)$

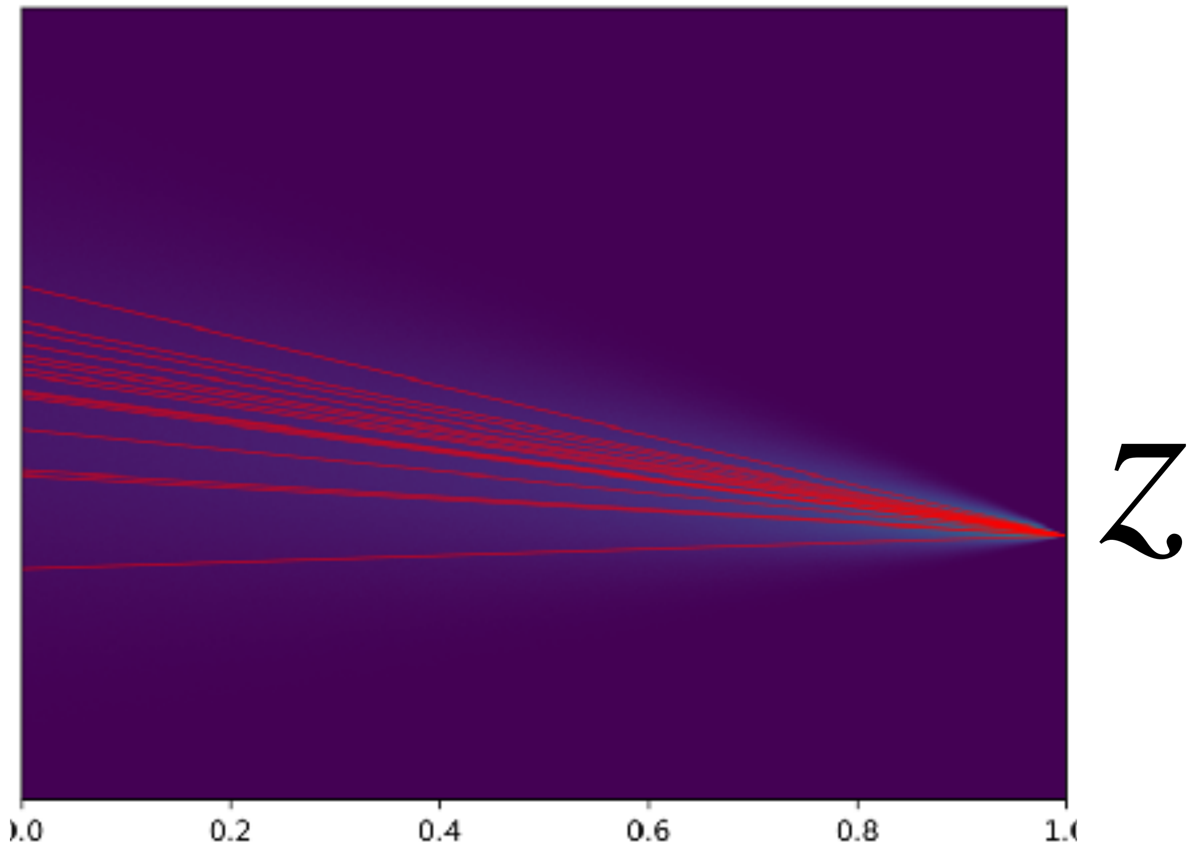
$$\mathscr{L}_{\text{FM}}(\theta) = \mathbb{E}_{X_t \sim p_t} \left\| u_t^\theta(X_t) - u_t(X_t) \right\|^2$$

- But we do not know to calculate the marginal velocity target $u_t(X_t)$



- Define the conditional Flow Match loss

$$\mathscr{L}_{\text{CFM}}(\theta) = \mathbb{E}_{X_t \sim p_{t|z}(x|z)} \left\| u_t^\theta(X_t) - u_t(X_t \,|\, X_1 = z) \right\|^2$$



Losses are equivalent

- $$\nabla_\theta \mathscr{L}_{\text{FM}}(\theta) = \nabla_\theta \mathscr{L}_{\text{CFM}}(\theta)$$

- Minimizing $\mathscr{L}_{\text{CFM}}(\theta)$ —> minimizing $\mathscr{L}_{\text{FM}}(\theta)$

Figures by Dmitrii Kobylianskii

1D Conditional Flow Match

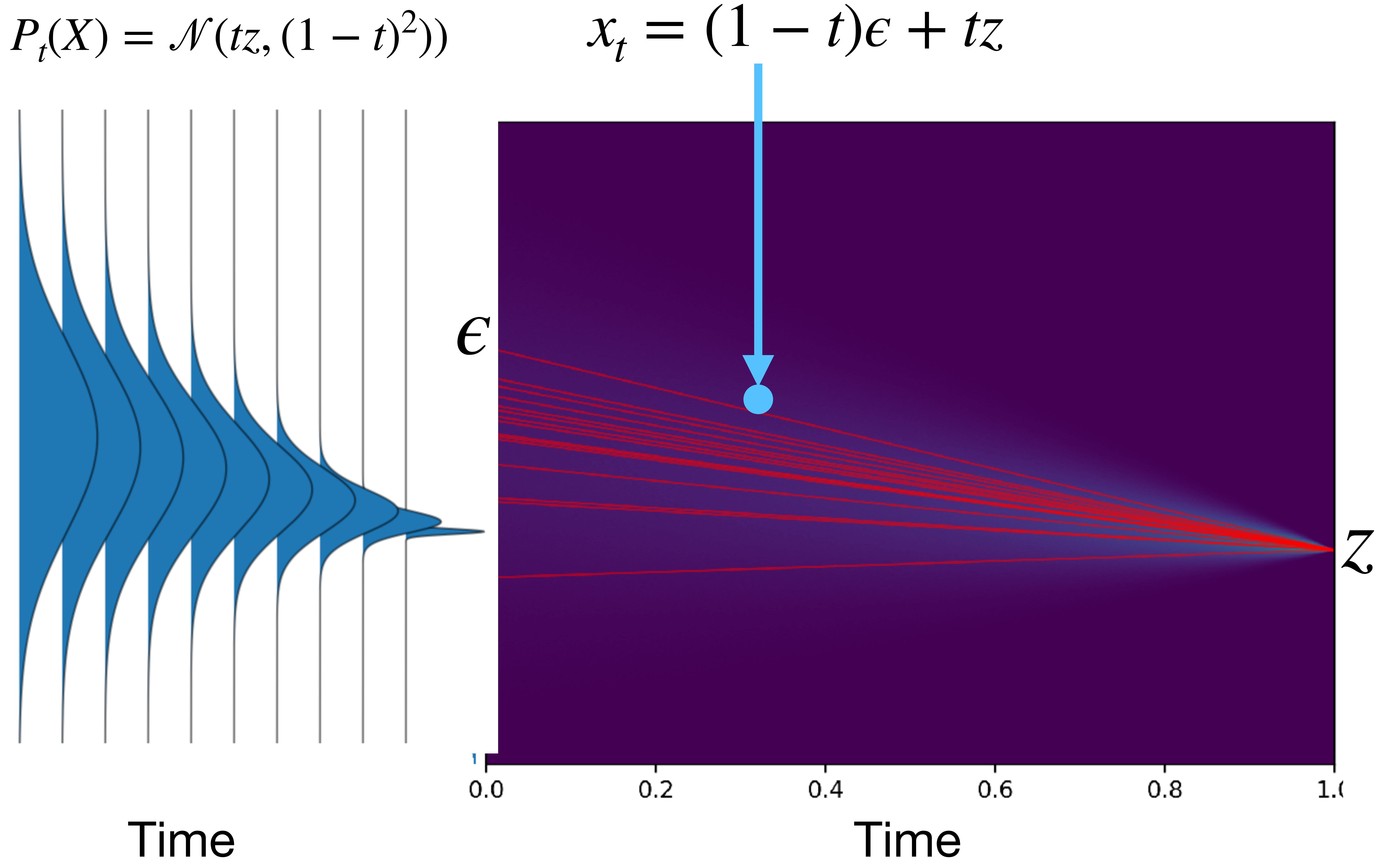
Lines are x_t sampled during training
So for one pair of (x_0, x_1)

Set $X_1 = z \in$ **Data Set**
$P_{init}(X) = \mathcal{N}(0,1)$
Noise $X_0 = \epsilon \sim \mathcal{N}(0, I_d)$

$t \sim Unif[0,1]$
$x_t = (1-t)\epsilon + tz$

$u_t(X) = z - \epsilon$

$P_t(X) = \mathcal{N}(tz, (1-t)^2))$

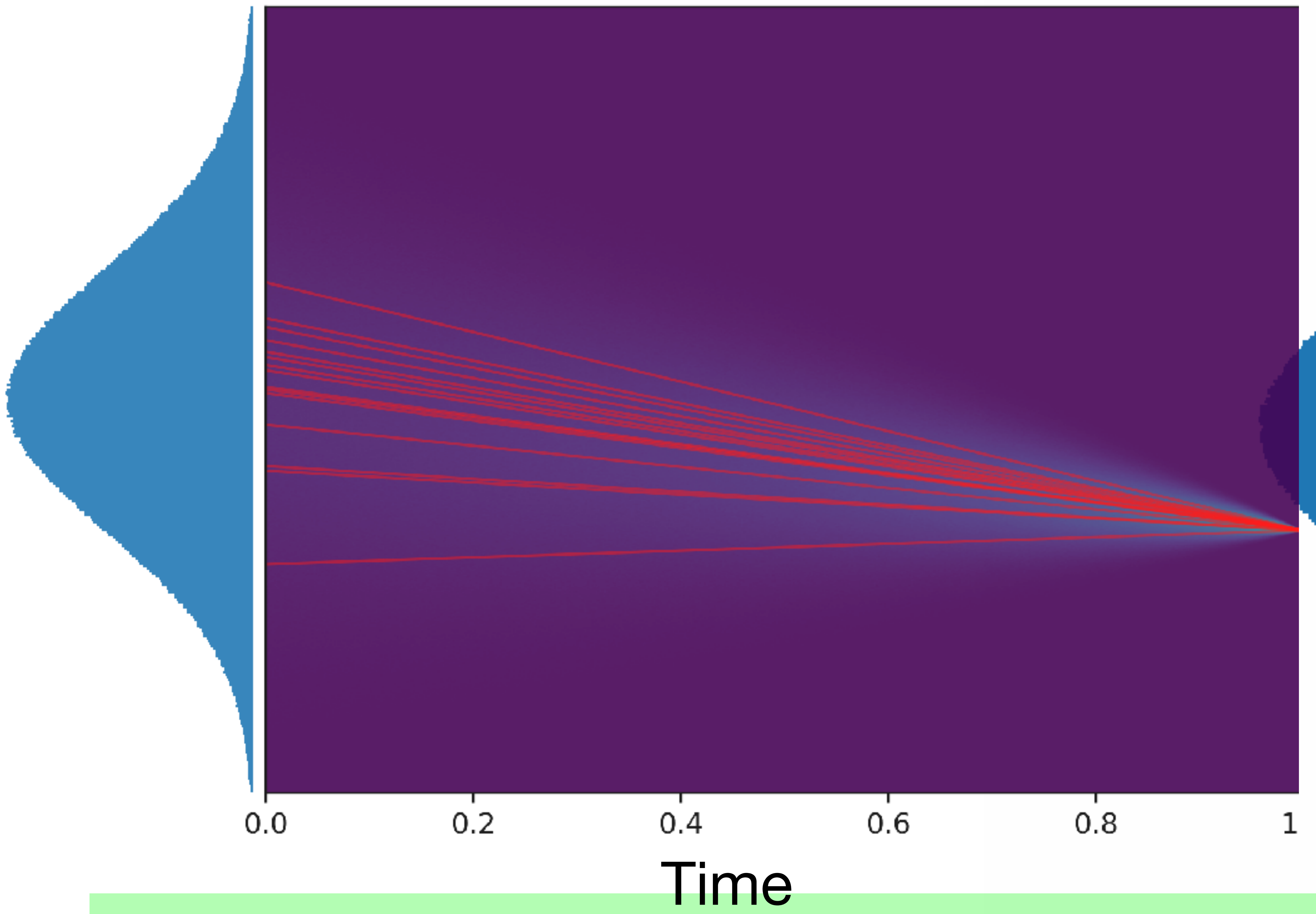


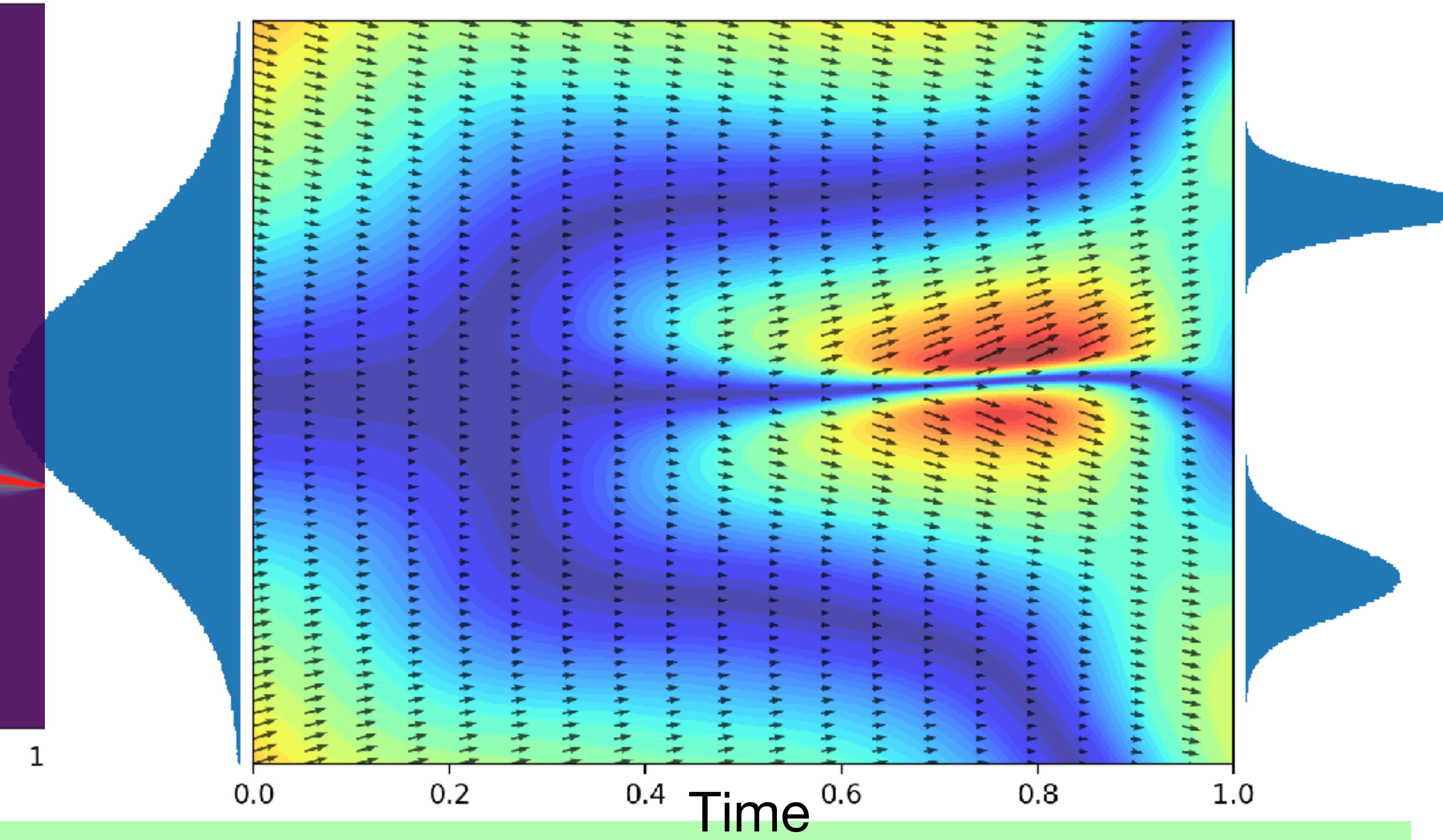
Figures by Dmitrii Kobylianskii

1D Conditional Flow Match

$$P_t(X) = \mathcal{N}(tz, (1-t)^2))$$

$$x_t = (1-t)\epsilon + tz$$

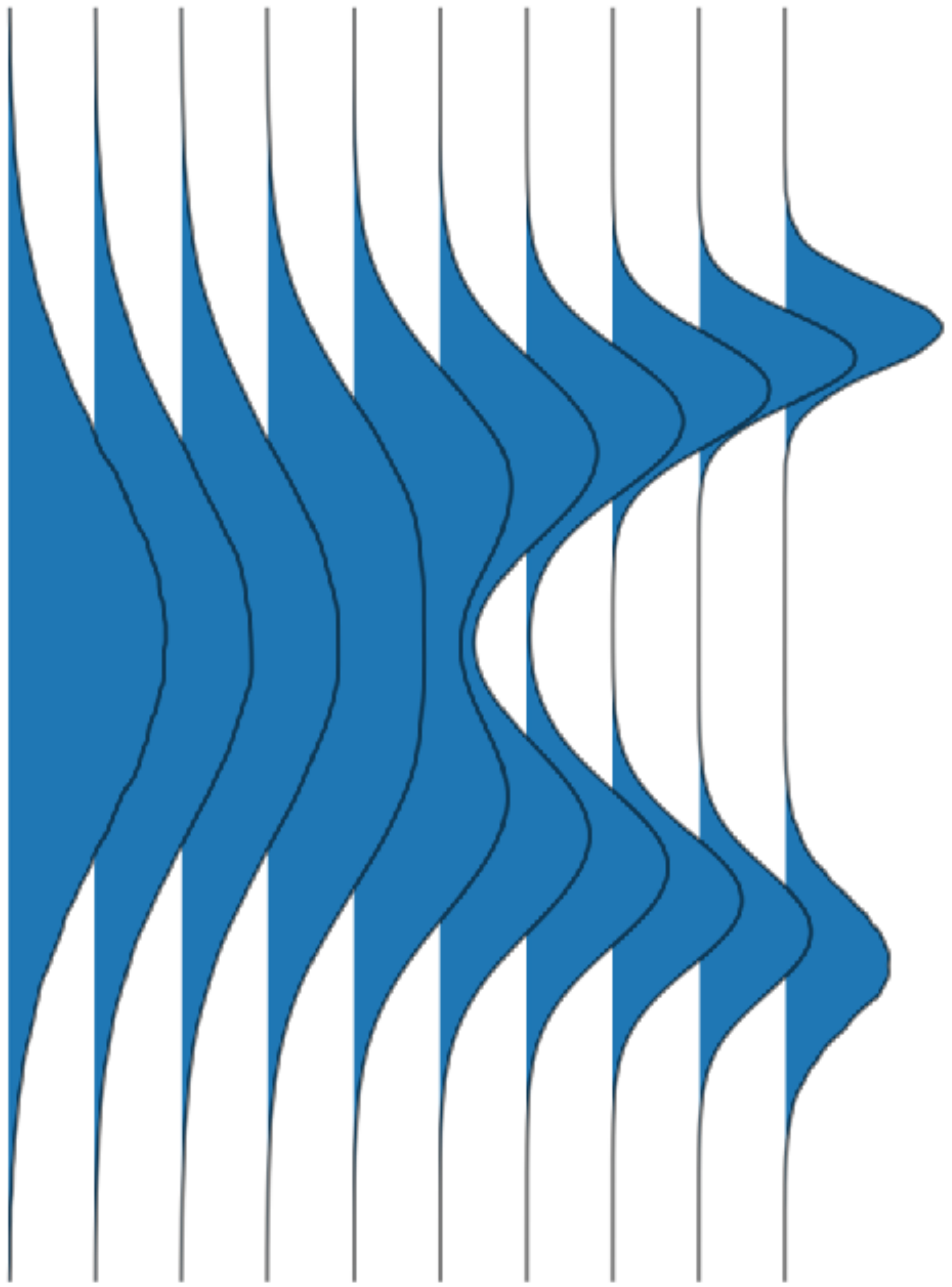




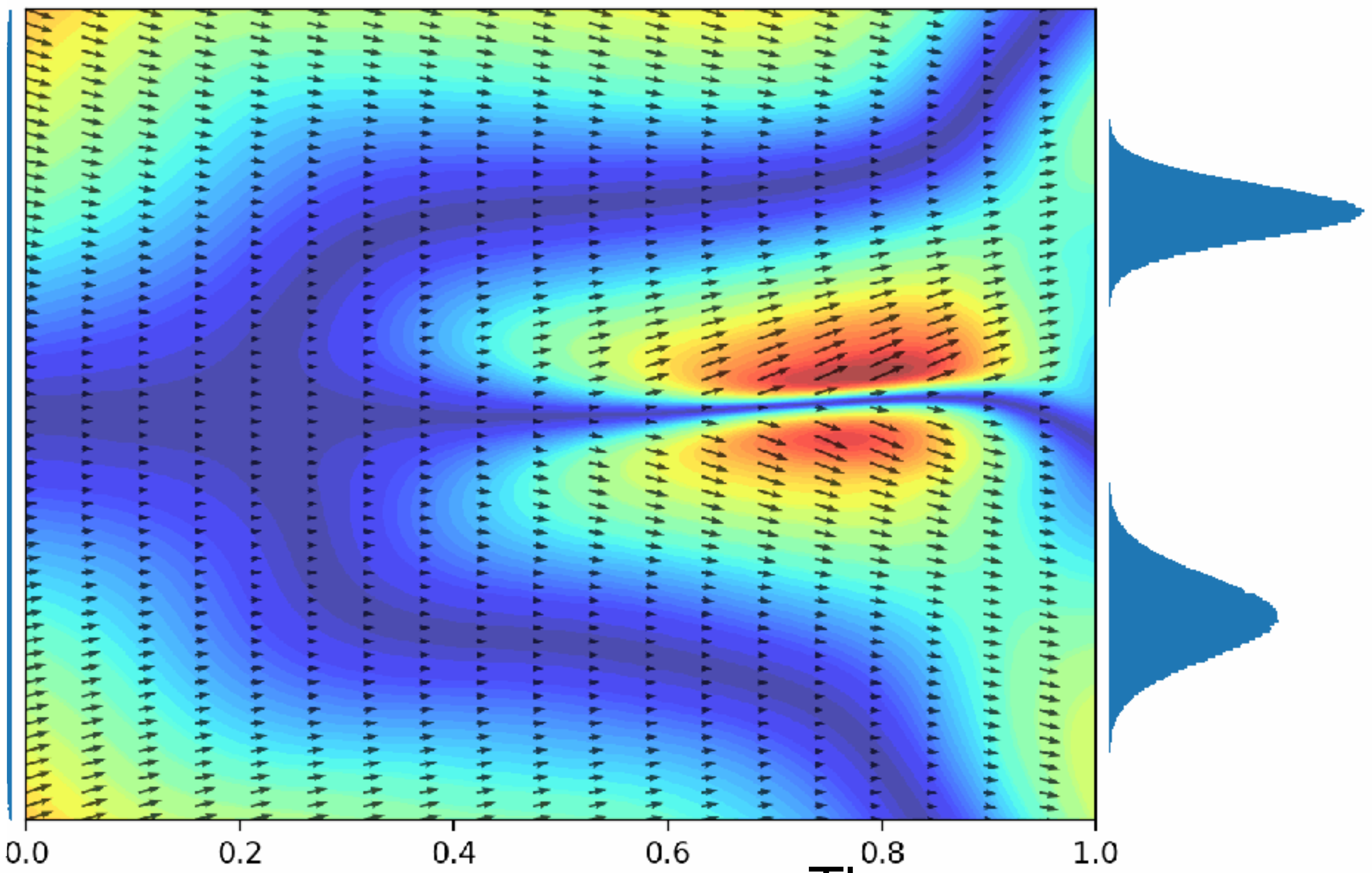
Conditional Velocity $u_t(X_t | X_1 = z)$ → **Marginal Velocity** $u_t(X_t) = \mathbb{E}_{X_1 \sim P_{data}} u_t(X_t | X_1)$

Figures by Dmitrii Kobylianskii

1D Conditional Flow Match



Time



Time

Conditional Velocity $u_t(X_t | X_1 = z)$ → $u_t(X_t) = \mathbb{E}_{X_1 \sim P_{data}} u_t(X_t | X_1)$

Conditional Probability $P_t(X_t | X_1)$ → $P_t(X_t) = \mathbb{E}_{X_0 \sim P_{init}, X_1 \sim P_{data}} P_t(X_t | X_1)$

Figures by Dmitrii Kobylianskii

Train by Sampling Pairs (X0,X1)=(ϵ, z)

Sample a data example

$X_1 = z$ **from Data Set**

Sample a random time

$t \sim Unif[0,1]$

Sample Noise

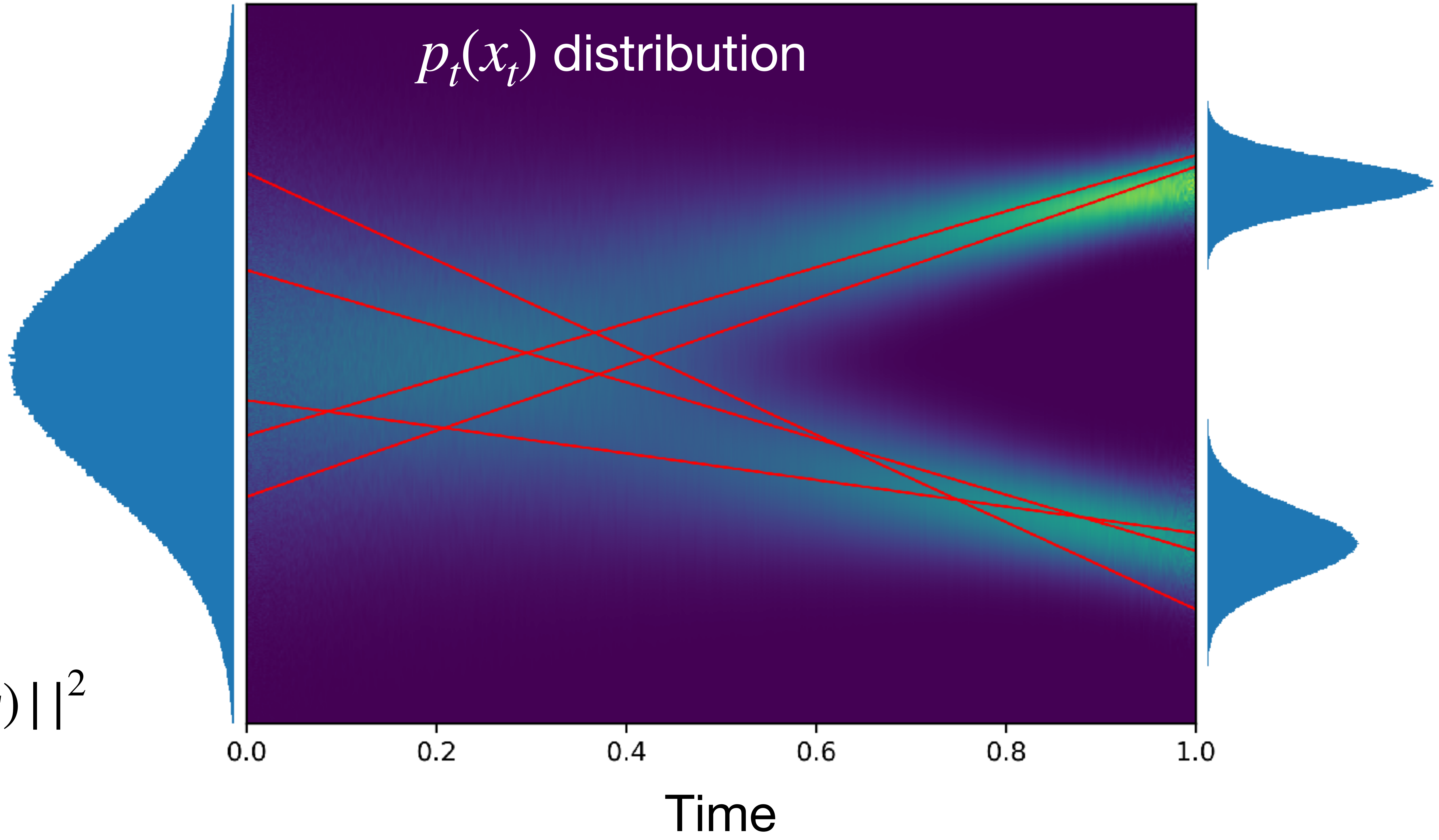
$X_0 = \epsilon \sim \mathcal{N}(0, I_d)$

Set $x_t = (1-t)\epsilon + tz$

Compute Loss

$\mathcal{L}(\theta) = ||u_t^\theta(x_t) - (z - \epsilon)||^2$

Update θ **via gradient descent on** $\mathcal{L}(\theta)$



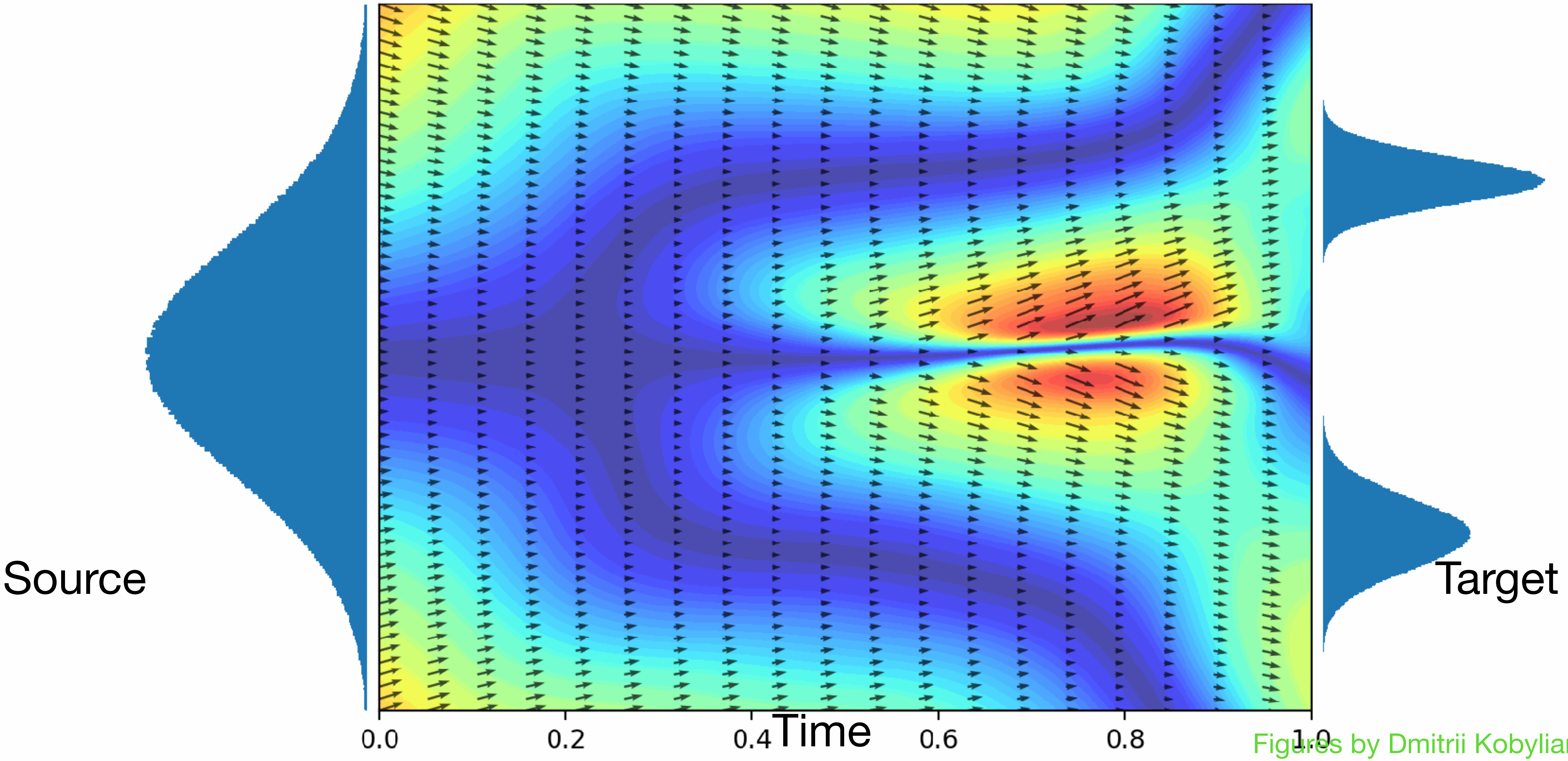
Figures by Dmitrii Kobylianskii

1D toy Example

Trained $u_t^\theta(X_t)$ distribution

Lines are flow — solutions of ODE

The trained $u_t^\theta(X_t)$ defines the velociy from which we calculate the flow $\psi_t(x)$ which is the solution of the ODE $dX_t = u_t^\theta(X_t)dt$



Figures by Dmitrii Kobylianskii

$u_t^\theta(X_t)$ **is learnt**

Sampling Algorithm

Lines are flow — solutions of ODE

Set $t = 0$

Set step size $h = \dfrac{1}{N}$

Draw a random sample $X_0 \sim p_{init}$

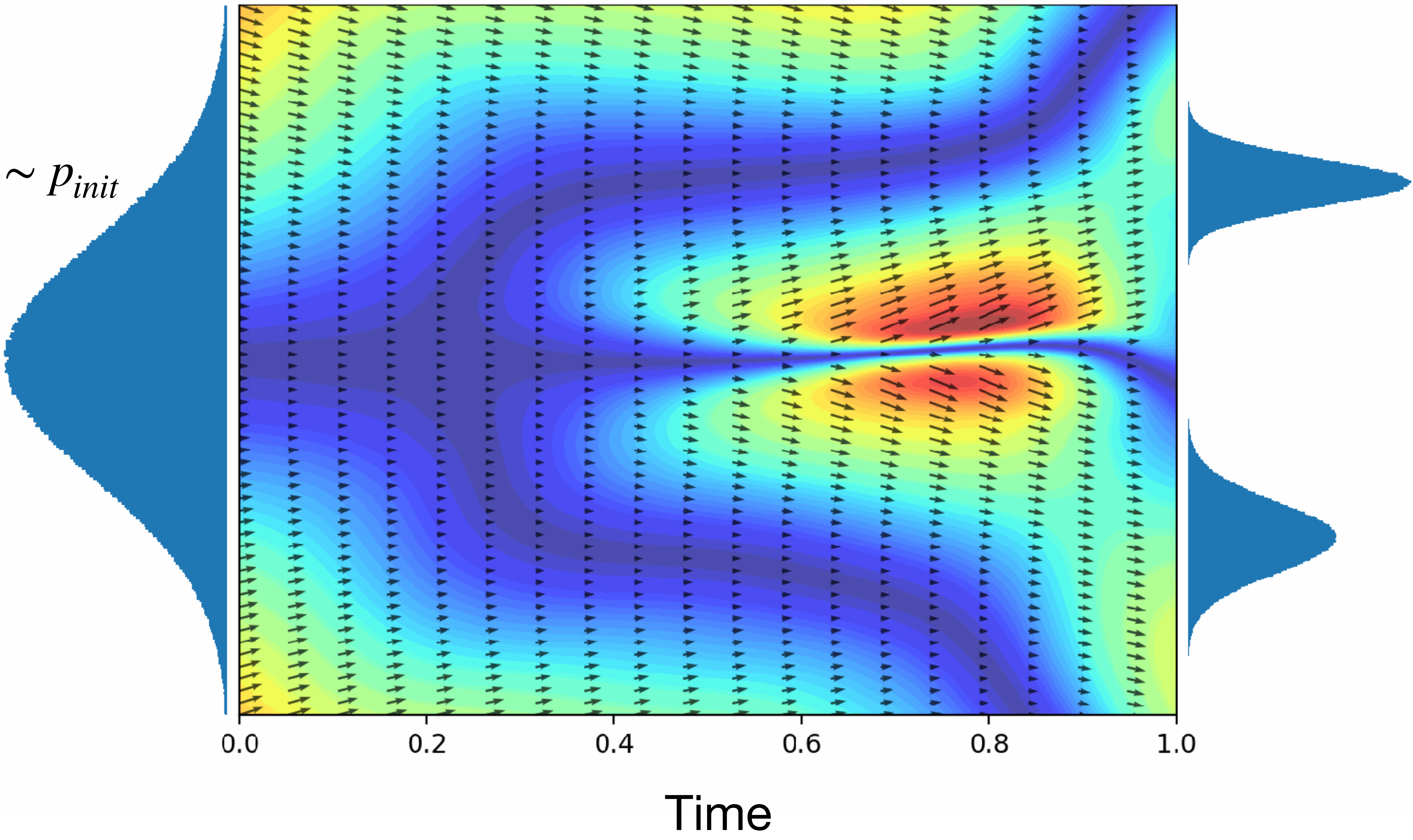
For $i = 1,\ldots,N-1$ **do**

$X_{t+1} = X_t + h u_t^\theta(X_t)$

$t \to t + h$

end for

Return X_1



Figures by Dmitrii Kobylianskii

Leveraging Stable Diffusion Techniques?

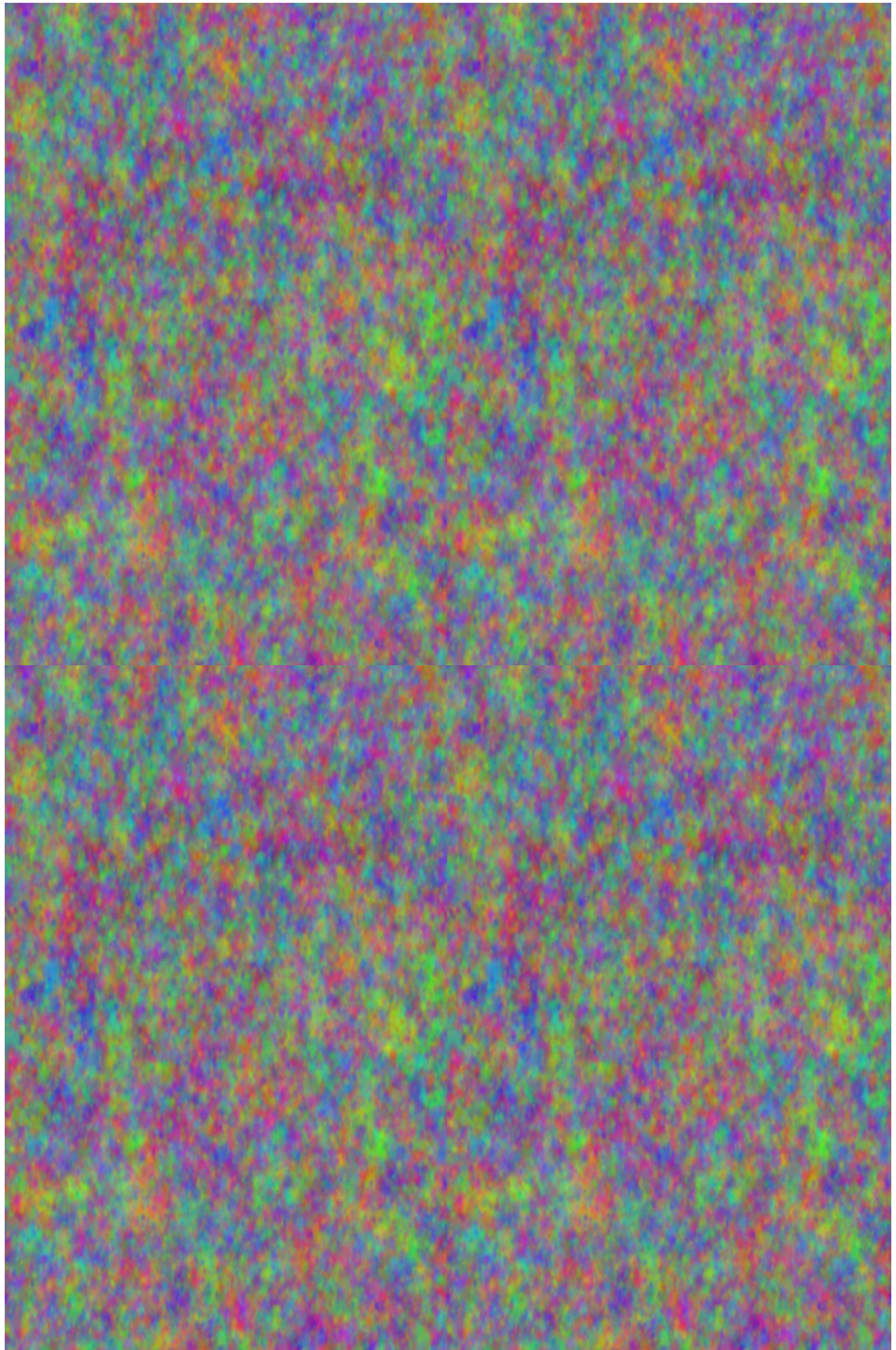
Feature/Principle	SD3.x	Parnassus
Output	Images (pixels)	Particle Flow Candidates - PFC
Conditioning	Text Prompts	Generator-level-particles (truth particles)
Generation	text—>image	2 stages: Event, PFC set (truth)—>set (PFC)
Generative Principle	Flow Matching (Rectified Flow)	Flow Matching (Rectified Flow)
Probability Path	Rectified Flow, straight trajectory	Rectified Flow, straight trajectory
Sampling Efficiency	ODE based	ODE based
Architecture	Transformer based	Transformer based
Generalization	Unseen Text/Image Prompts	Unseen Phyiscs Processes

Leveraging Stable Diffusion Techniques?

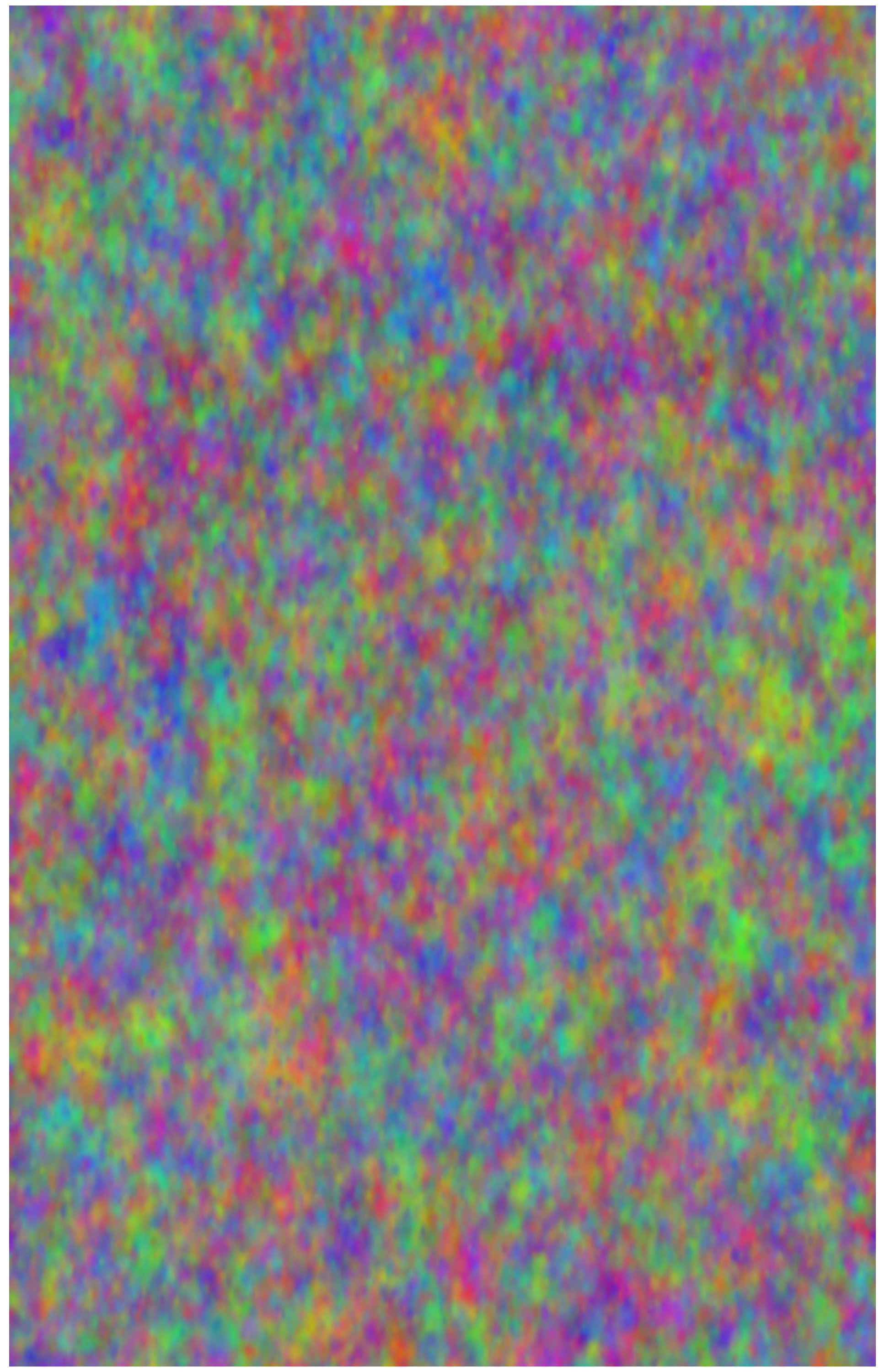
Feature/Principle	SD3.x	Parnassus
Output	Images (pixels)	Particle Flow Candidates - PFC
Conditioning	Text Prompts	Generator-level-particles (truth particles)
Generation	text—>image	2 stages: Event, PFC set (truth)—>set (PFC)
Generative Principle	Flow Matching (Rectified Flow)	Flow Matching (Rectified Flow)
Probability Path	Rectified Flow, straight trajectory	Rectified Flow, straight trajectory
Sampling Efficiency	ODE based	ODE based
Generalization	Unseen Text/Image Prompts	Unseen Phyiscs Processes

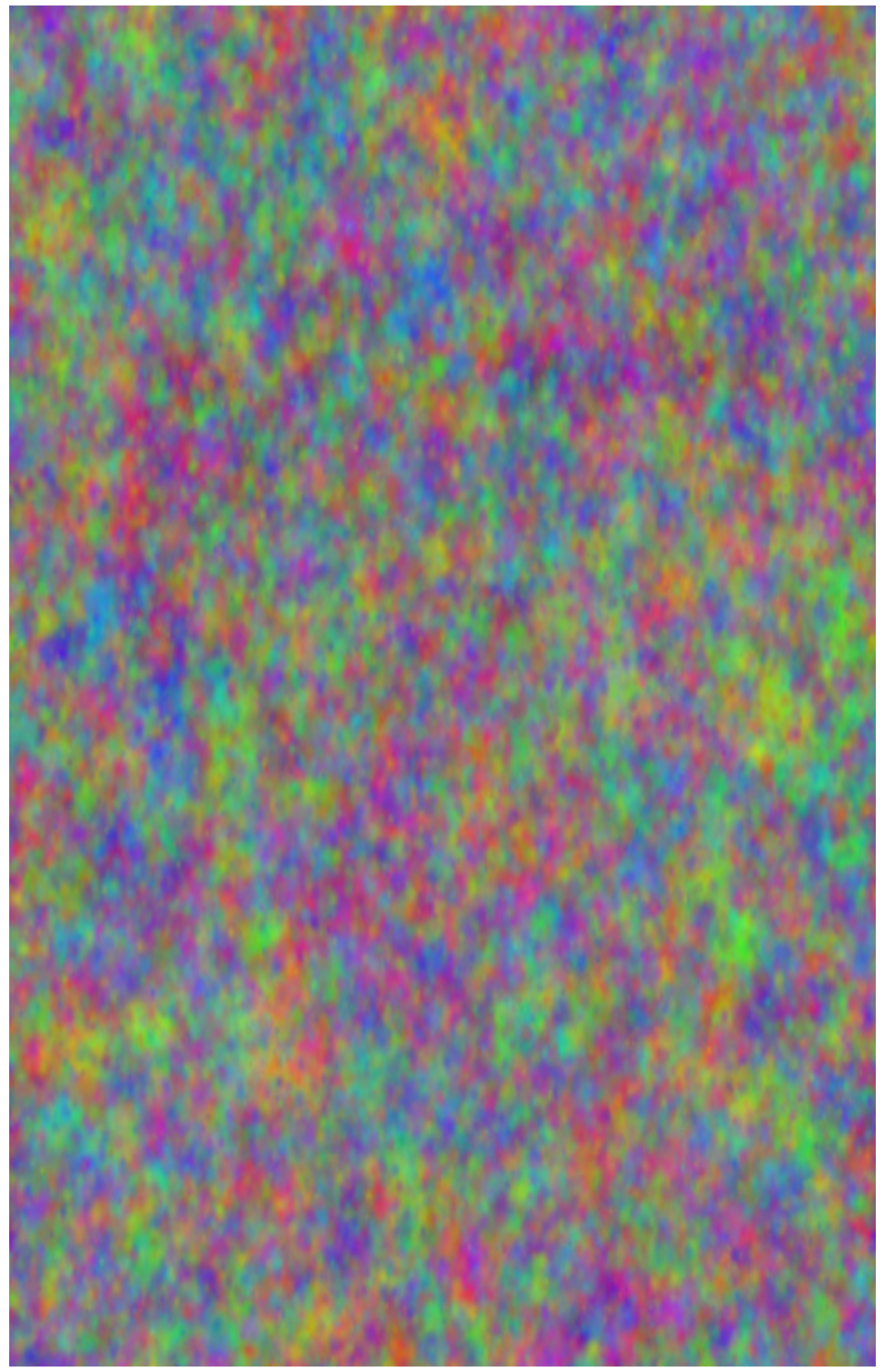
Transformer based

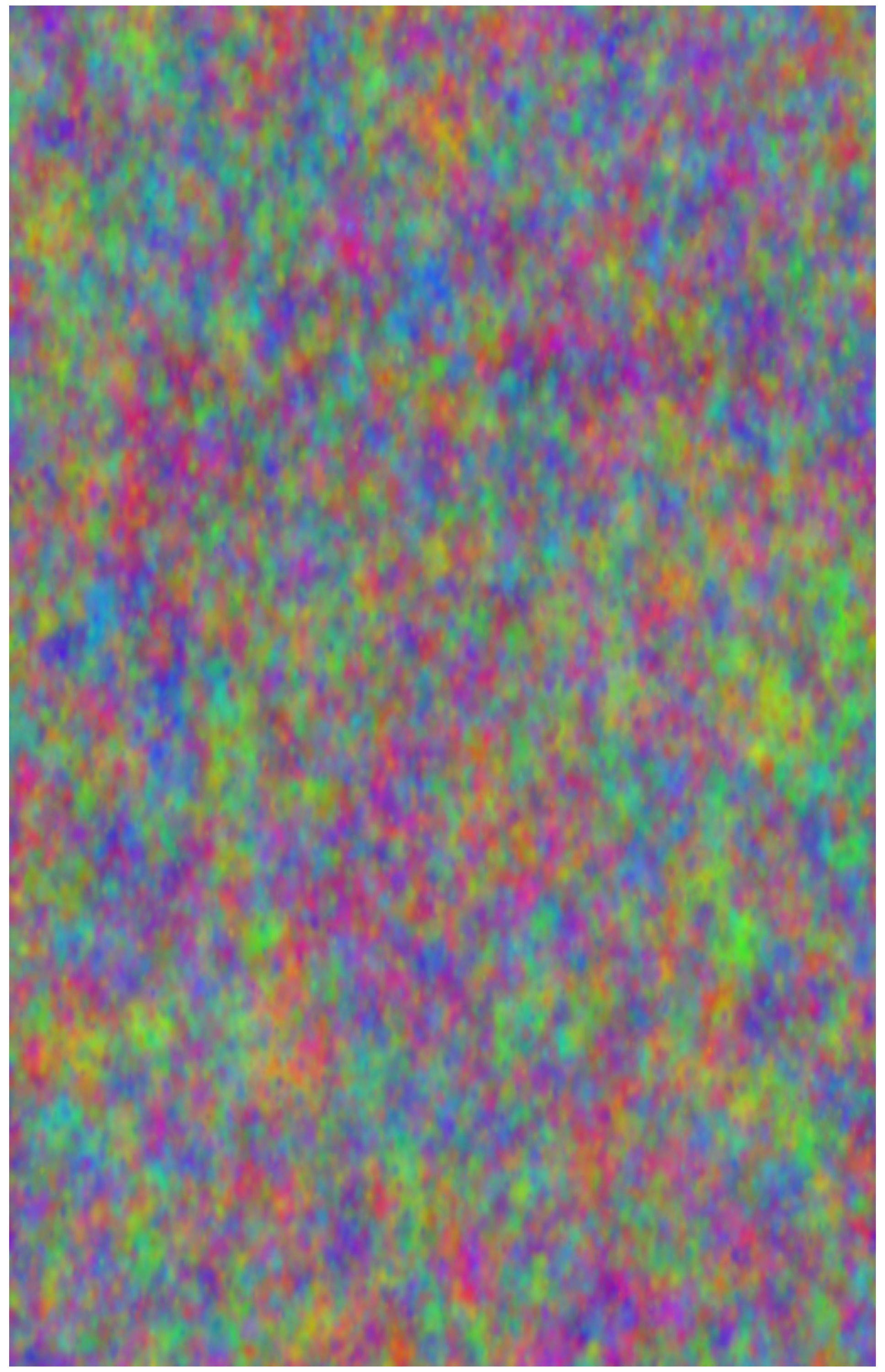
Draw a
walking
smiling cat

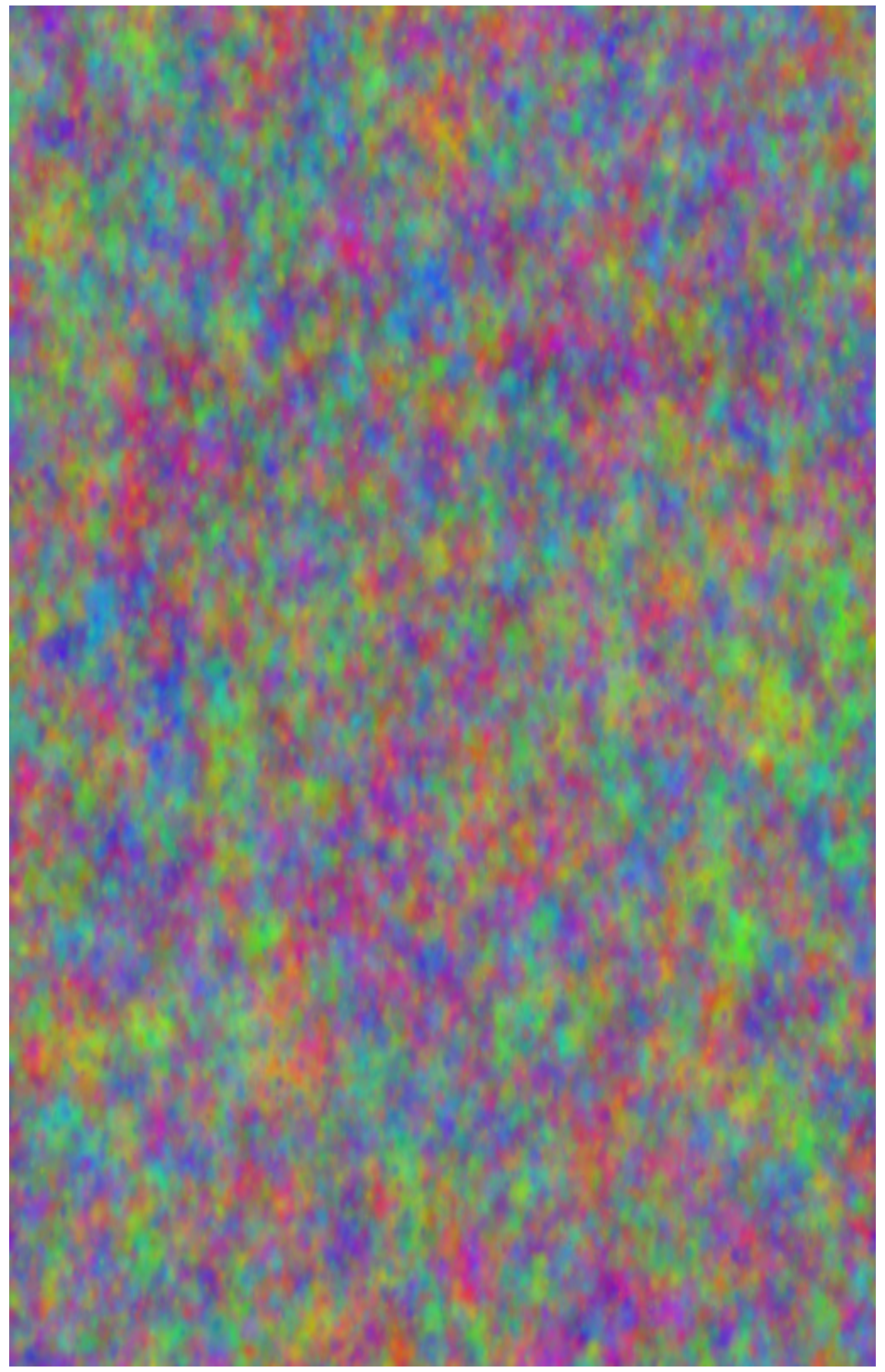


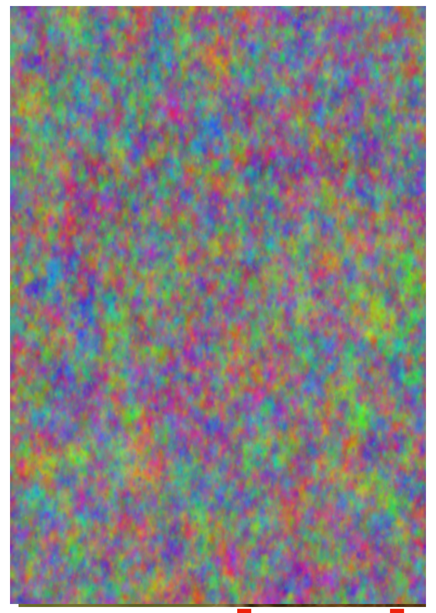
Draw a walking smiling cat

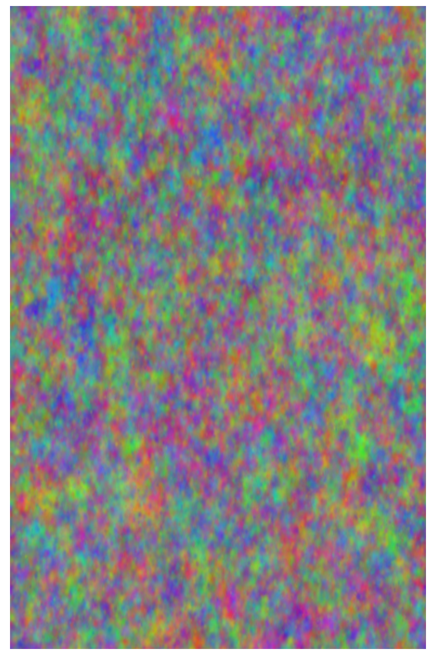


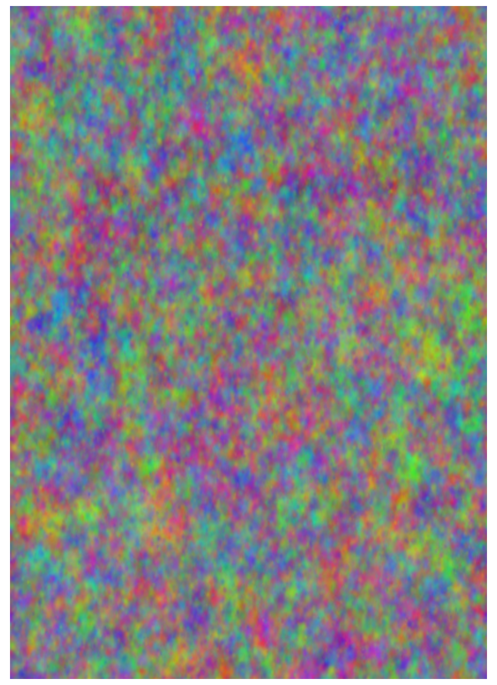


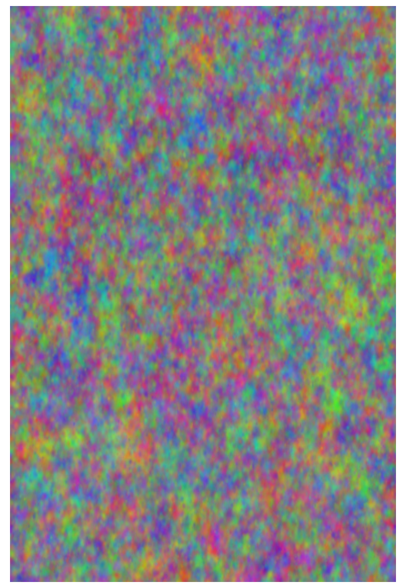












The embedded Image issues a Querry: **For this patch of the image,** which **words are relevant**

The embedded text projects a Key: How relevant **each word is**

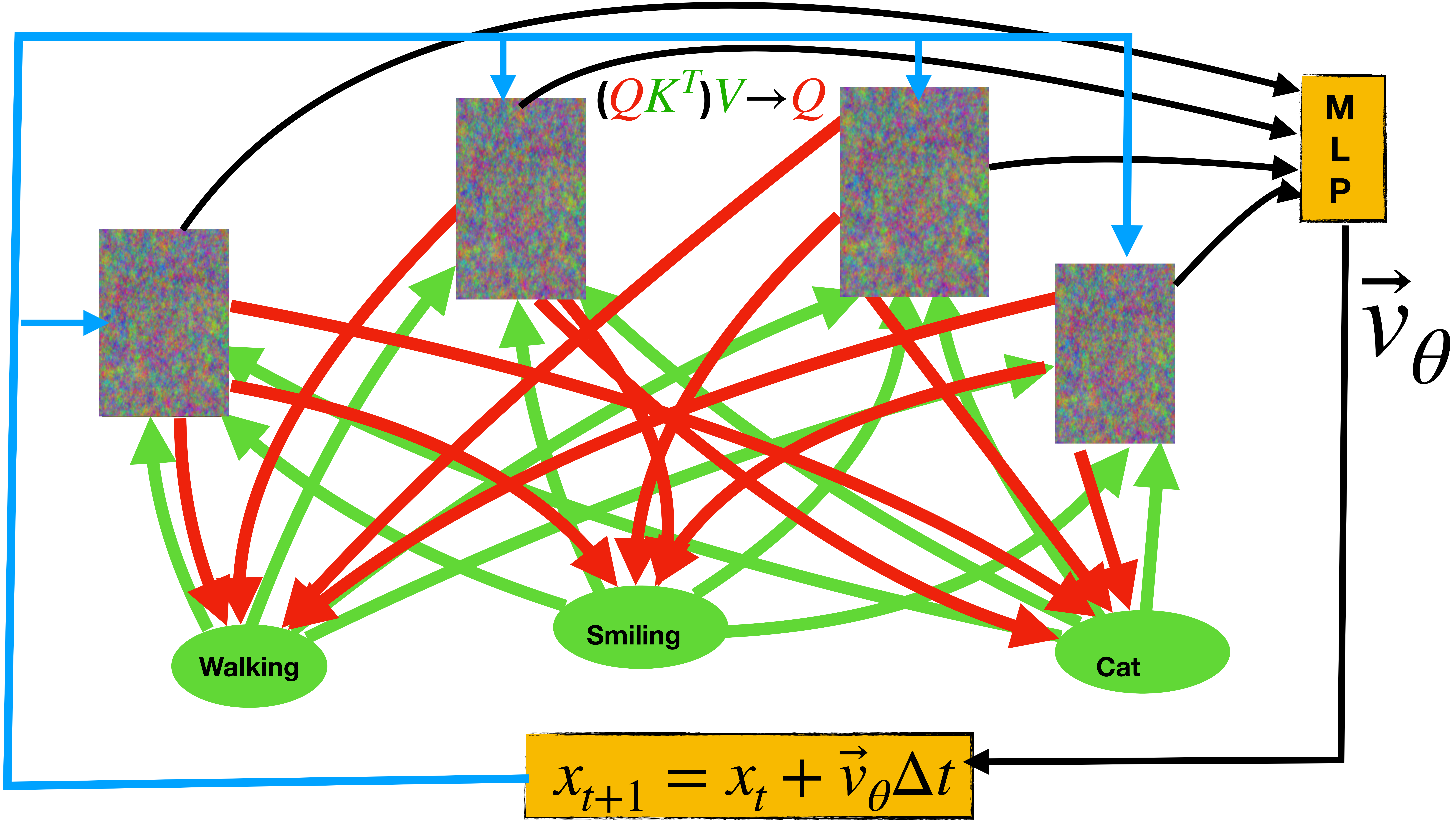
and a Value: **How much** content **should be injected into the image?**

$$(QK^T)V \rightarrow Q$$

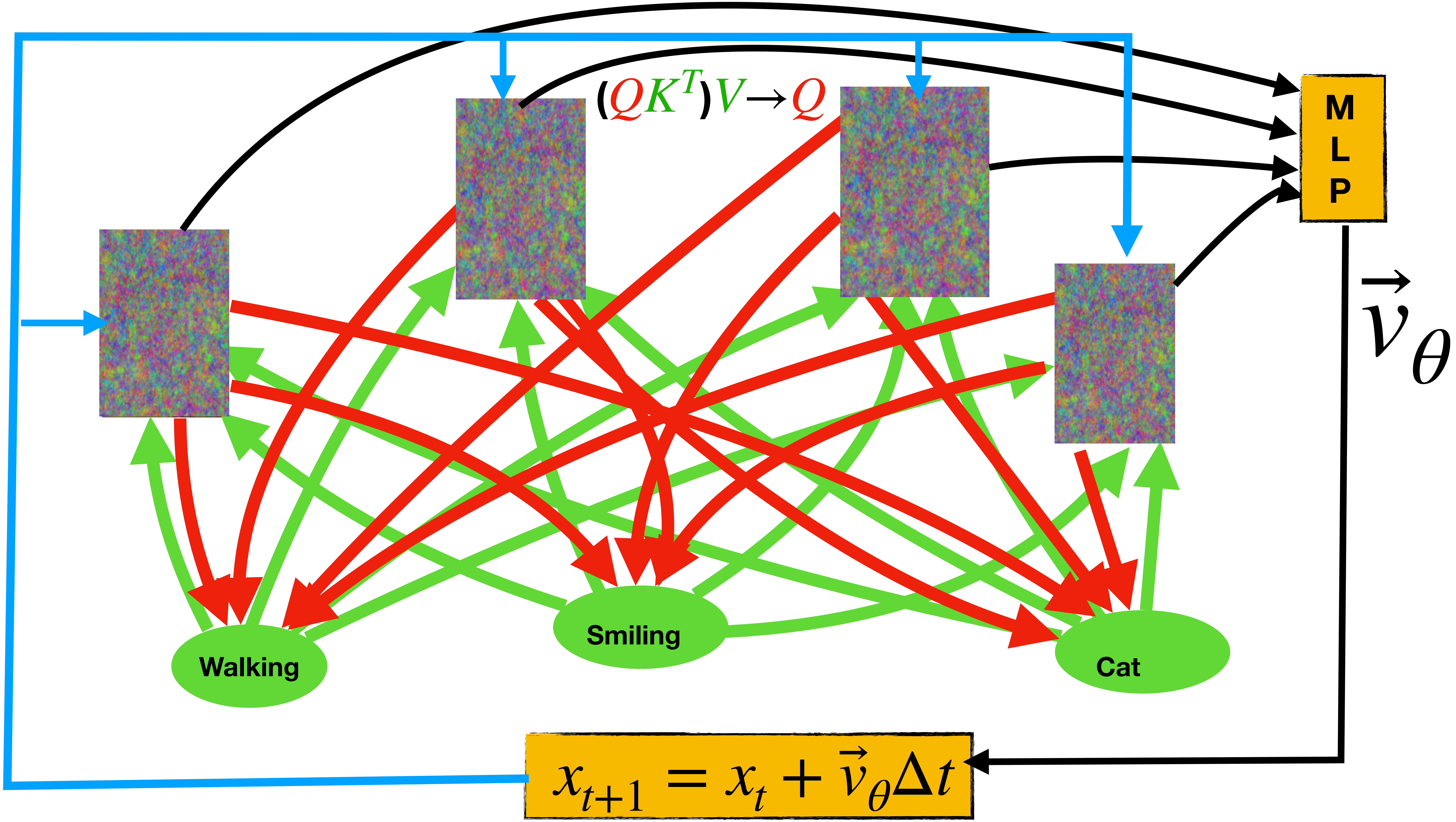
Walking

Smiling

Cat



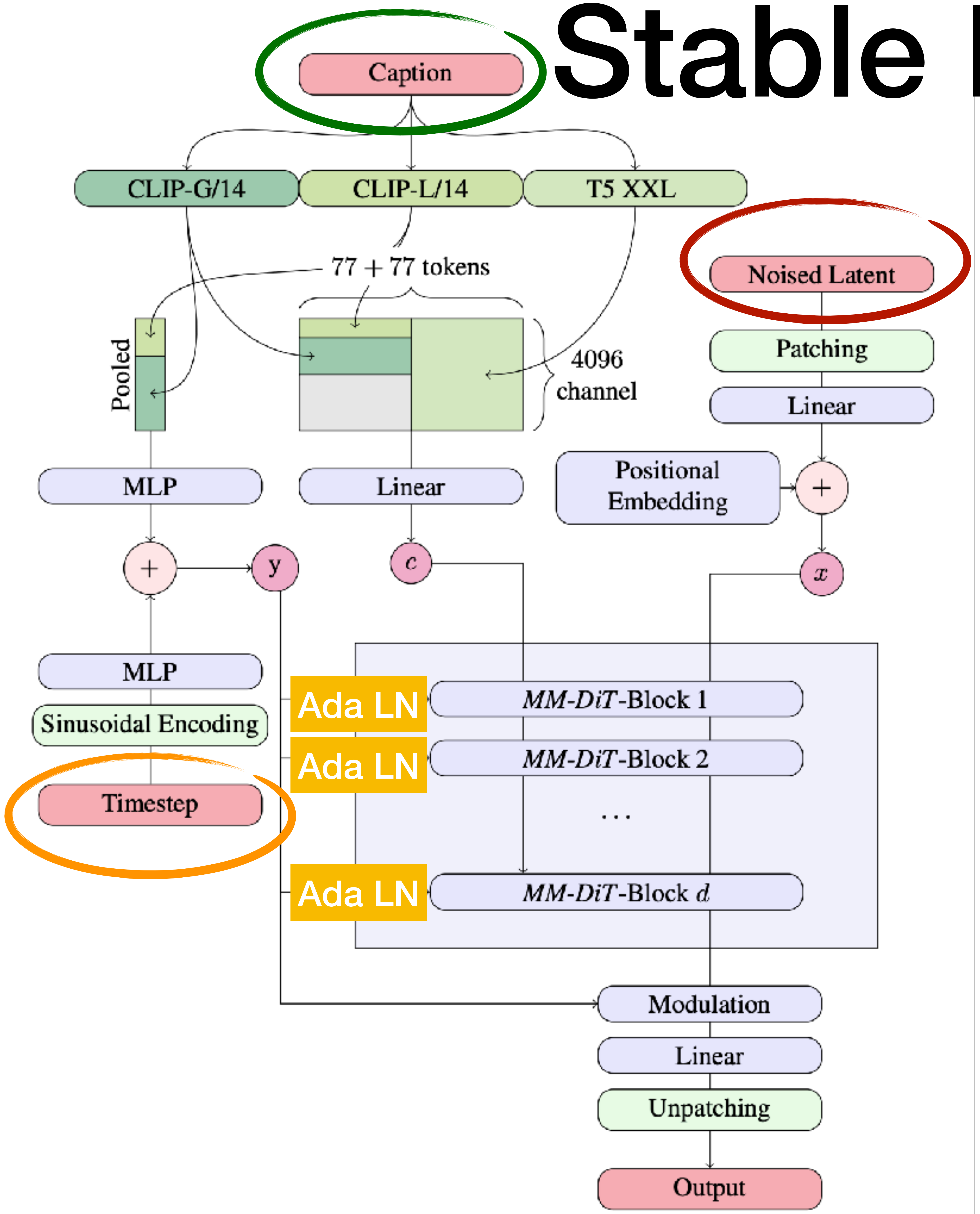
$(QK^T)V \to Q$
MLP
$\vec{v}_\theta$
Walking
Smiling
Cat
$x_{t+1} = x_t + \vec{v}_\theta \Delta t$



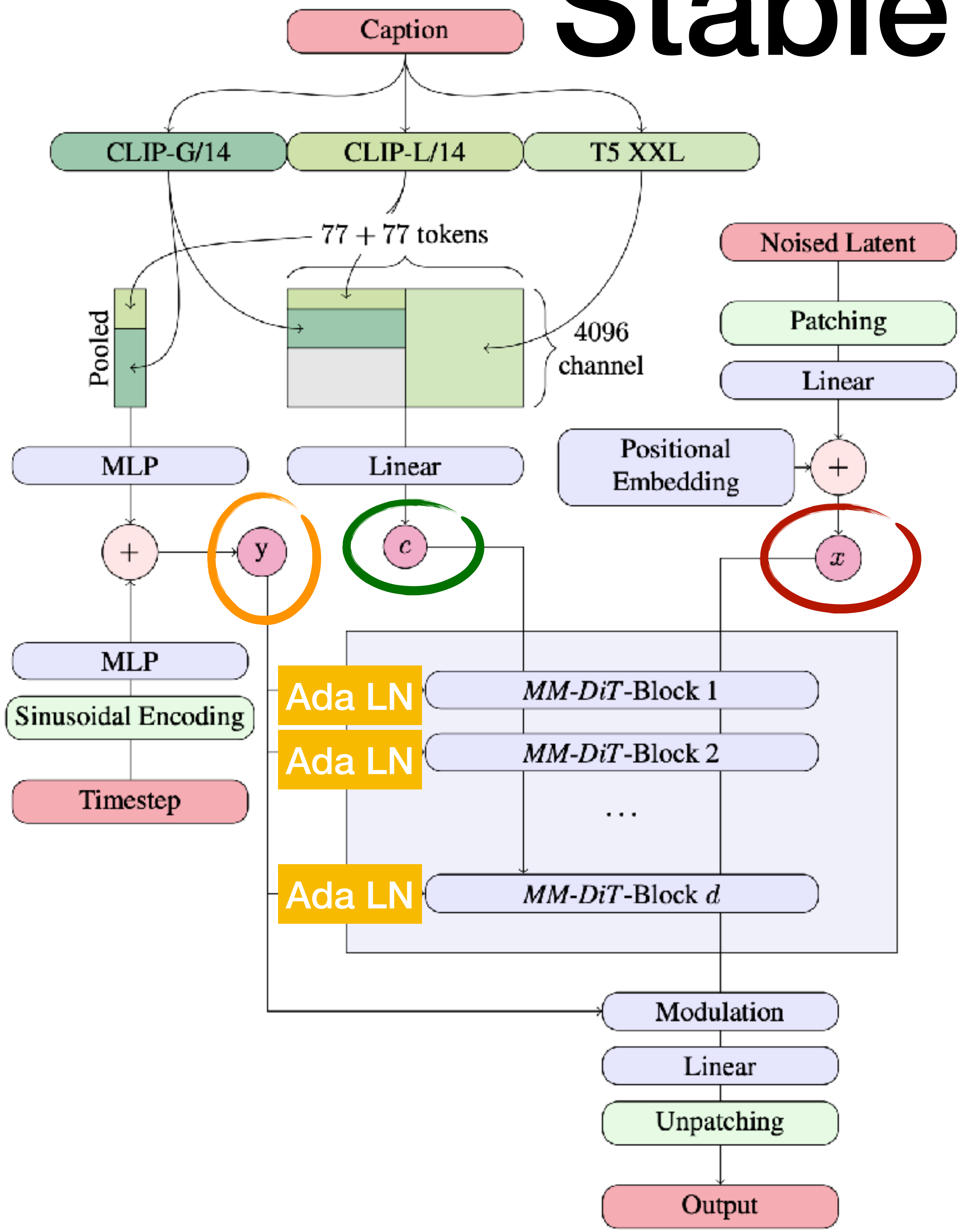
$(QK^T)V \to Q$
M
L
P
$\vec{v}_\theta$
Walking
Smiling
Cat
$x_{t+1} = x_t + \vec{v}_\theta \Delta t$

Draw a walking smiling cat

Stable Diffusion 3



Stable Diffusion 3



Leveraging Stable Diffusion Techniques?

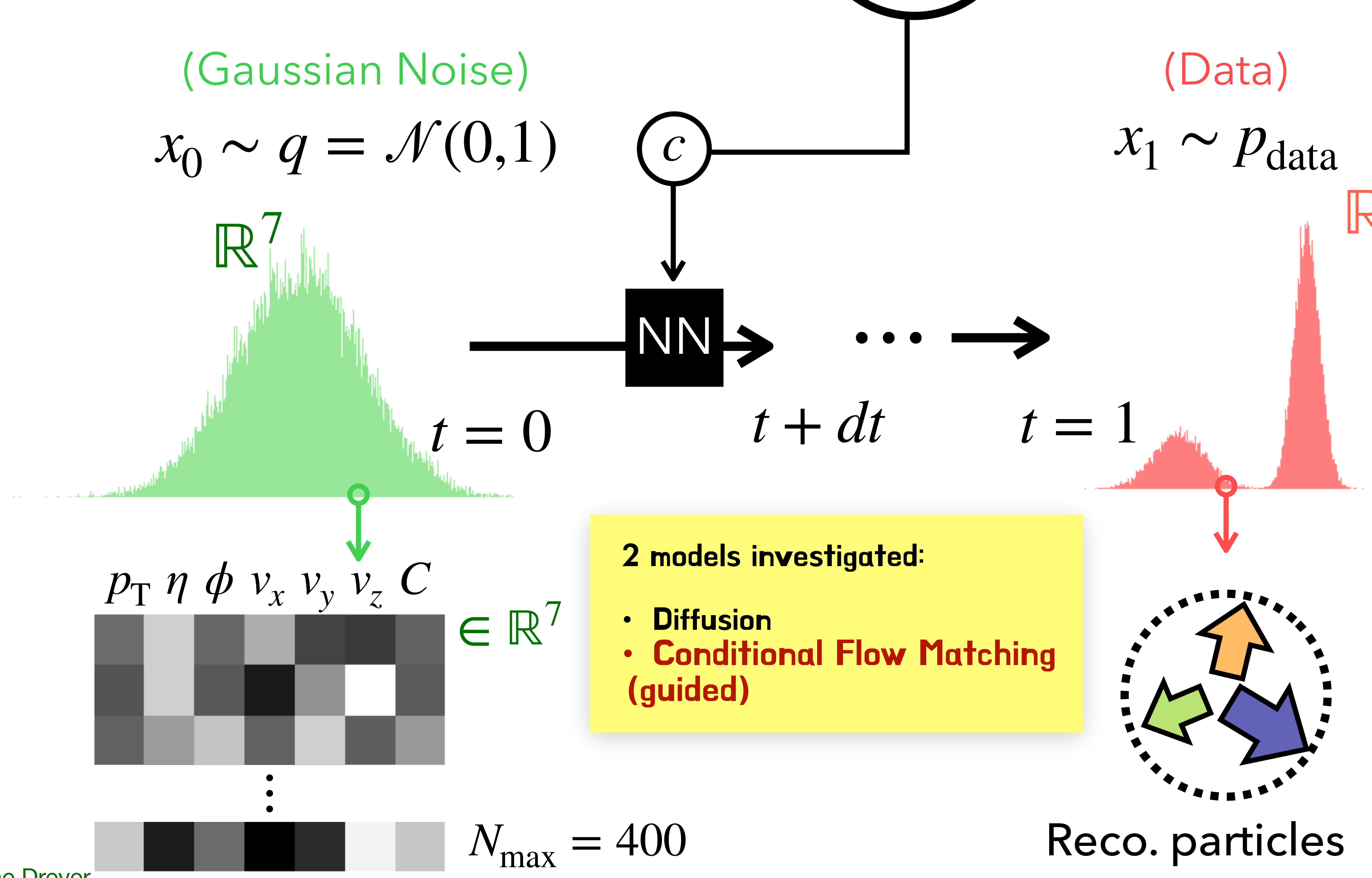
Feature/Principle	SD3.x	Parnassus
Output	Images (pixels)	Particle Flow Candidates - PFC
Conditioning	Text Prompts	Generator-level-particles (truth particles)
Generation	text—>image	truth—>event Level truth+event—>PFC
Generative Principle	Flow Matching (Rectified Flow)	Flow Matching (Rectified Flow)
Probability Path	Rectified Flow, straight trajectory	Rectified Flow, straight trajectory
Sampling Efficiency	ODE based	ODE based
Architecture	Transformer based	Transformer based
Generalization	Unseen Text/Image Prompts	Unseen Phyiscs Processes

WEIZMANN INSTITUTE OF SCIENCE

BERKELEY LAB

2024	PRD 110, 092013	*Advancing set-conditional set generation: Diffusion models for fast simulation of reconstructed particles*
2024	PRL 133, 211902	*Automated Approach to Accurate, Precise, and Fast Detector Simulation and Reconstruction*
2025	arXiv:2503.19981	*Conditional Deep Generative Models for Simultaneous Simulation and Reconstruction of Entire Events*

Conditional generation



2 models investigated:

- **Diffusion**
- **Conditional Flow Matching (guided)**

Slide from Etienne Dreyer

DATA SET

- CMS Open 2011 Data

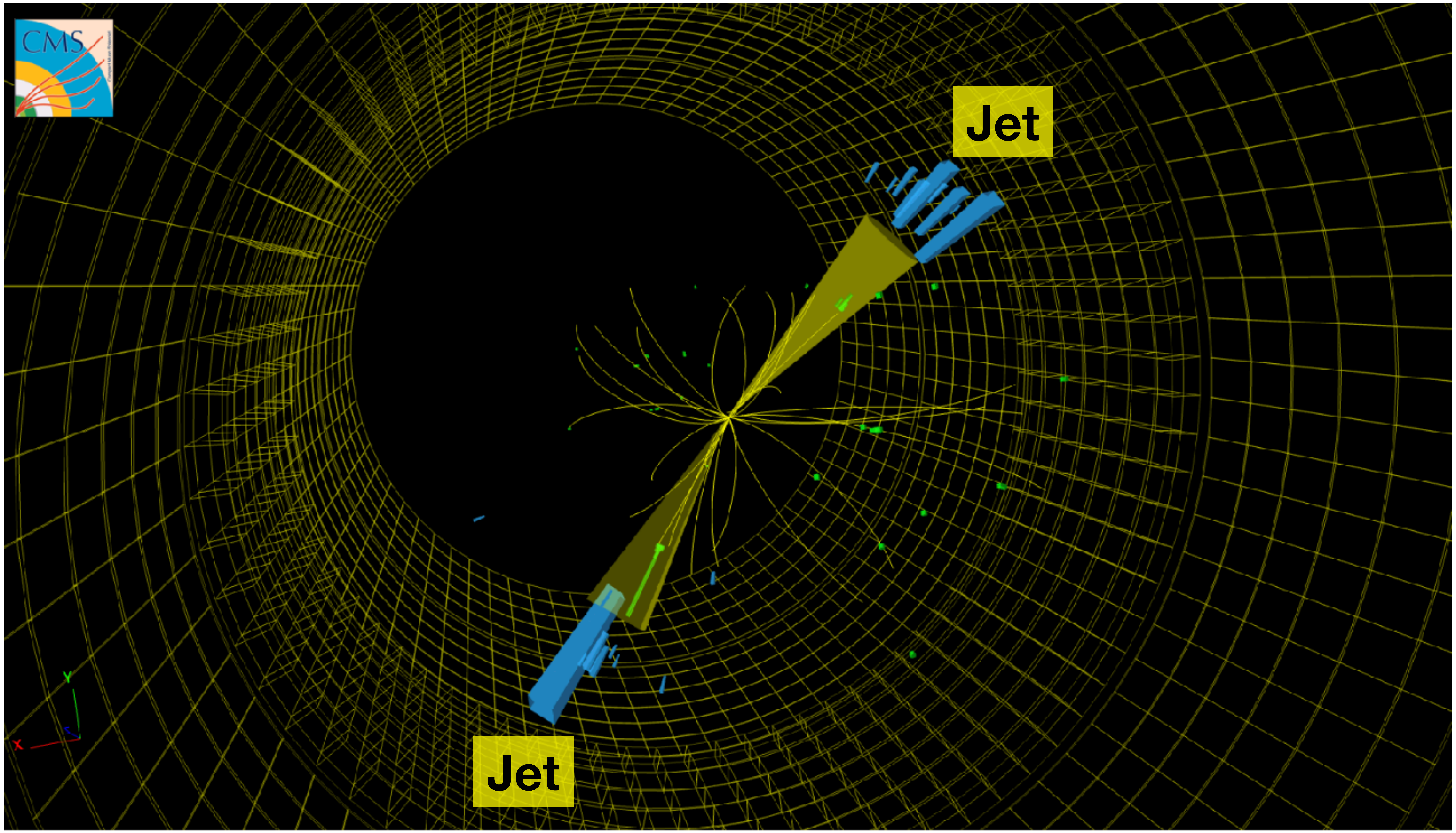
- Trained on QCD (mainly di-jet ($q\bar{q}$) events
  and top-anti top ($t\bar{t}$) production events

- Tested on QCD,$t\bar{t}$ and
  High momentum QCD,H->4 leptons (out of distribution)

2011 CMS Open Data

Train: 2.8M QCD and $t\bar{t}$ events

Test: 200k QCD, $t\bar{t}$, and $H \to 4\ell$



Input: set of stable generator particles in event

Target: set of CMS particle flow candidates

Parnassus Methodology

- Two networks:

missing Transverse Energy	Sum of Scalar Transverse Energy	Number of Particle Flow Candidates

  - Event Level Network $\varepsilon^{pf} = (E_x^{miss}, E_y^{miss}, H^T, N_{part})$
  - Particle Flow Candidates Network

$$\mathscr{P}^{pf} = (p_T^{rel}, \eta, \phi, \overrightarrow{vertex}, class) \in \mathbb{R}^7$$

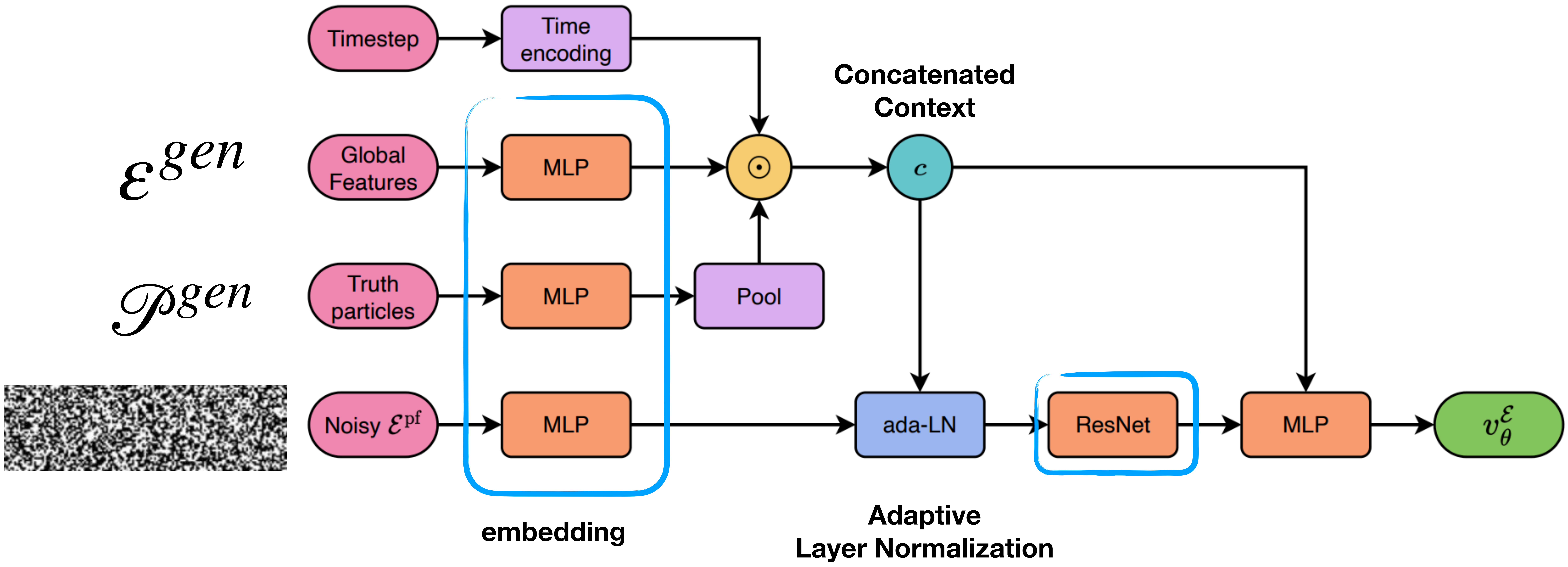
$$p_T^{rel} = \frac{p_T}{H_T}$$

- Networks conditions on the set of Gen particles $\varepsilon^{gen}, \mathscr{P}^{gen}$

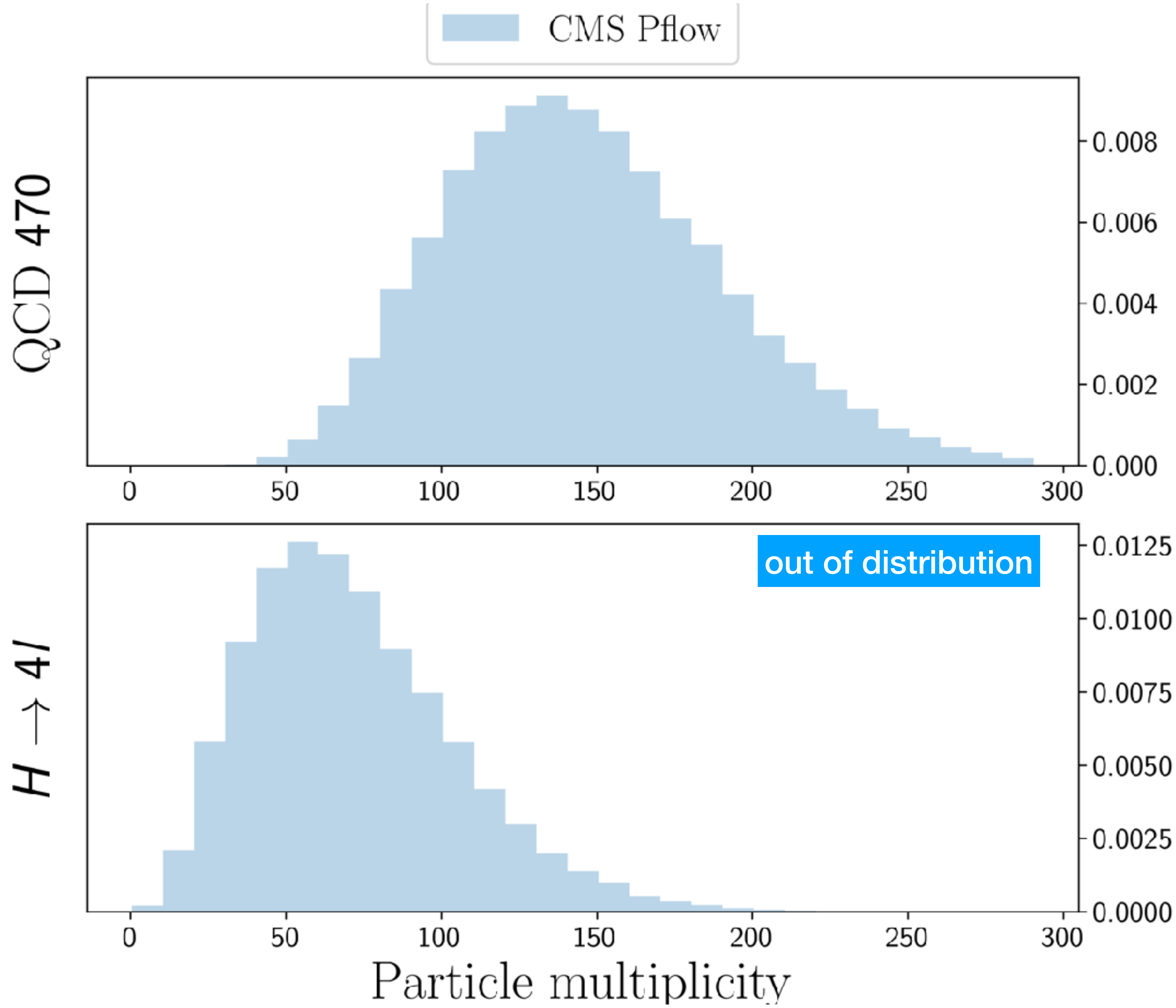
- Classes: Charged Hadron, Electron, Muon, Neutral hadron, Gamma

Event Level Network

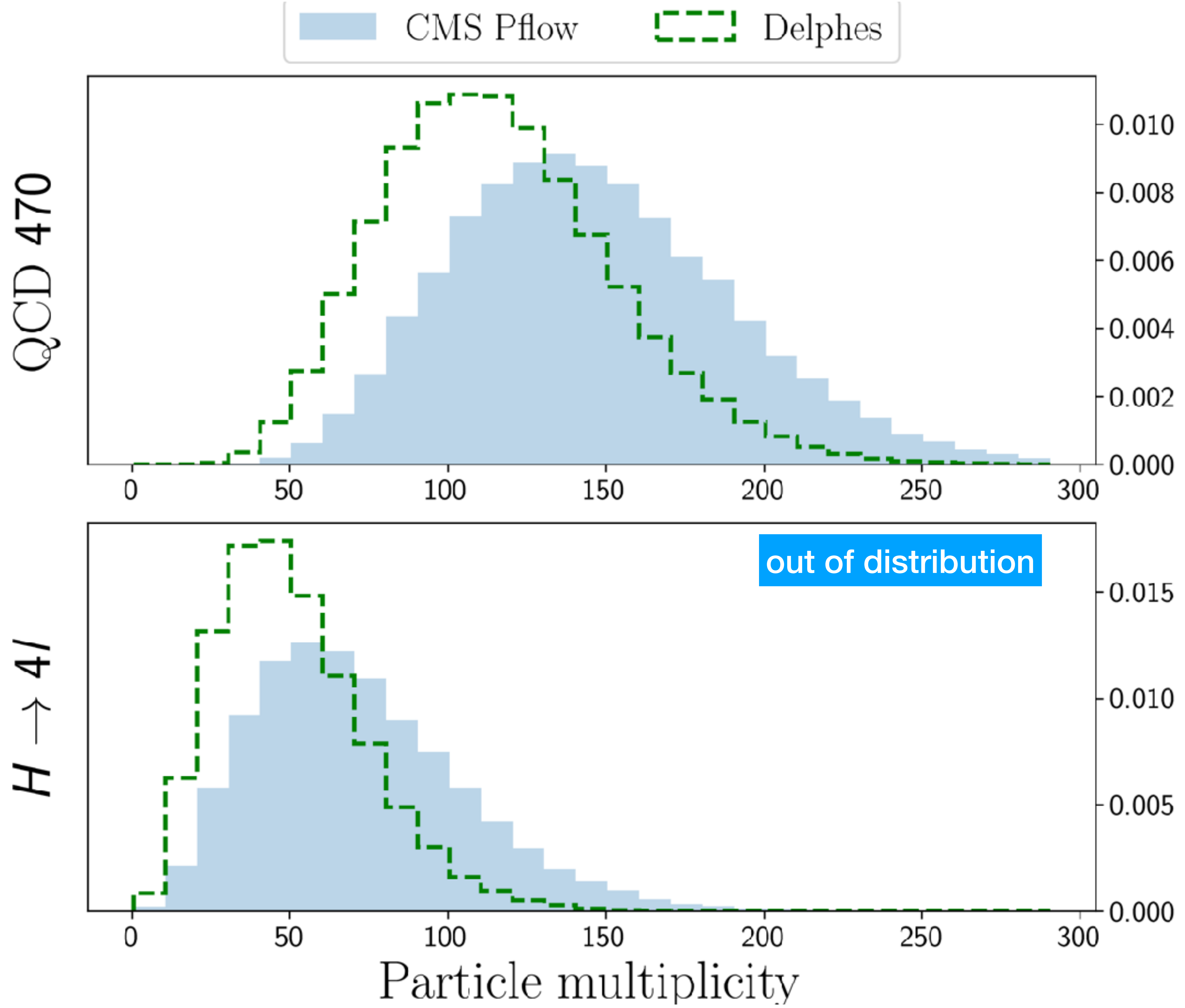
- Based on ResNet (NN, No Transformer)



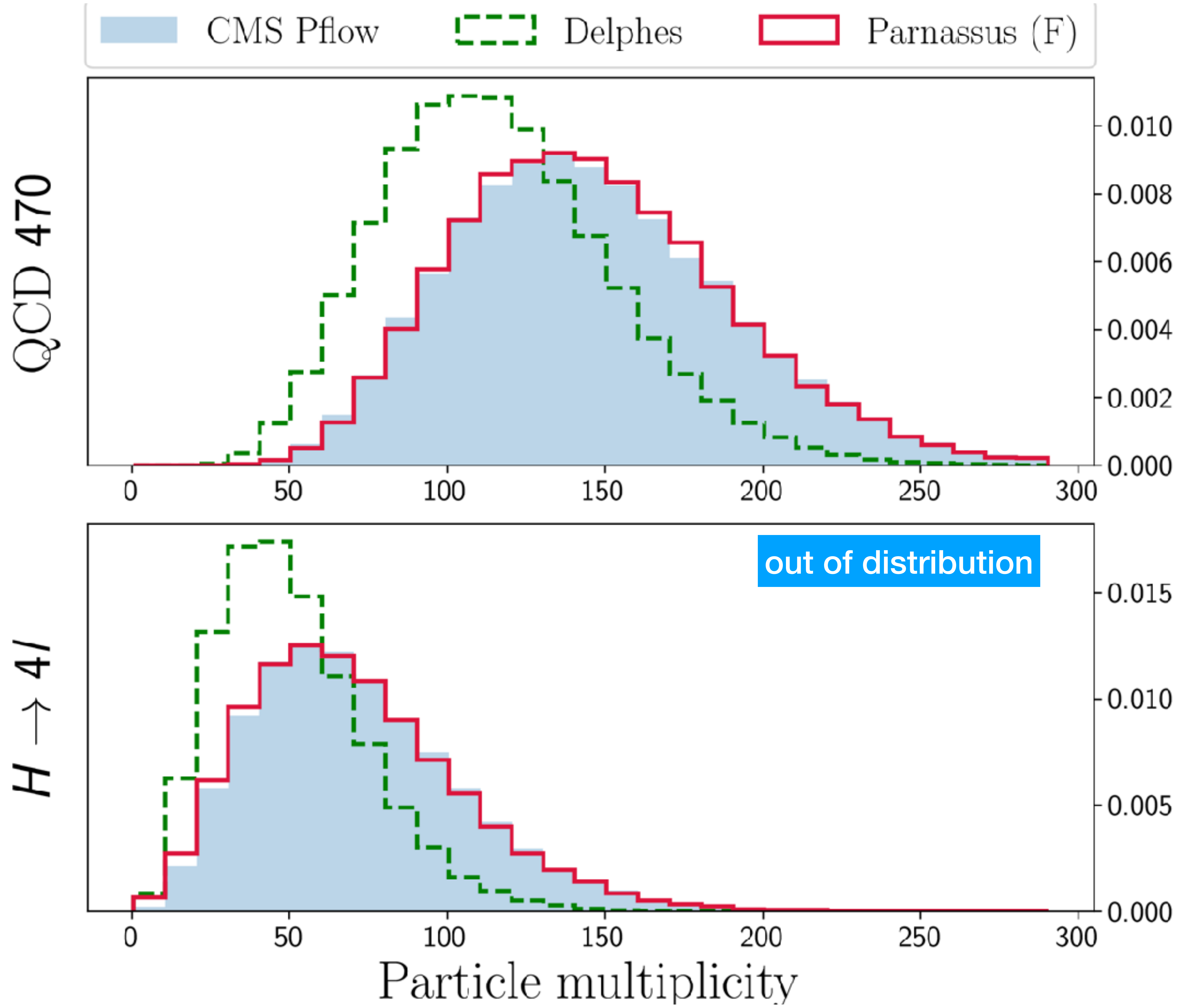
Event Level Performance



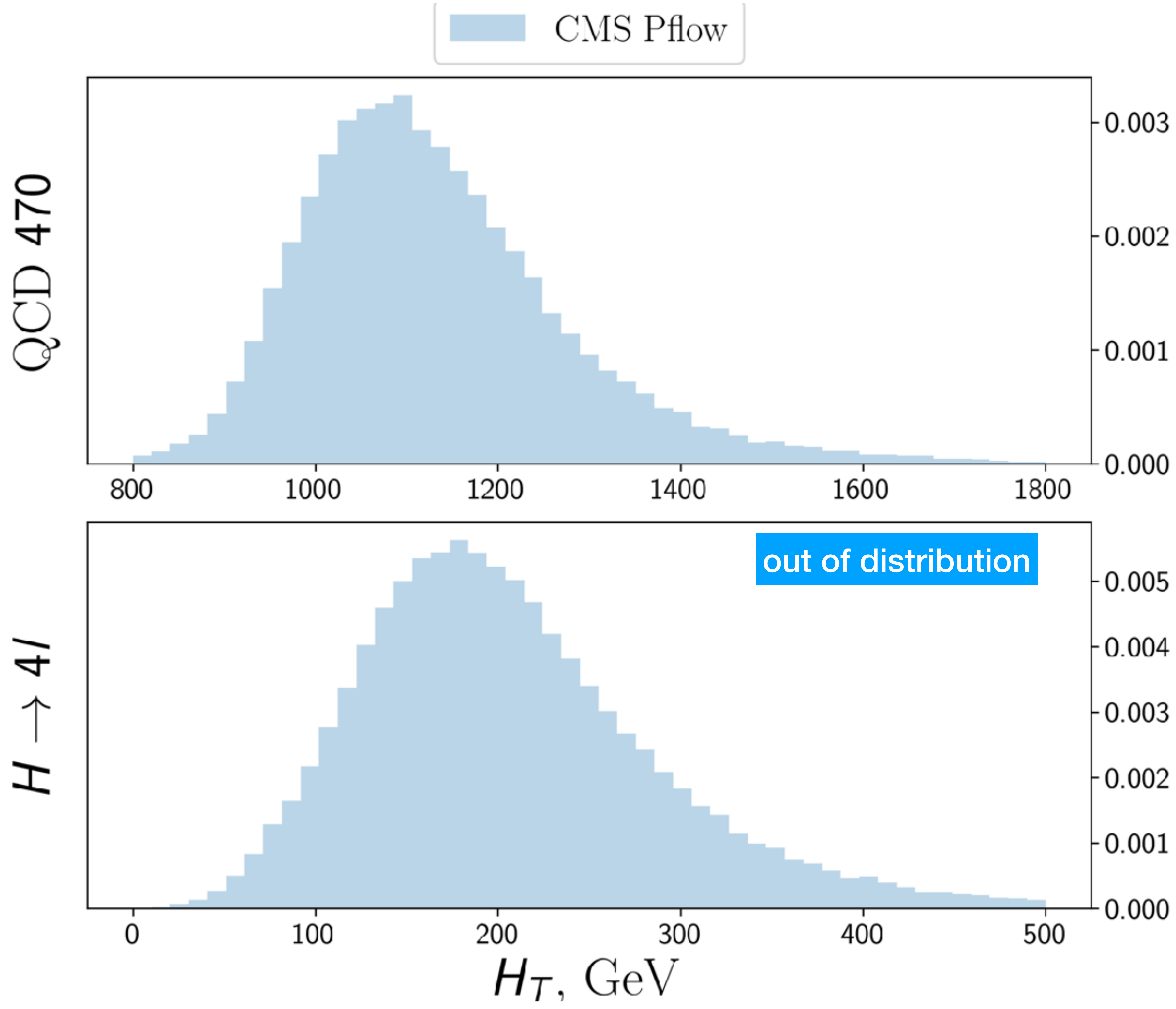
Event Level Performance



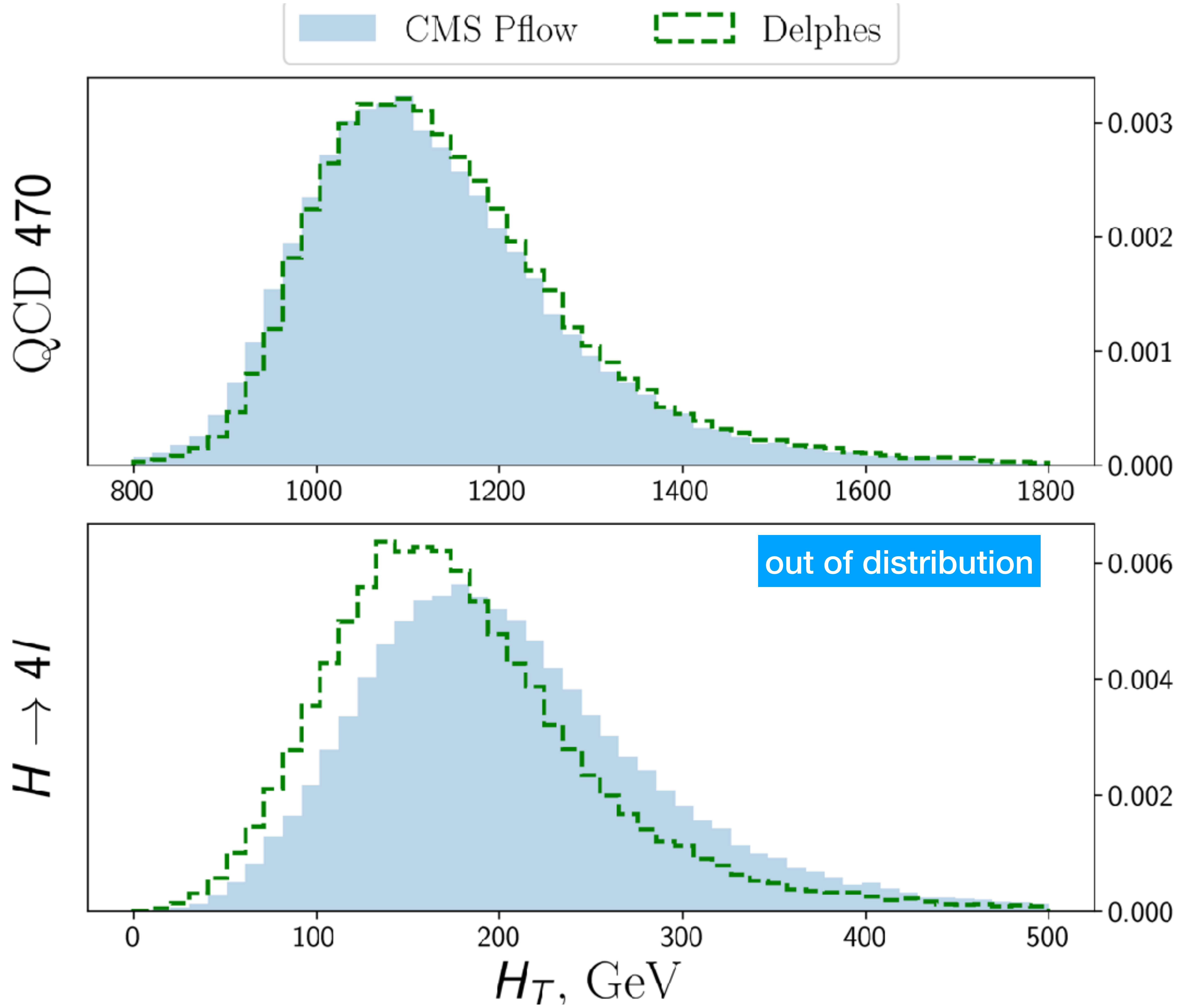
Event Level Performance



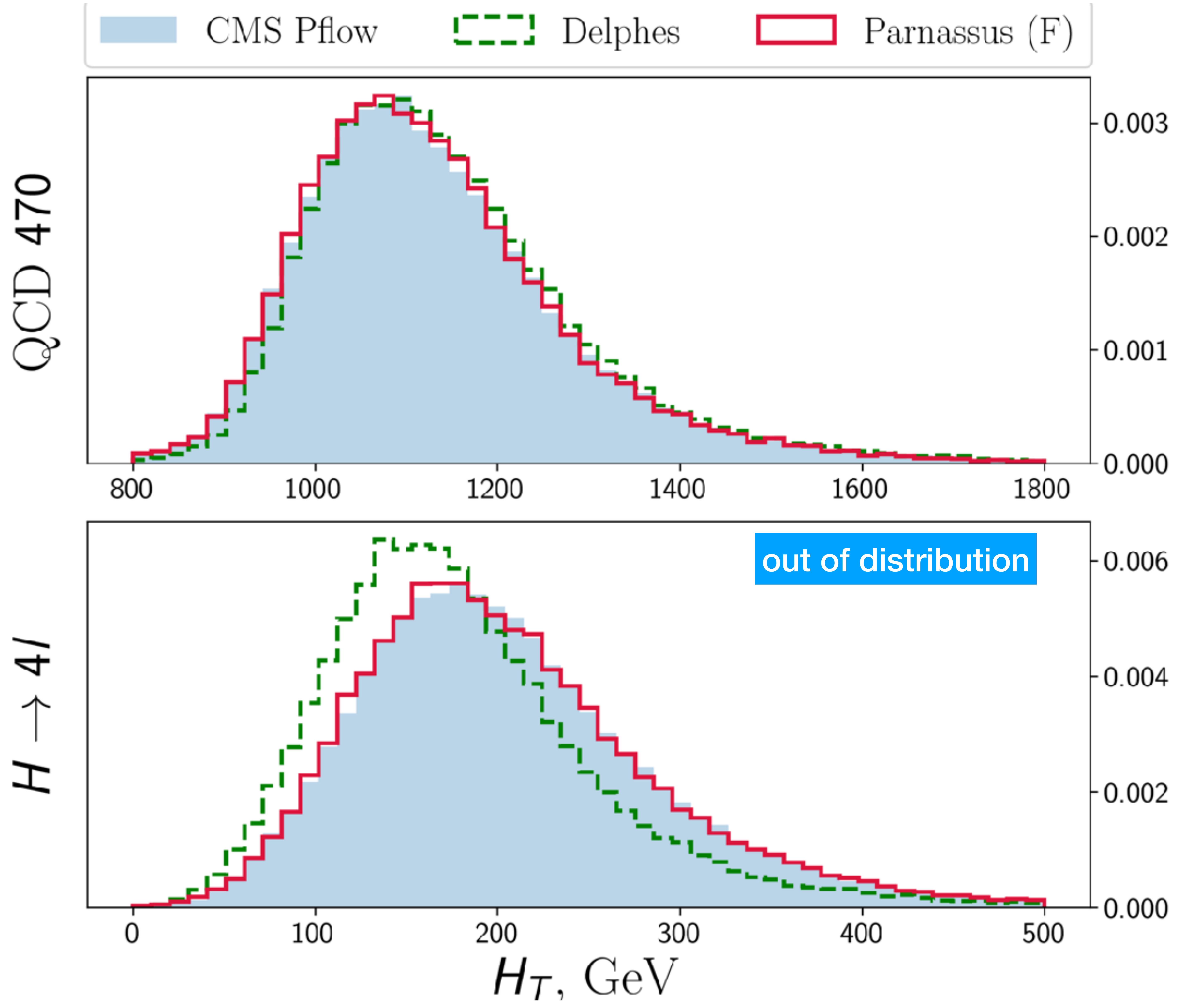
Event Level Performance



Event Level Performance

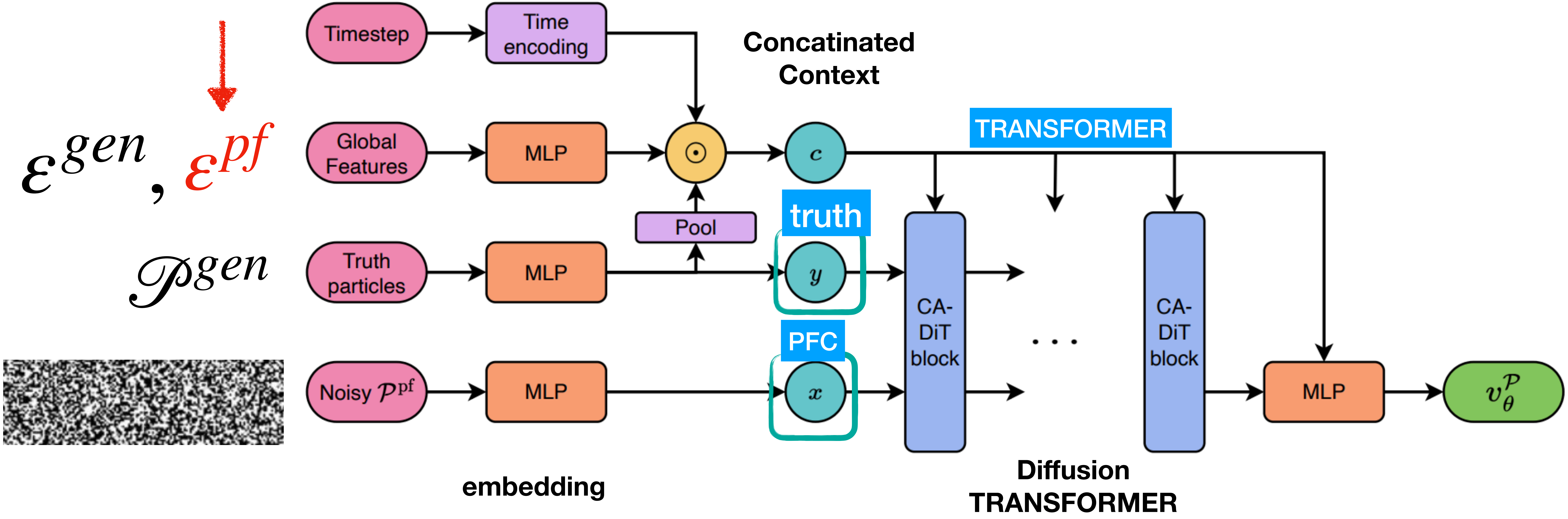


Event Level Performance

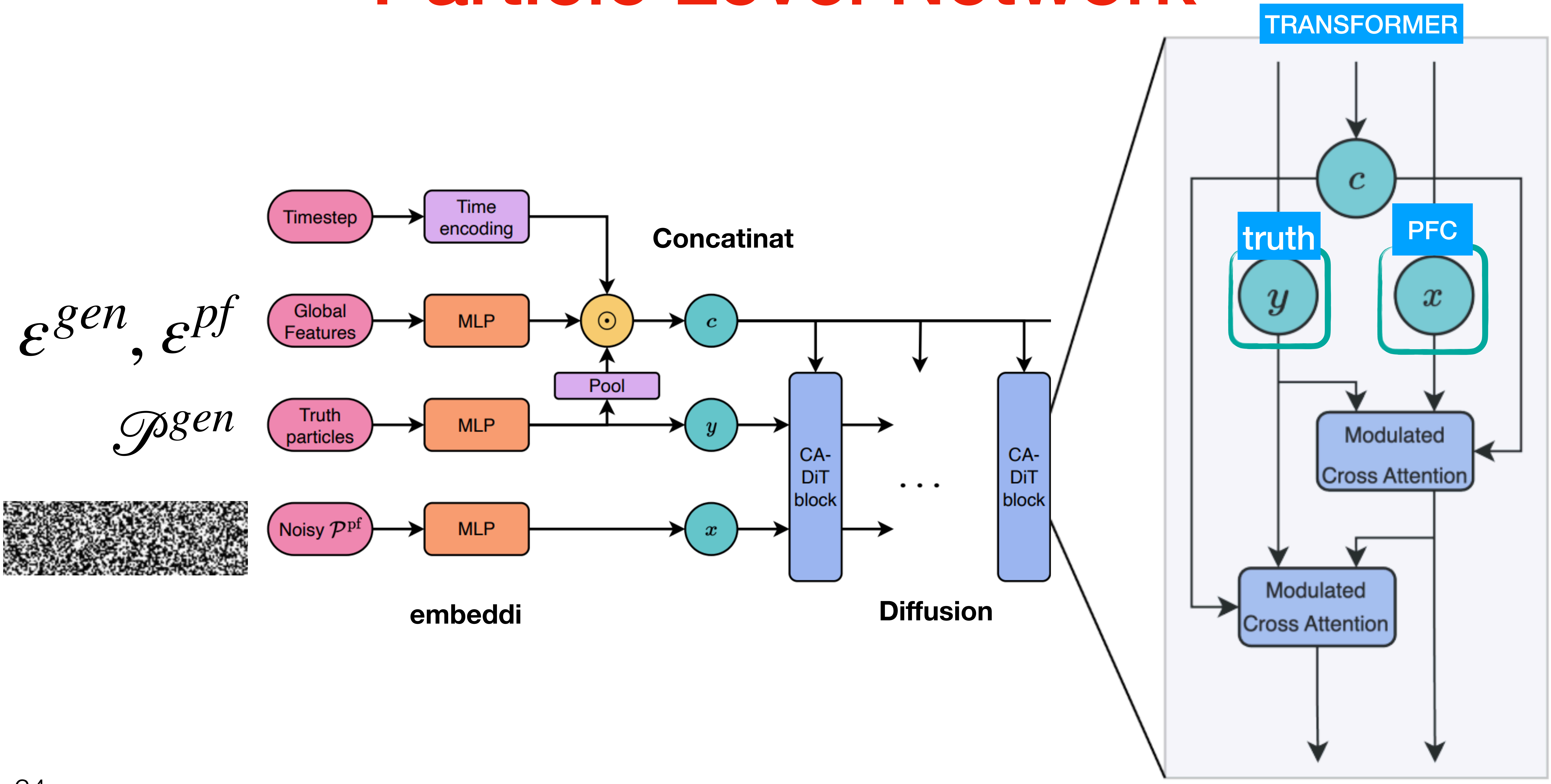


Particle Level Network

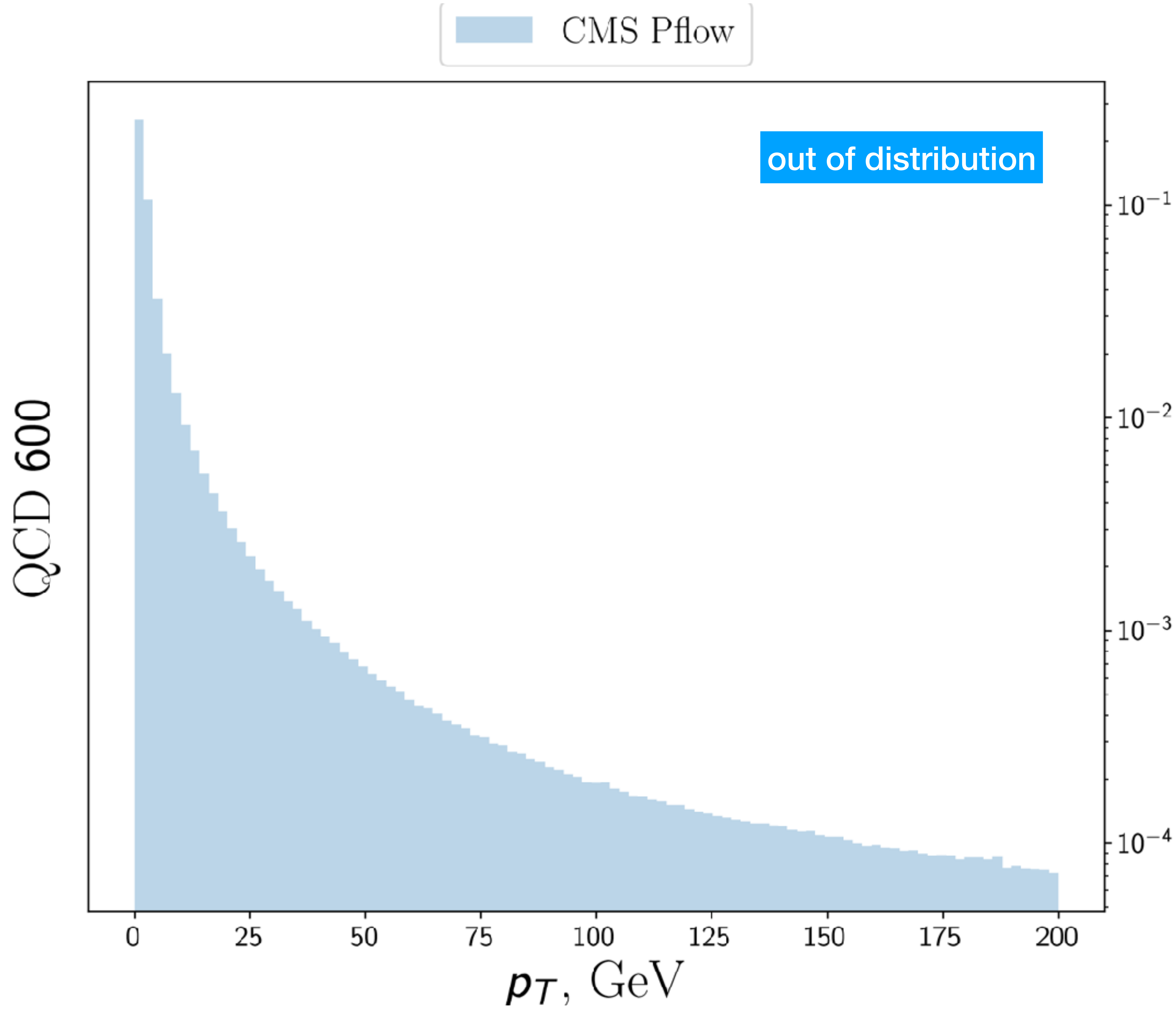
- Based on Diffusion Transformer



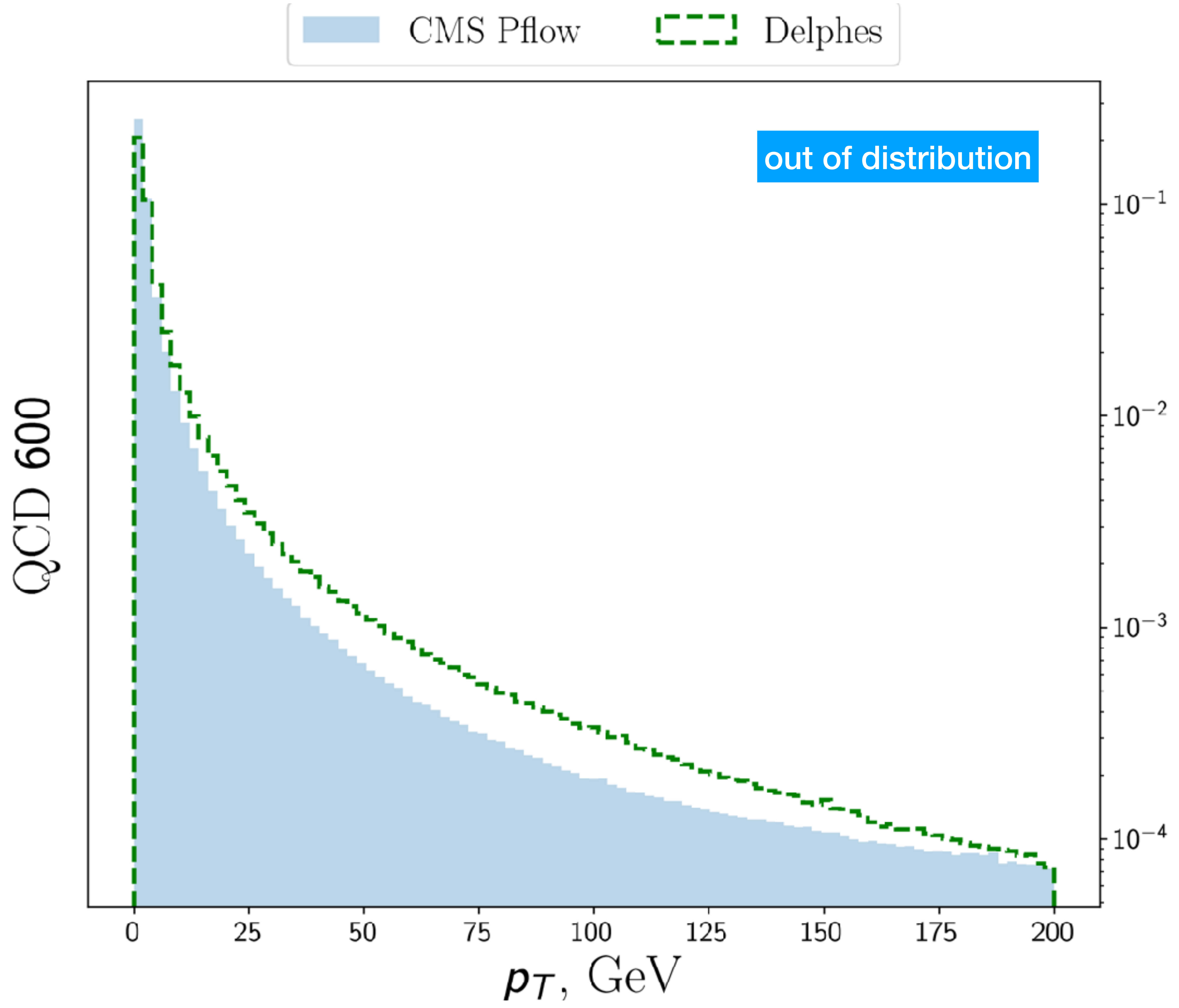
Particle Level Network



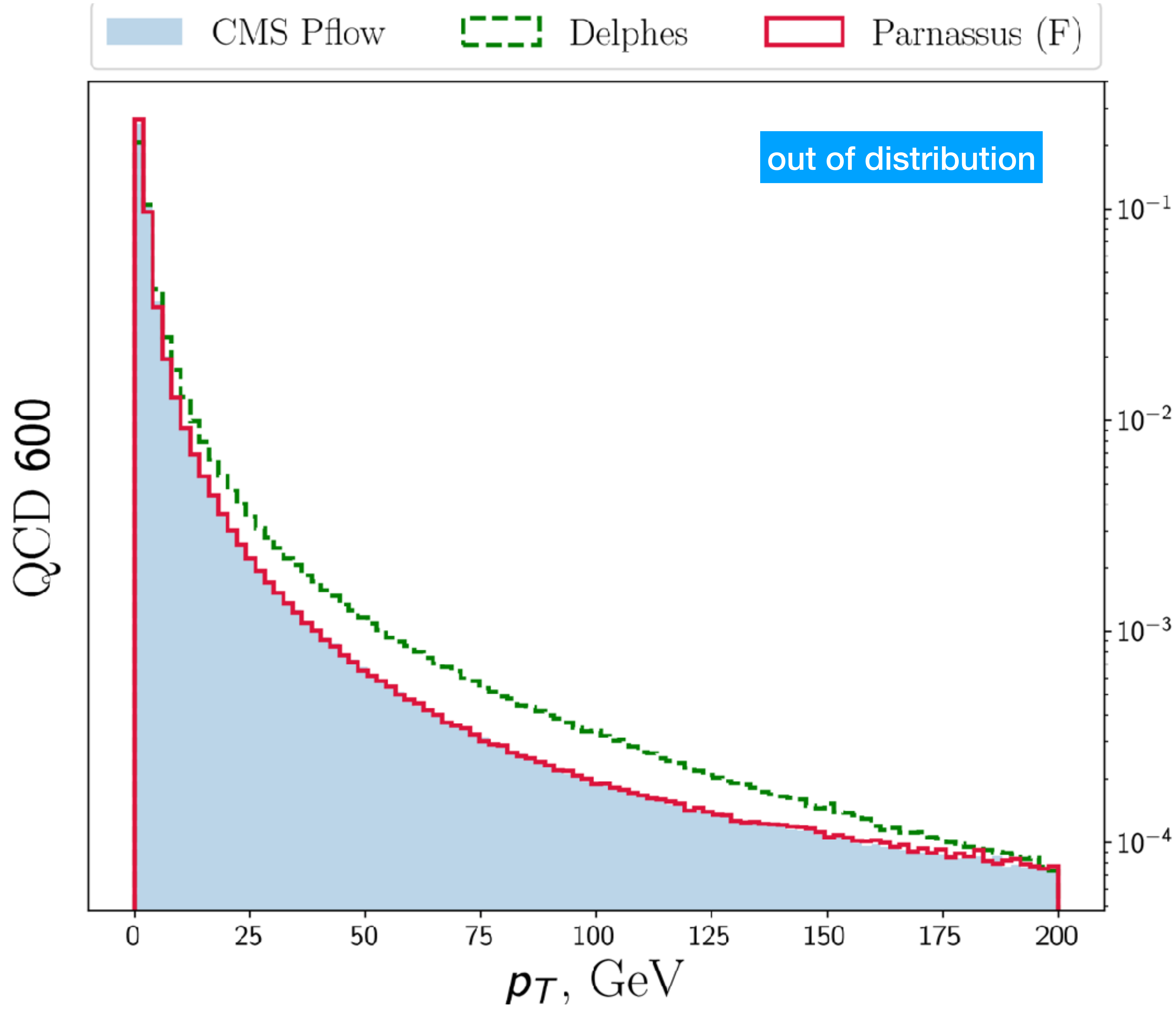
Particle Level Performance



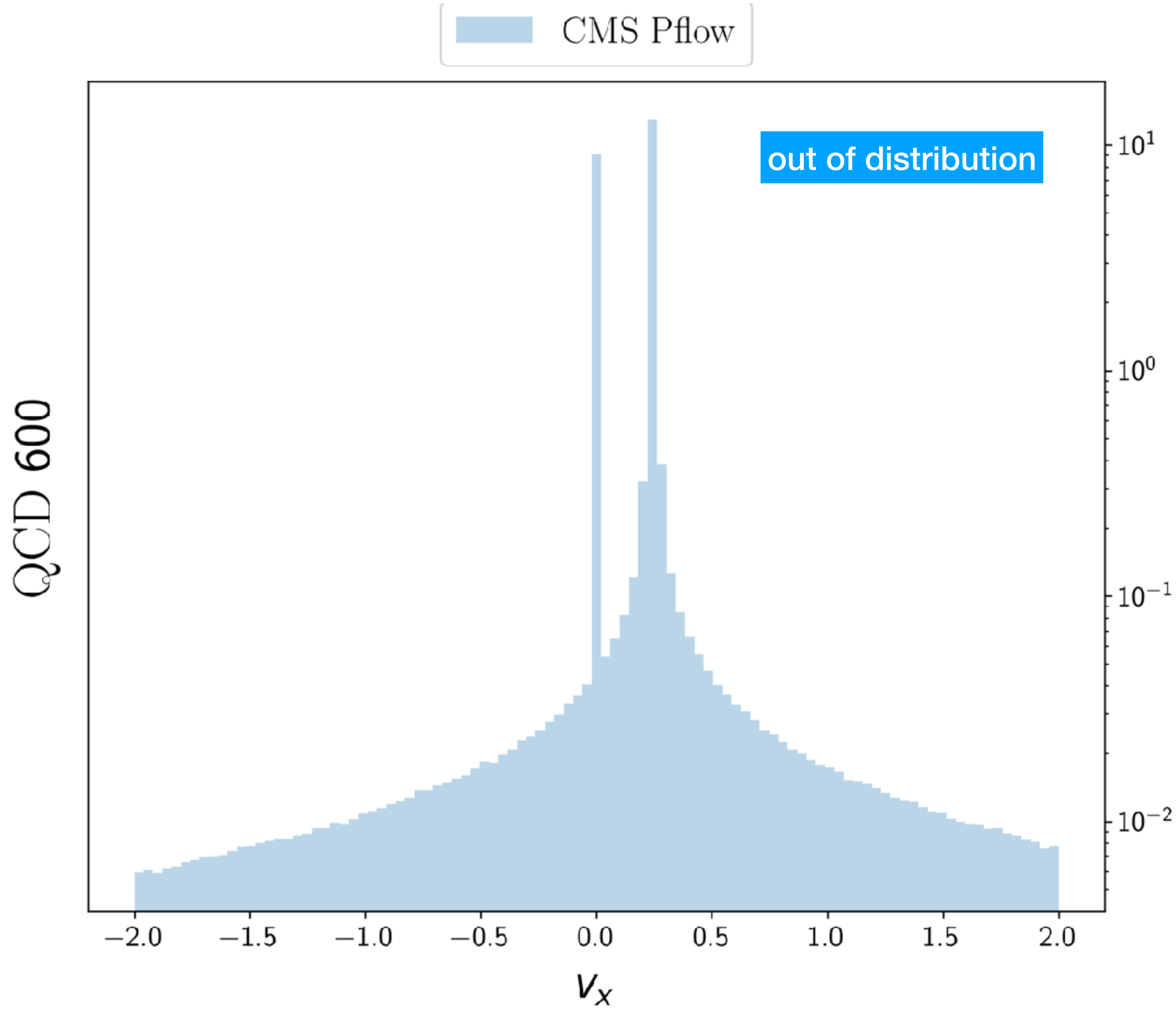
Particle Level Performance



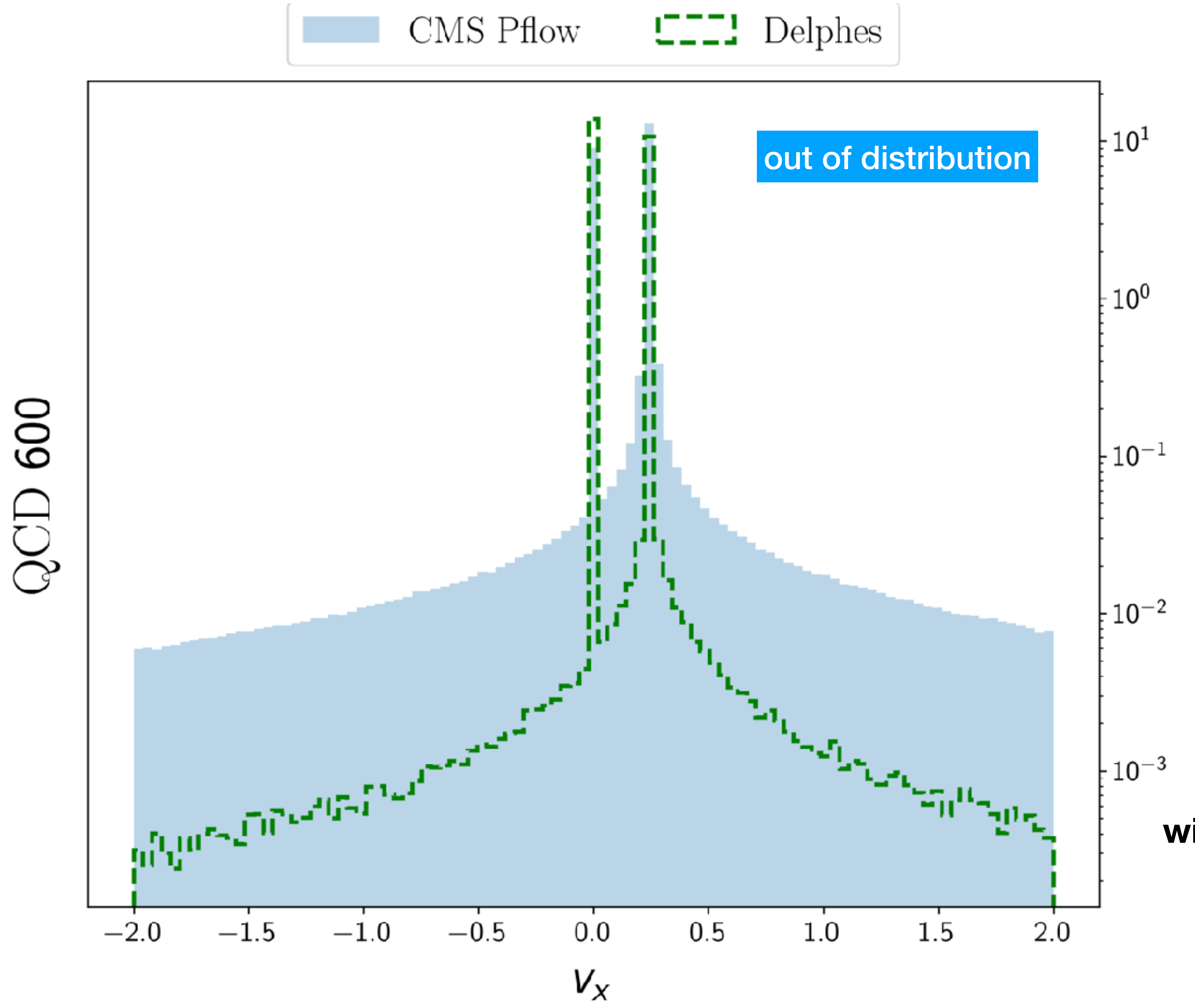
Particle Level Performance



Particle Level Performance

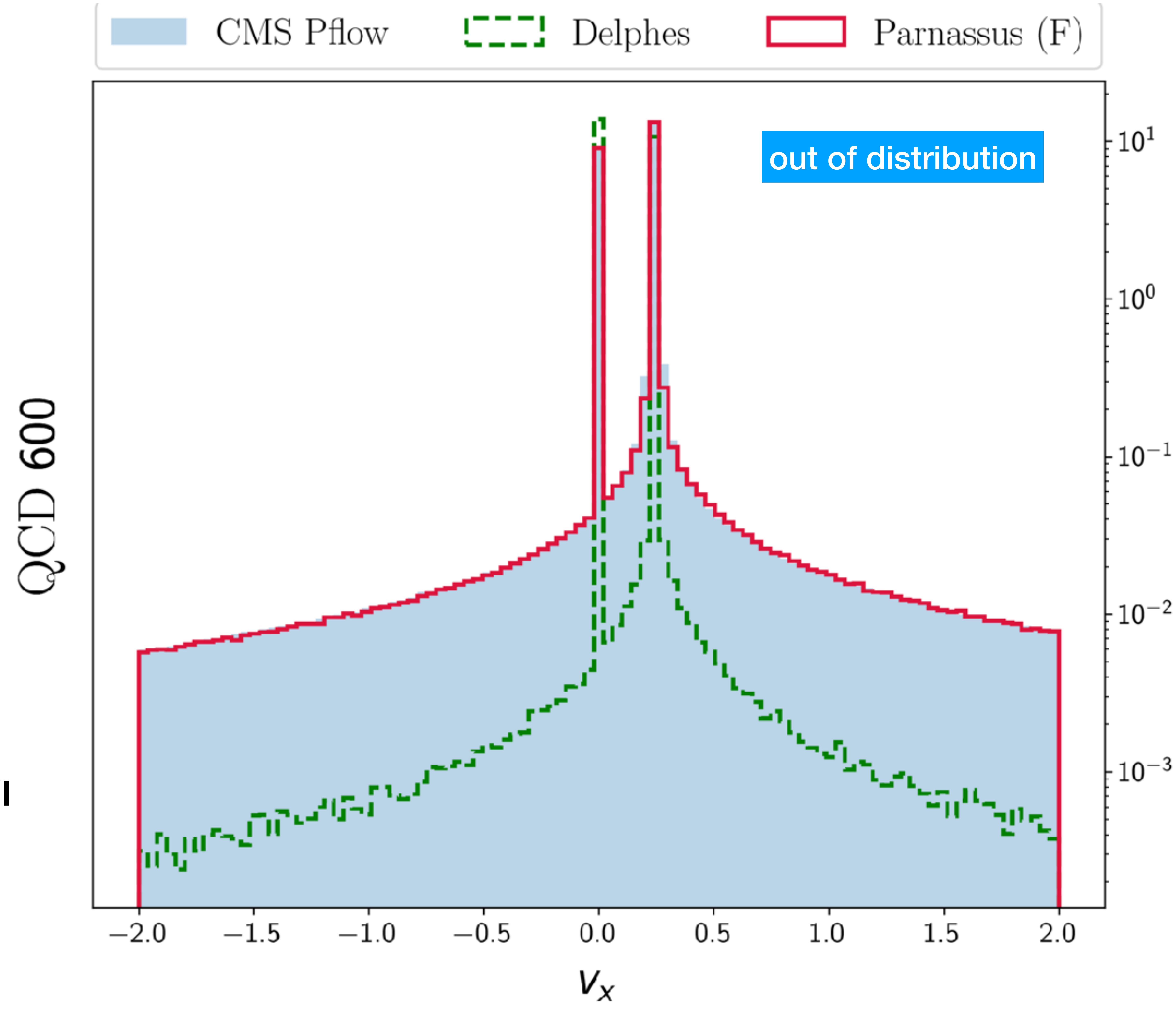


Particle Level Performance



Delphes cannot cope with vertex resolution modelling

Particle Level Performance

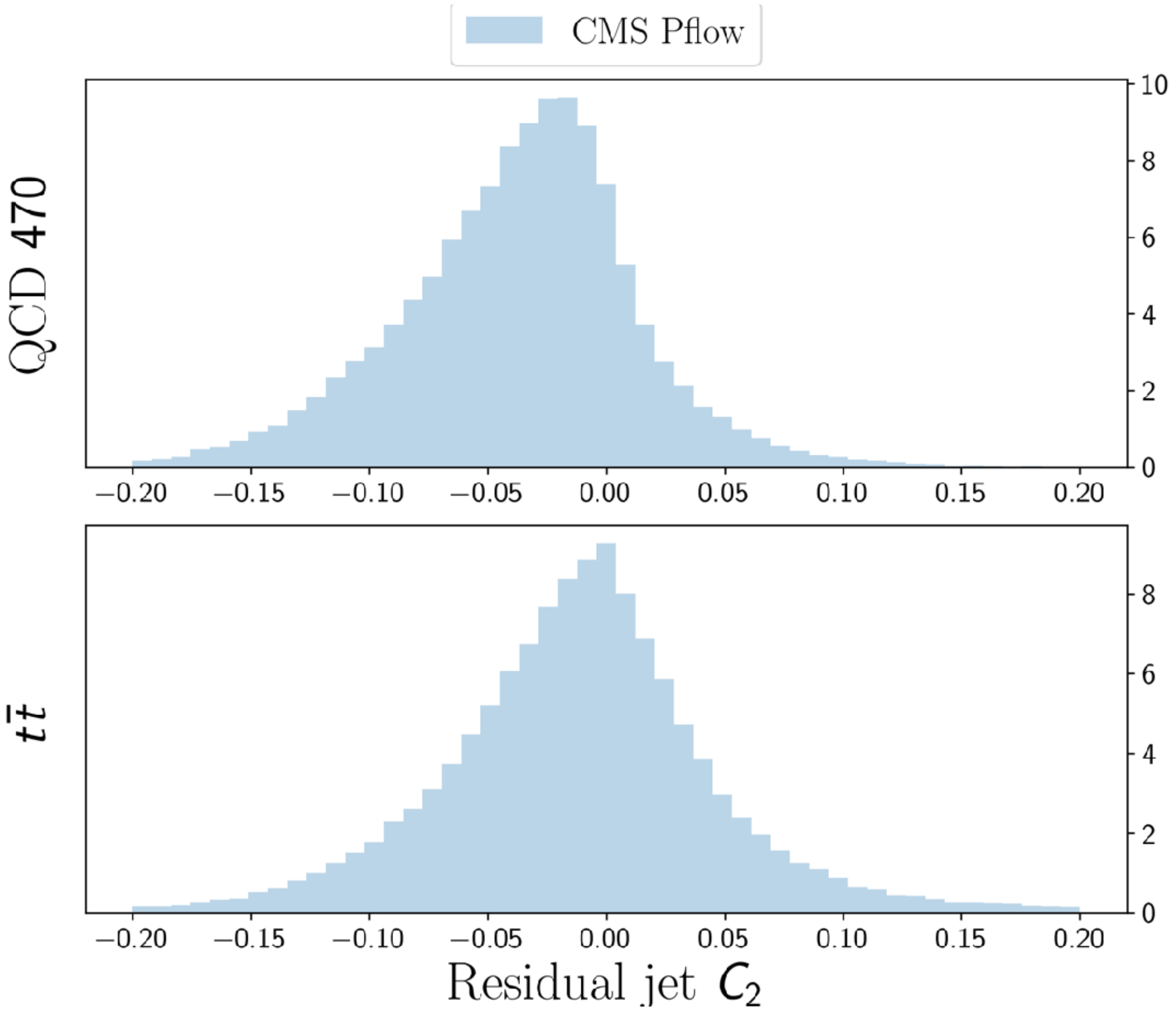


Parnassus models well transition regions

Jet Substructure Performance

Jet Substructure Variables are Sensitive to the Radiation Pattern (angular correlations) within a Jet

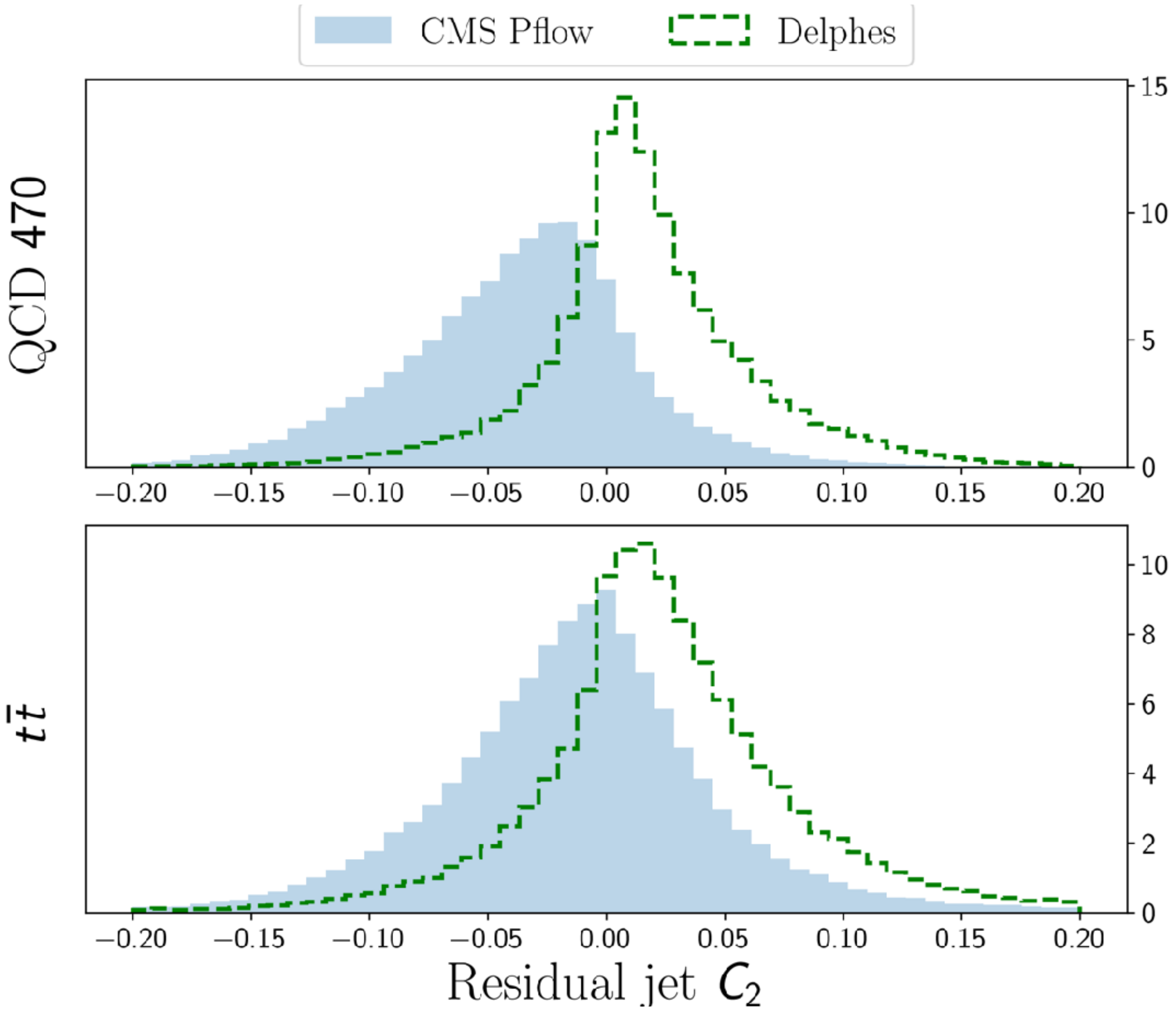
Low C2->Single Hard Core (q/g jets), High C2->Resolved Substructure (W->qq)



Jet Substructure Performance

Jet Substructure Variables are Sensitive to the Radiation Pattern (angular correlations) within a Jet

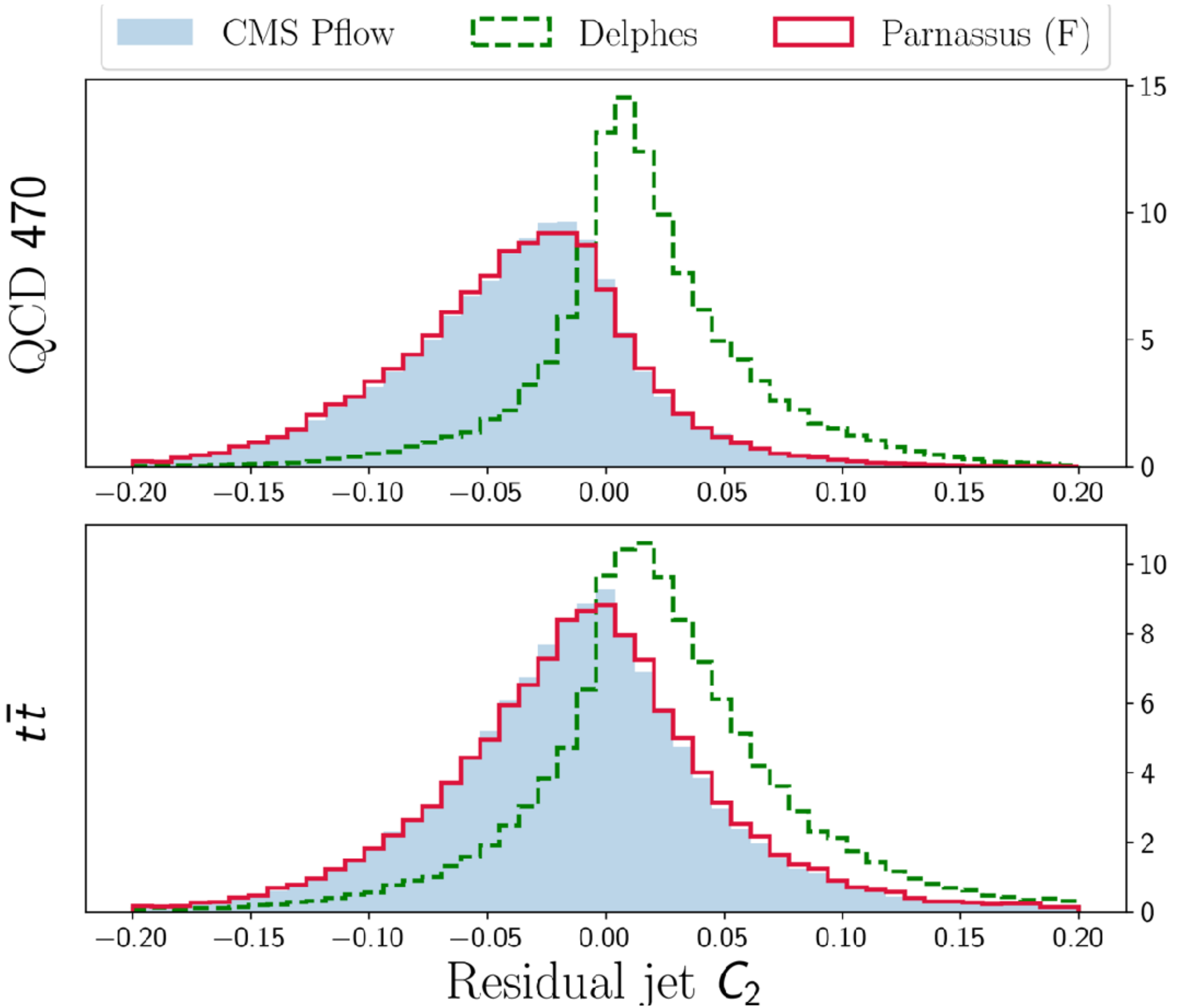
Low C2->Single Hard Core (q/g jets),
High C2->Resolved Substructure (W->qq)



Jet Substructure Performance

Jet Substructure Variables are Sensitive to the Radiation Pattern (angular correlations) within a Jet

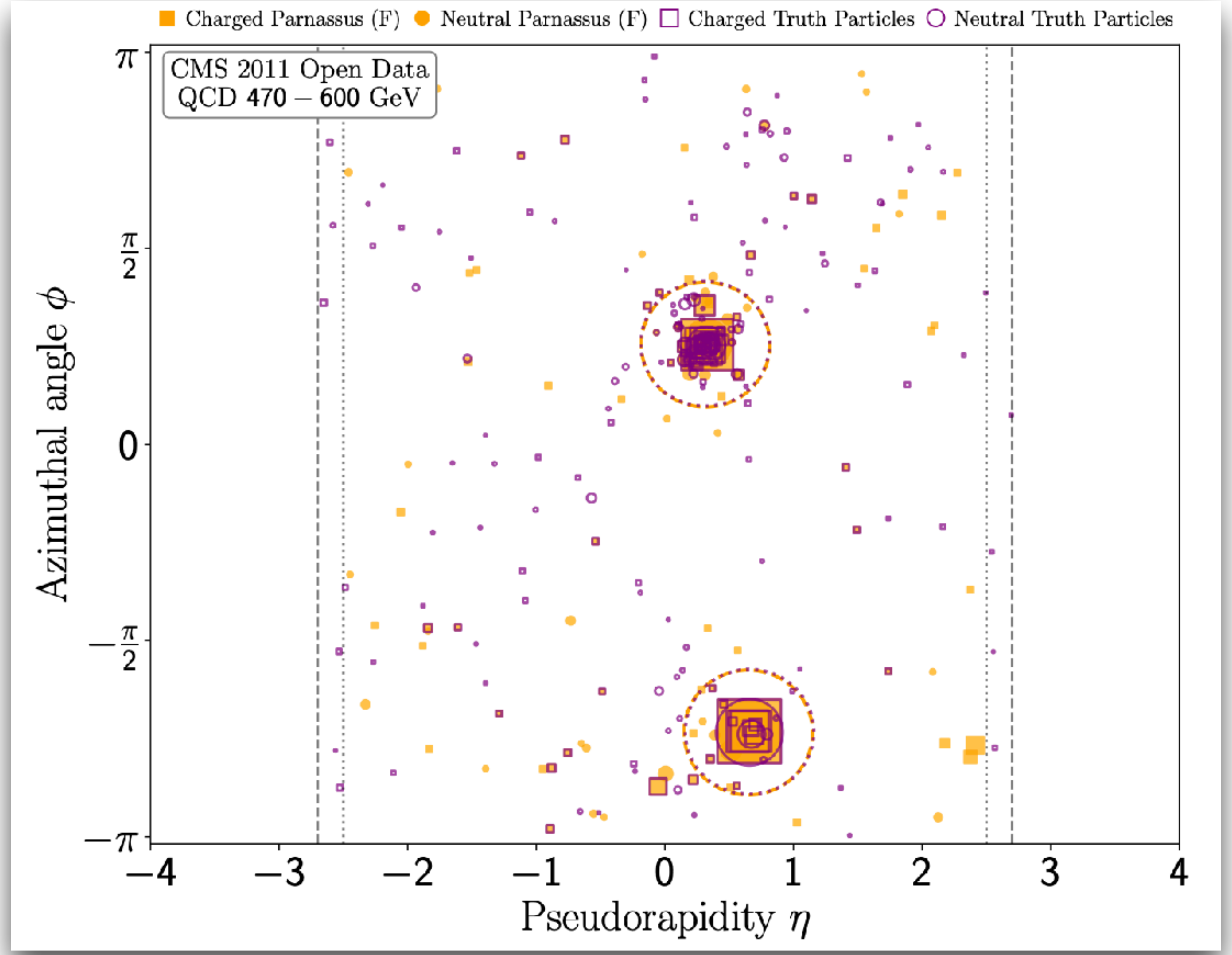
Low C2->Single Hard Core (q/g jets), High C2->Resolved Substructure (W->qq)



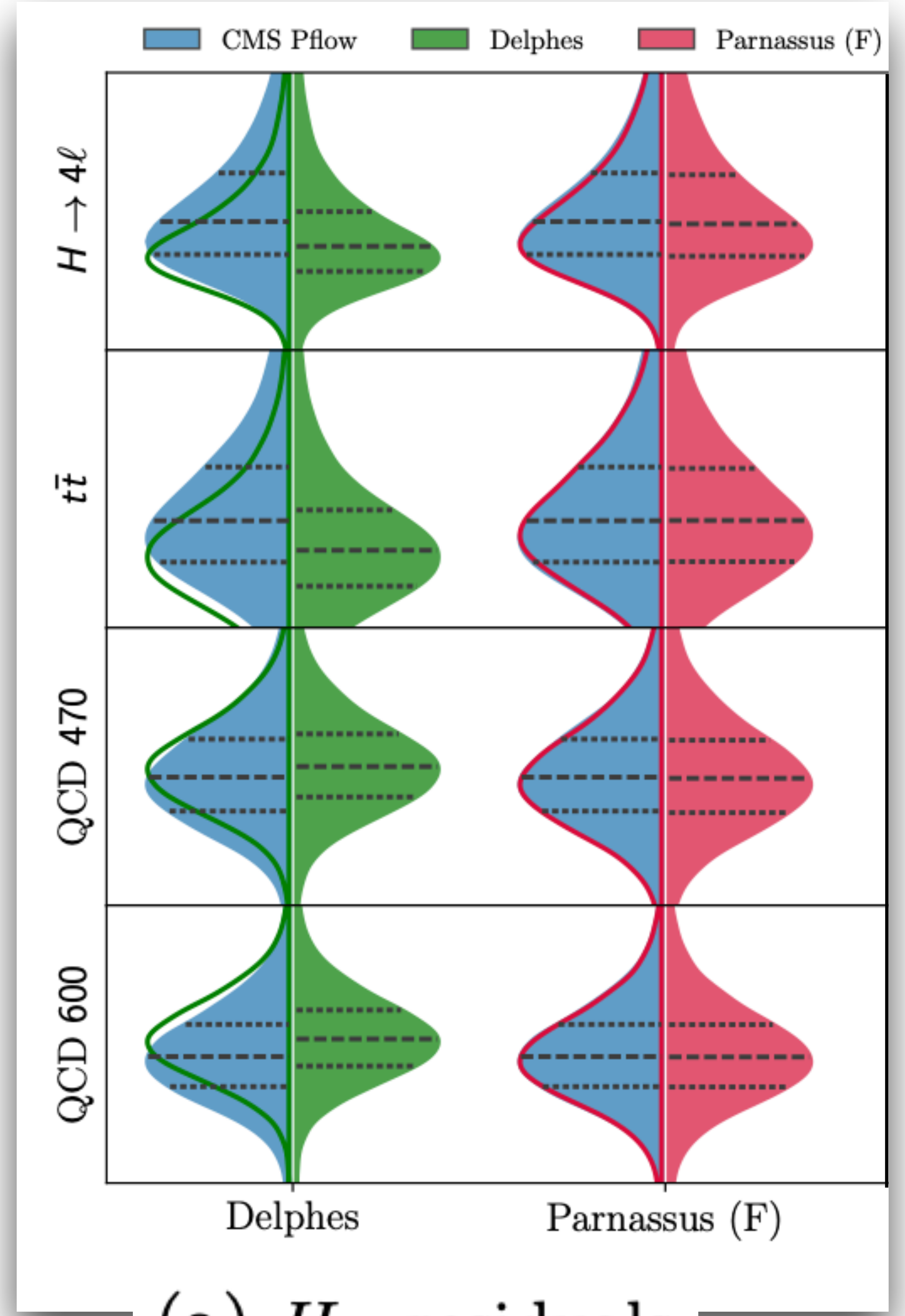
Performance Summary

Charged Parnassus (F)
Charged Truth Particles

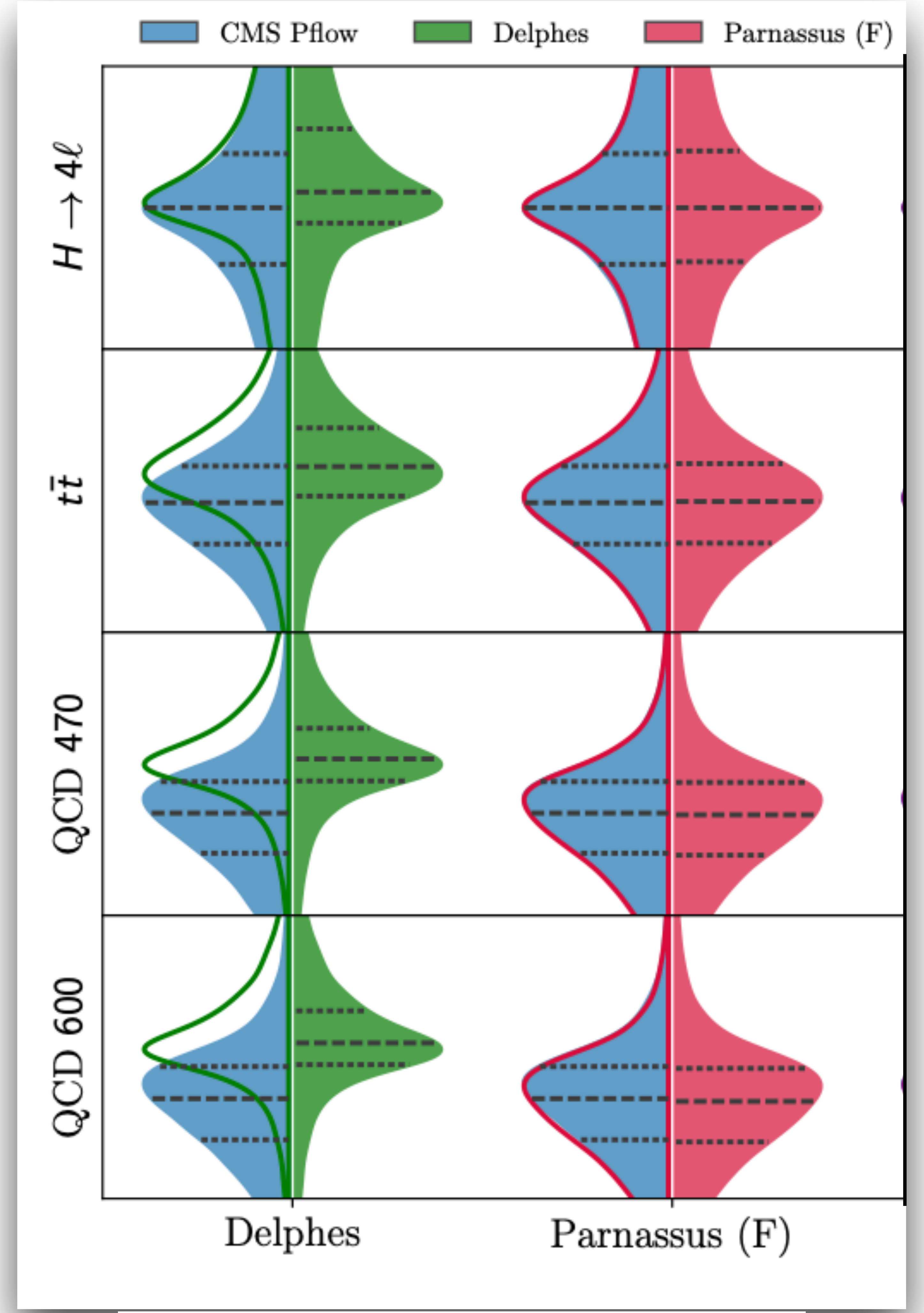
Neutral Parnassus (F)
Neutral Truth Particles



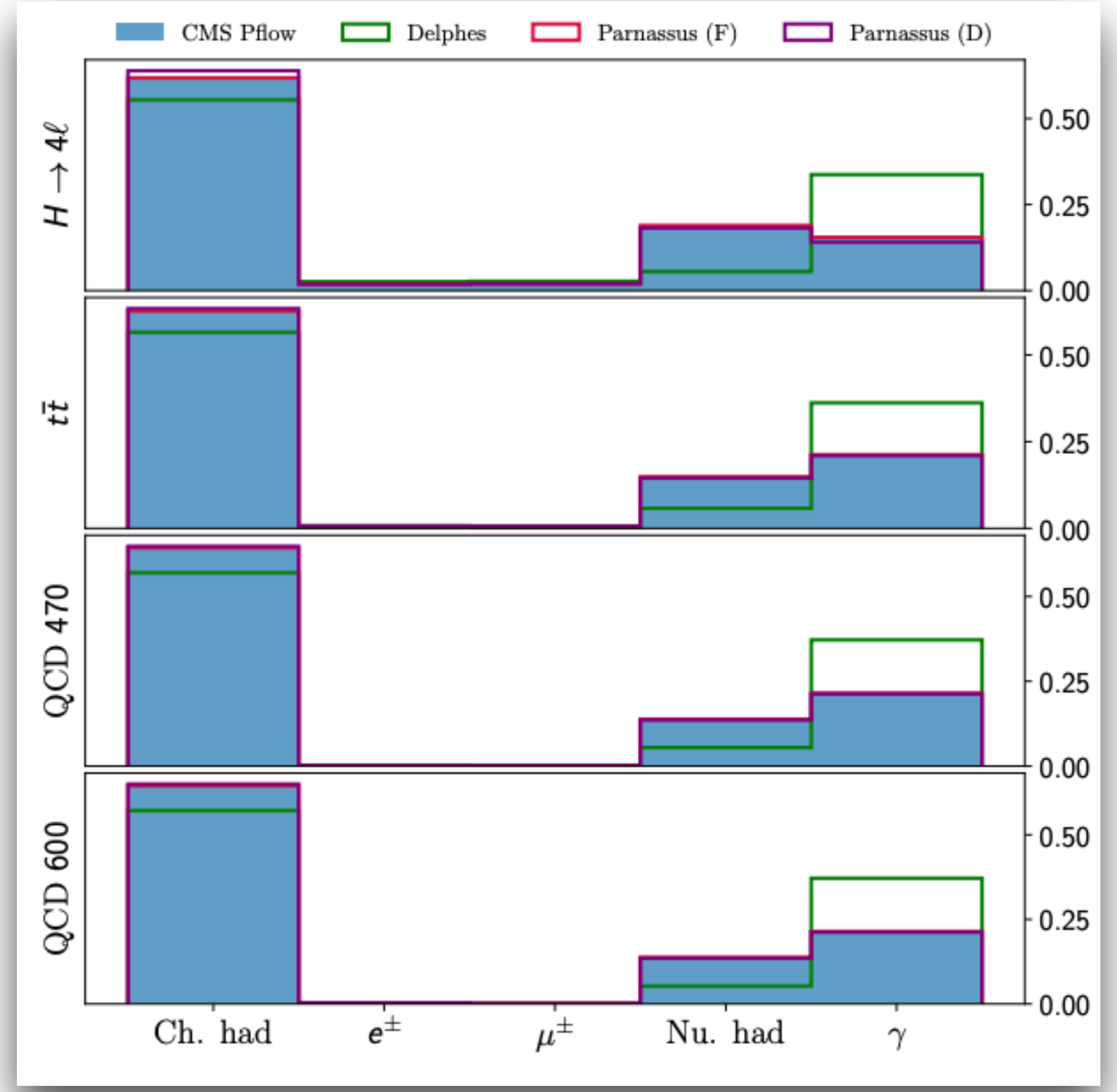
Performance Summary



(a) H_T residuals

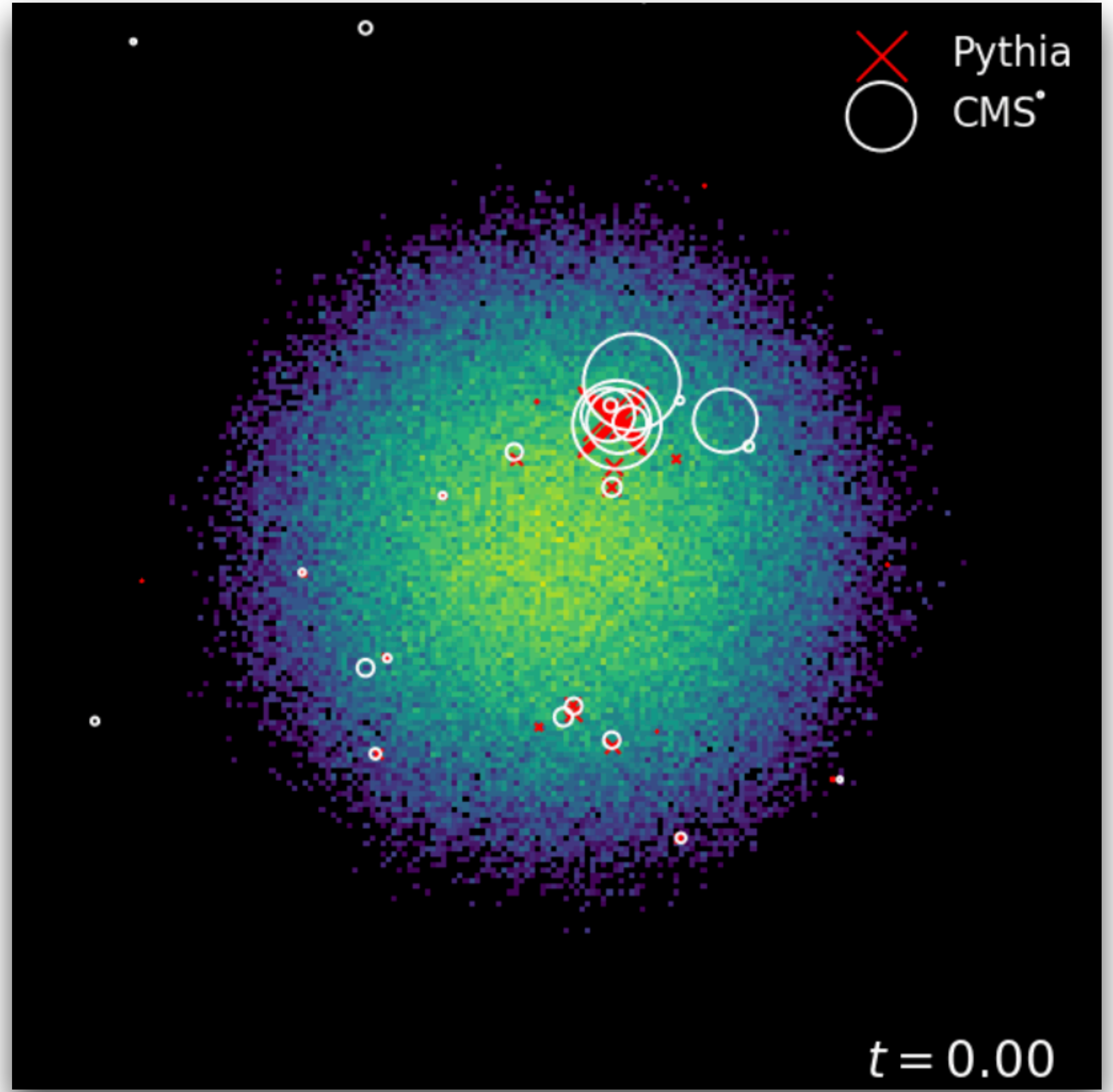


(b) Jet C_2 residuals



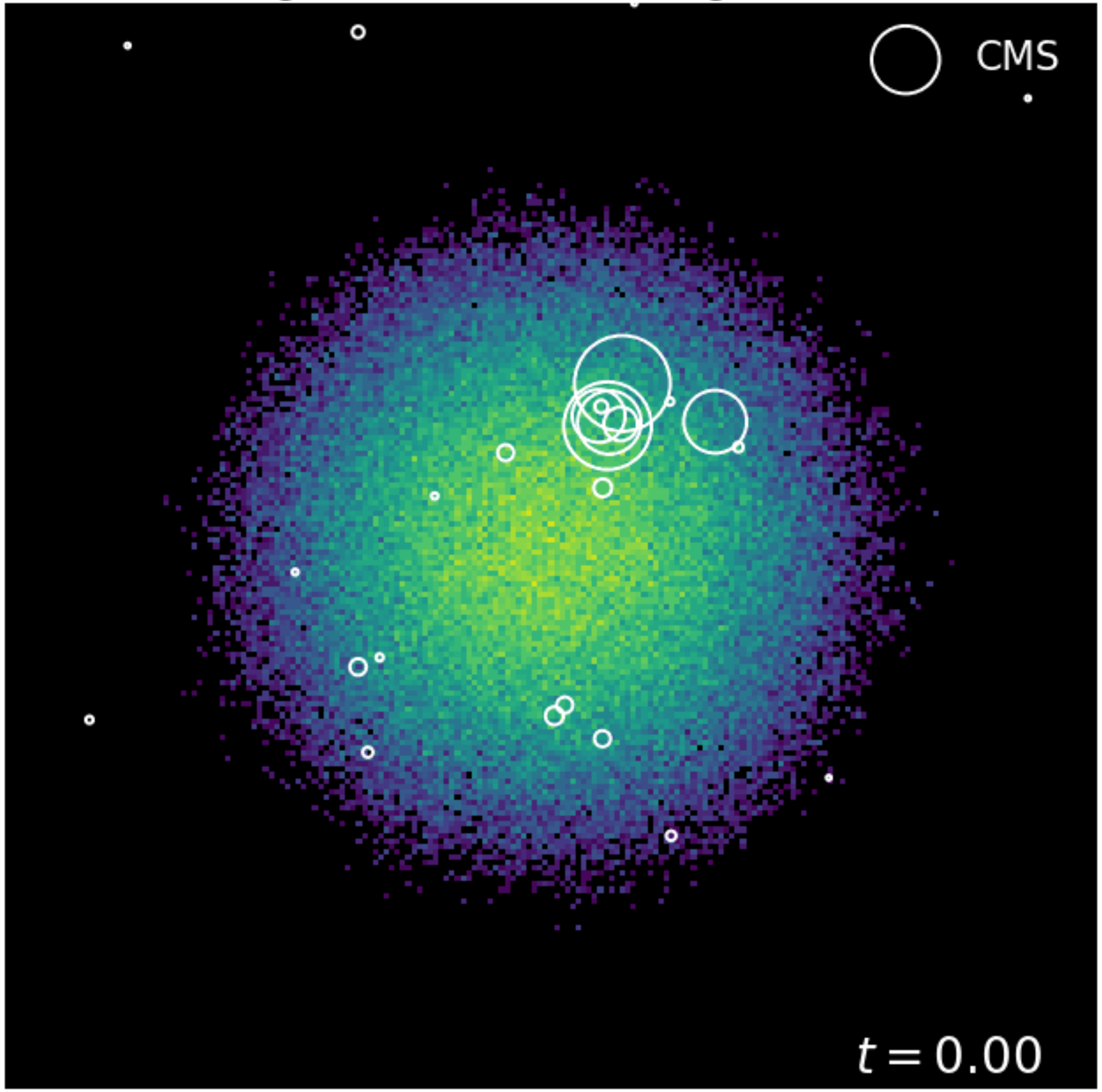
(c) Reconstructed particle classes

The Magic using Replicas



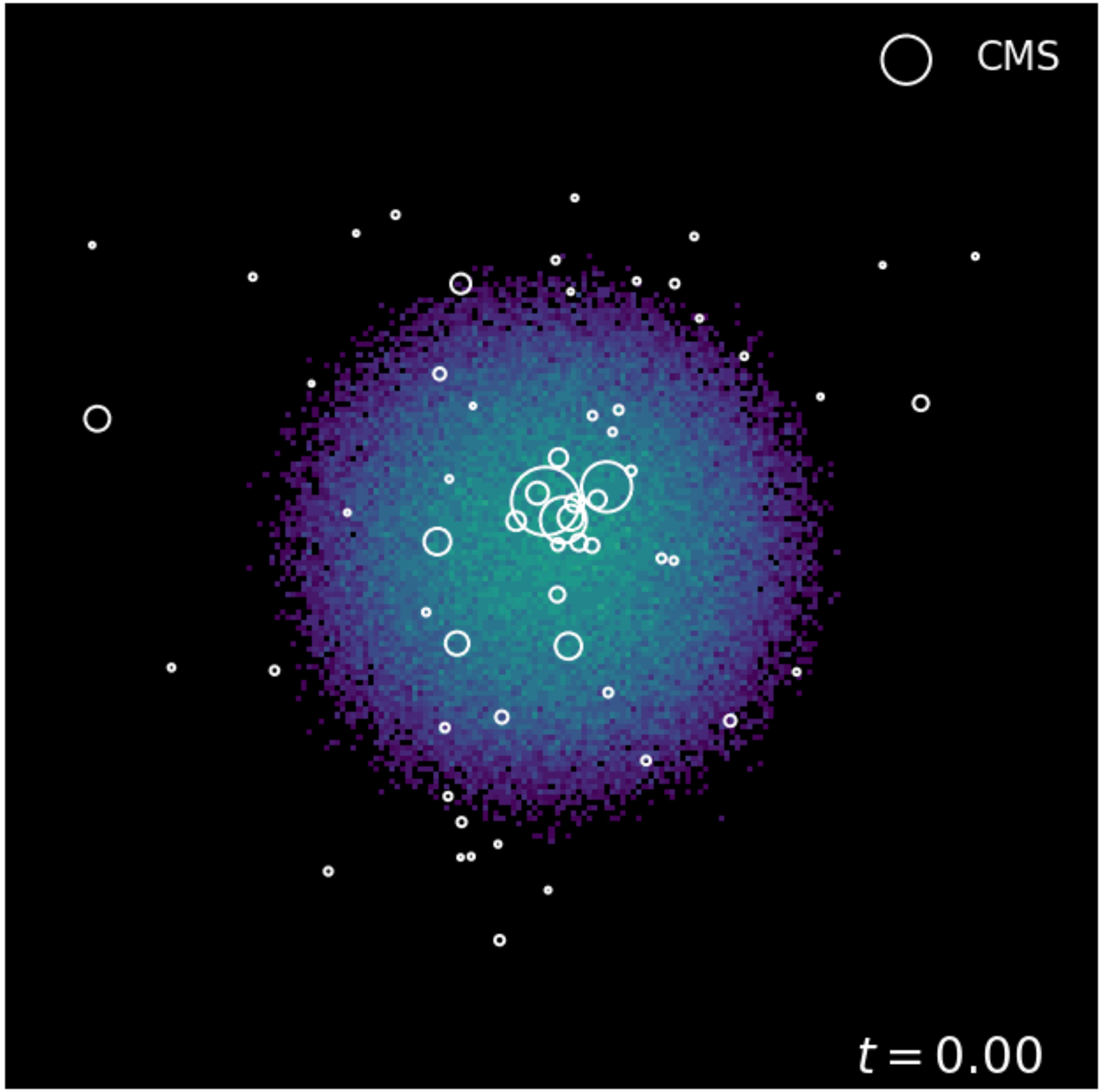
Animation by Etienne Dreyer

The Magic using Replicas



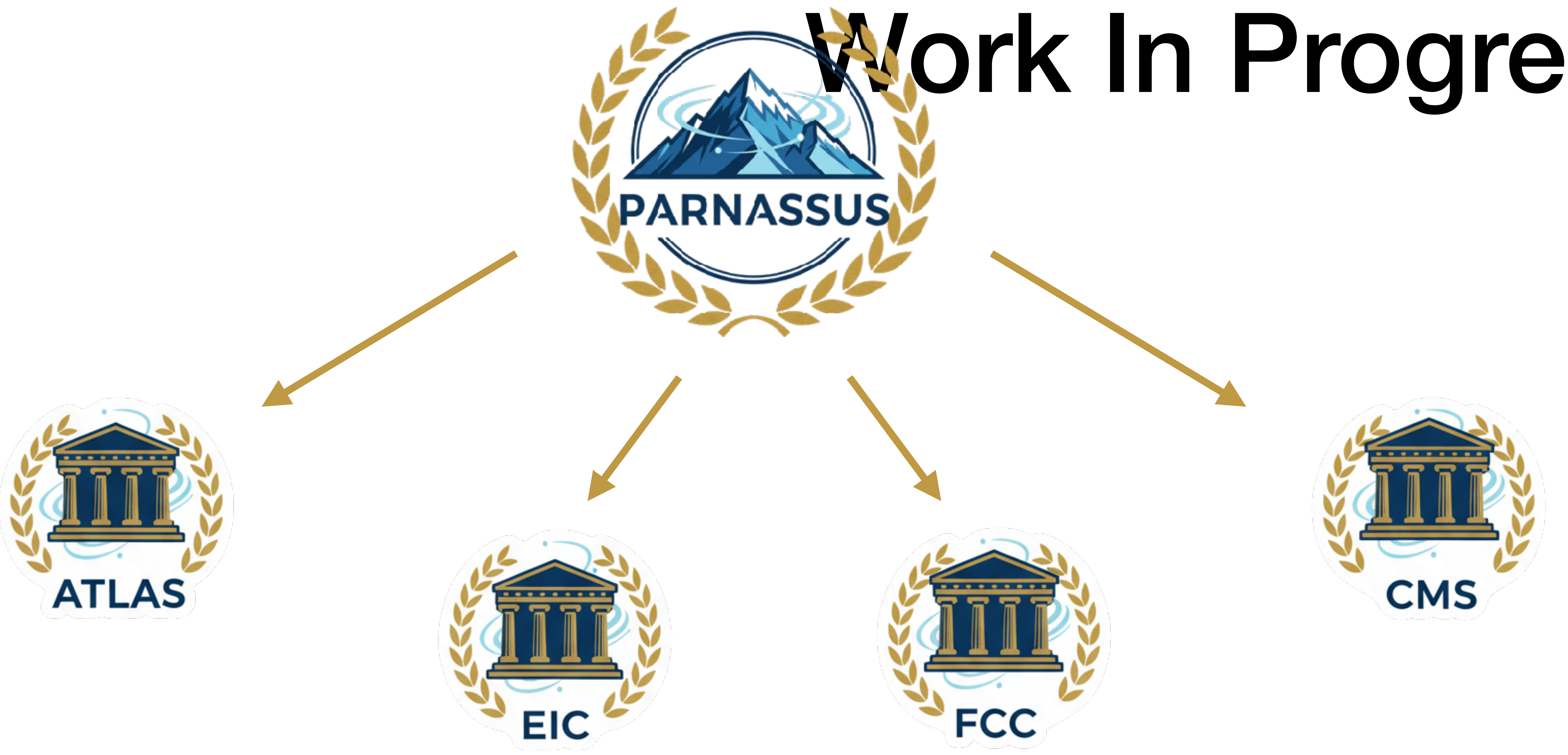
Animation by Etienne Dreyer

The Magic using Replicas



Animation by Etienne Dreyer

Work In Progress



Common standalone Python-only framework

Separate model checkpoint for every experiment

Open and easy-to-use for non-expert

Conclusions I/IV

- **Accuracy: Parnassus achieves simulation accuracy very close to full simulation (GEANT4+PF) across a wide range of observables.**
  Distributions of energies, angles, multiplicities, and jet variables are all well-modeled, significantly better than the traditional smeared base fast-sim (Delphes)

- **Speed: Flow matching provides a substantial speed-up in generating events (compared to Full Sim).**
  In practice, this means fast turn-around and possibility of online or on-demand simulation in analysis workflows.

- **Generalization: The model successfully generalized to processes and energy ranges beyond its training**,
  showing promise that it can be trained on a representative sample and then used for many physics studies. It was not narrowly overfit to one process.

Conclusion II/IV

- **Particle-Flow as a Learning Target:** We have shown that complex reconstruction (which involves tracking, calorimetry, clustering) can be approximated by a learned function. **The ML model effectively learned to "reconstruct" an event like the PF algorithm does, in one go.** This is an interesting validation of AI techniques on a structured physics task.

- **Flow Matching as a Technique: The success of Parnassus (F) highlights the power of flow matching in physics simulation.** It could pave the way for other uses, e.g., fast simulation of other detectors or even cosmological simulations, where you want to morph one distribution to another quickly.

Conclusions III/IV

- We demonstrated **a successful marriage of cutting-edge AI (flow-matching transformers) with HEP simulation.**

- It achieves **the Holy Grail of fast simulation: significant speedup with full-sim accuracy.**
  This could become a foundation for the next generation of simulation tools in high-energy physics, complementing traditional methods and enabling the community to tackle the computational challenges of future experiments.

Conclusions

- **Current and Future work:**
  We are now establishing a collaboration with ATLAS and CMS Simulation group to train Parnasus on the RUN 2/3 detectors response

- We are aiming at training Parnasus on prospective future detectors (FCC, CEPC, etc….)

- We believe Parnasus is the future of Fast Simulation

- A Friendly GUI will be available in the 1st quarter of 2026

Backup

84

Performance & Timings

	CFM		Delphes
Batch size/Time for 1 event	GPU	CPU	CPU
1	0.669	4.38	**0.0112**
10	0.0734	1.59	
100	0.0147	1.29	
1000	**0.0136**	-	

Only 5M parameters model

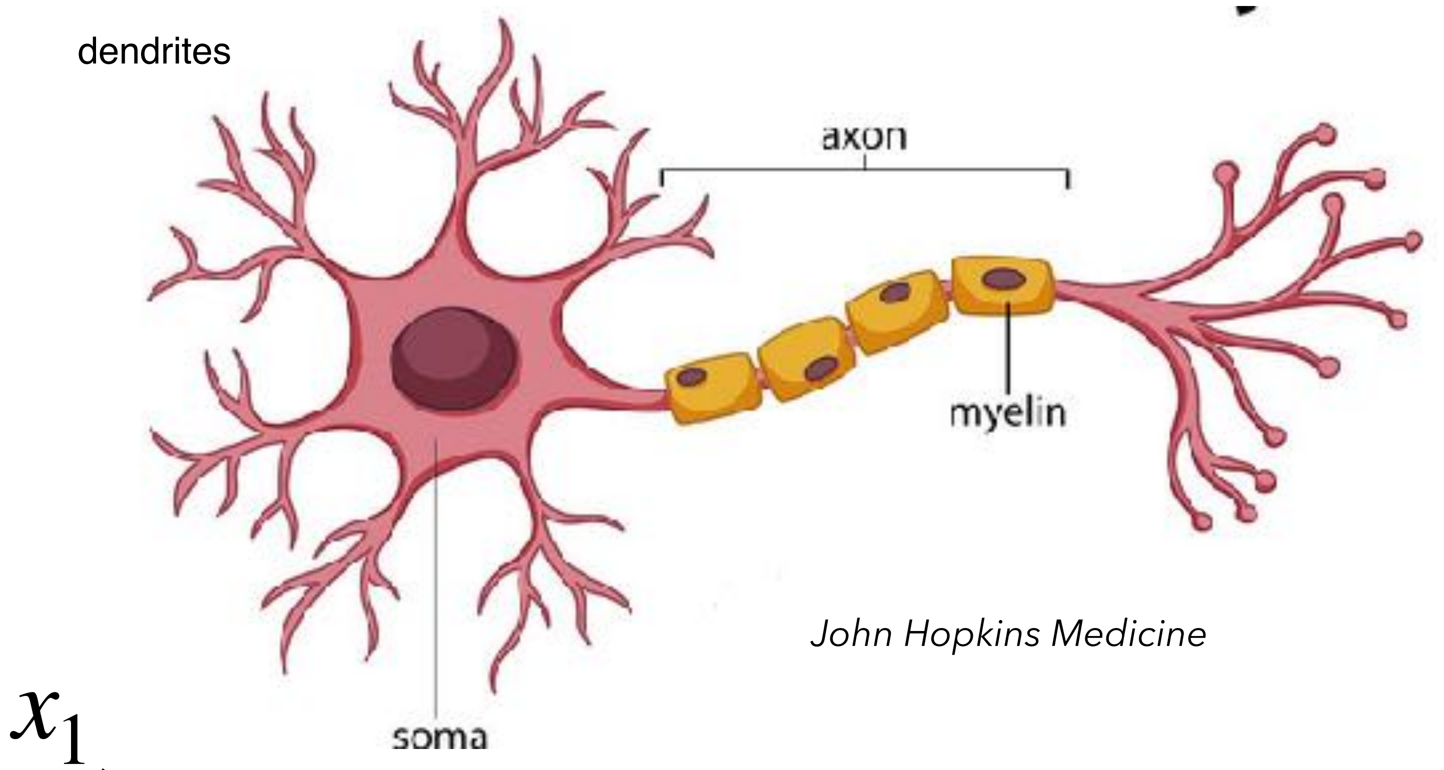
40 Flow-DPM Solver steps

Can achieve almost same performance with 20

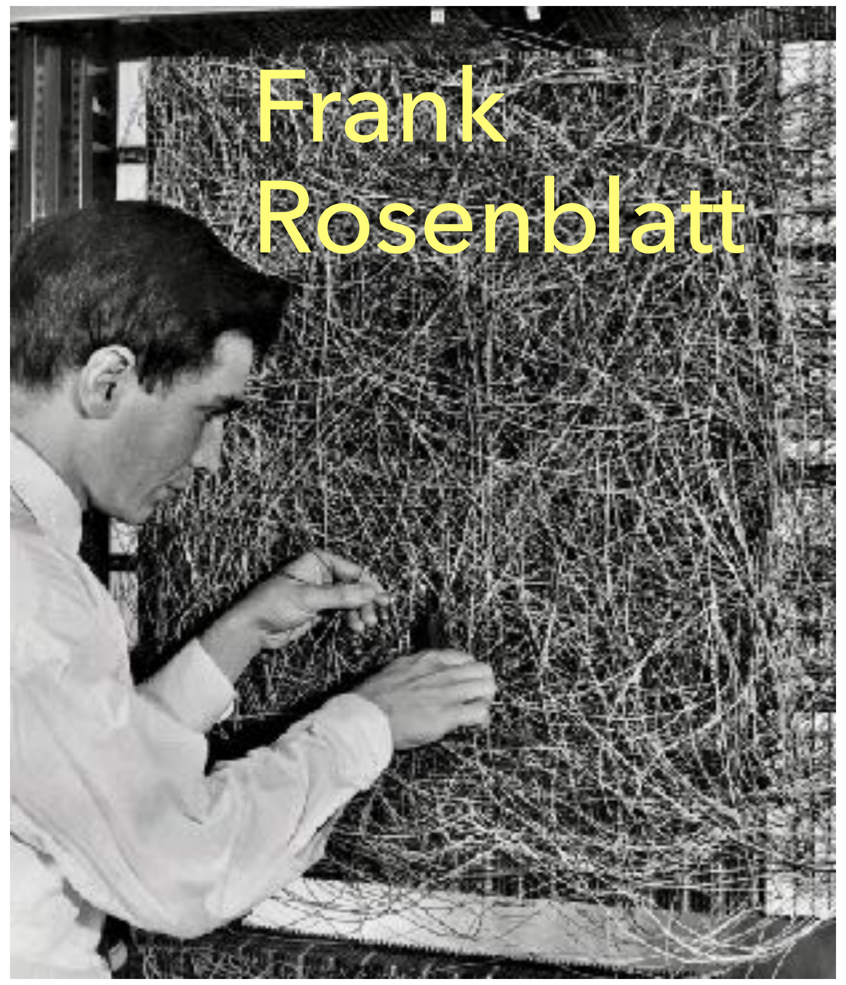
Can be optimized more?

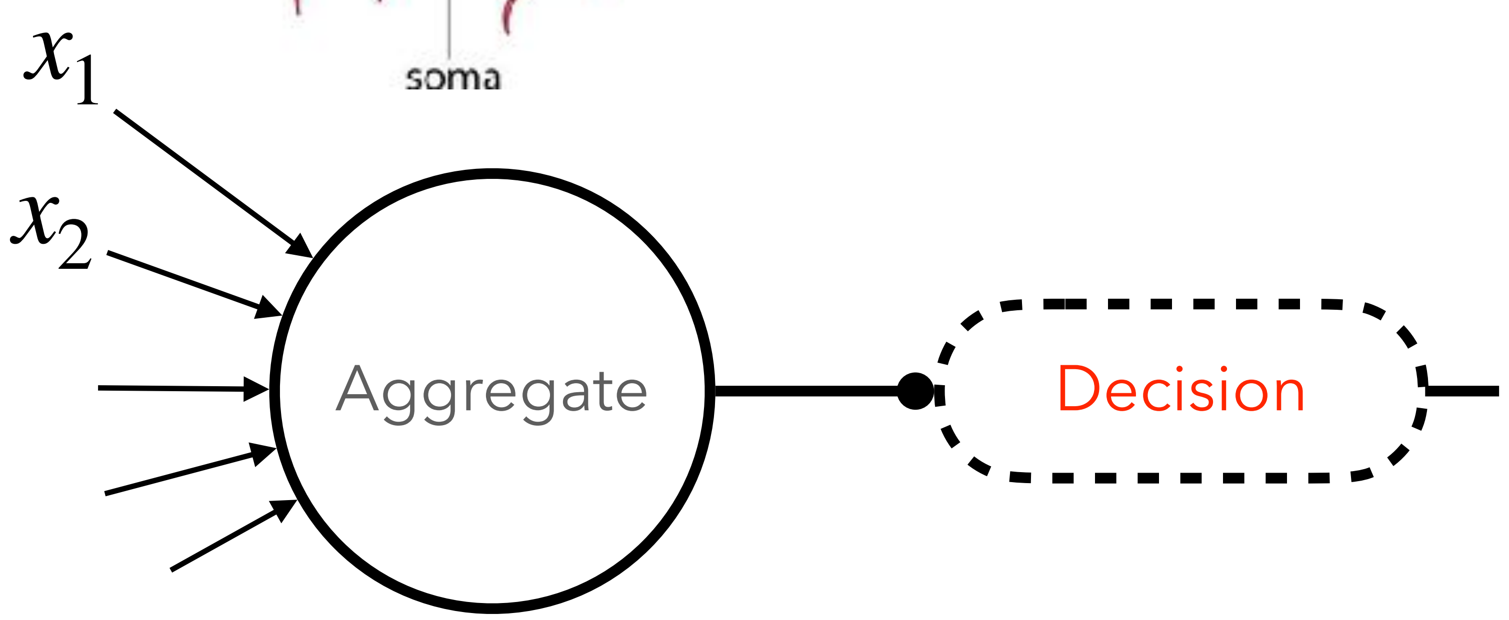
History of NNs

1943 McCulloch-Pitts neuron



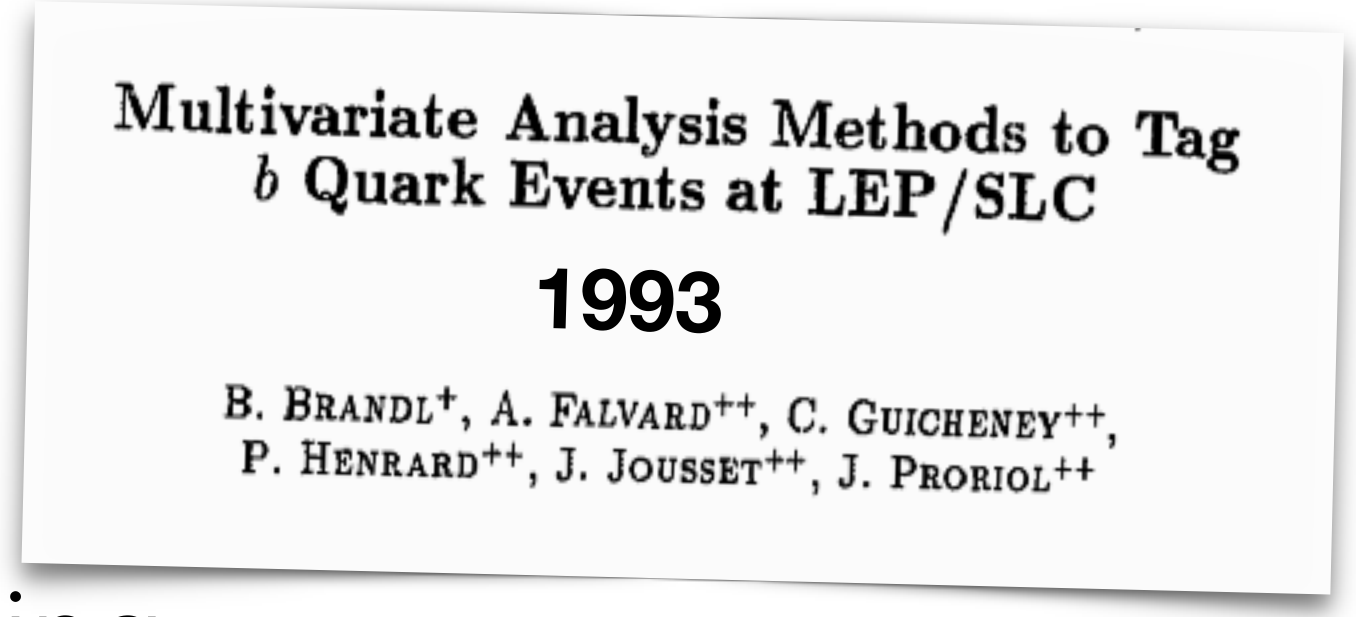
1958 Perceptron





"a perceptron may eventually be able to learn, make decisions, and translate languages"

1970s – '80s: "AI winter", ups & downs



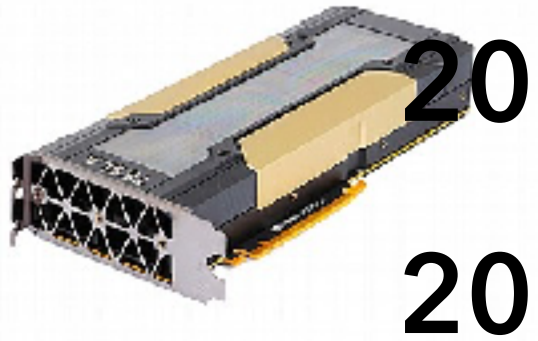
Multivariate Analysis Methods to Tag
b Quark Events at LEP/SLC

1993

B. Brandl[+], A. Falvard[++], C. Guicheney[++],
P. Henrard[++], J. Jousset[++], J. Proriol[++]

'90s recurrent models, computational barriers

2006 Hinton et al. kickstart modern deep learning

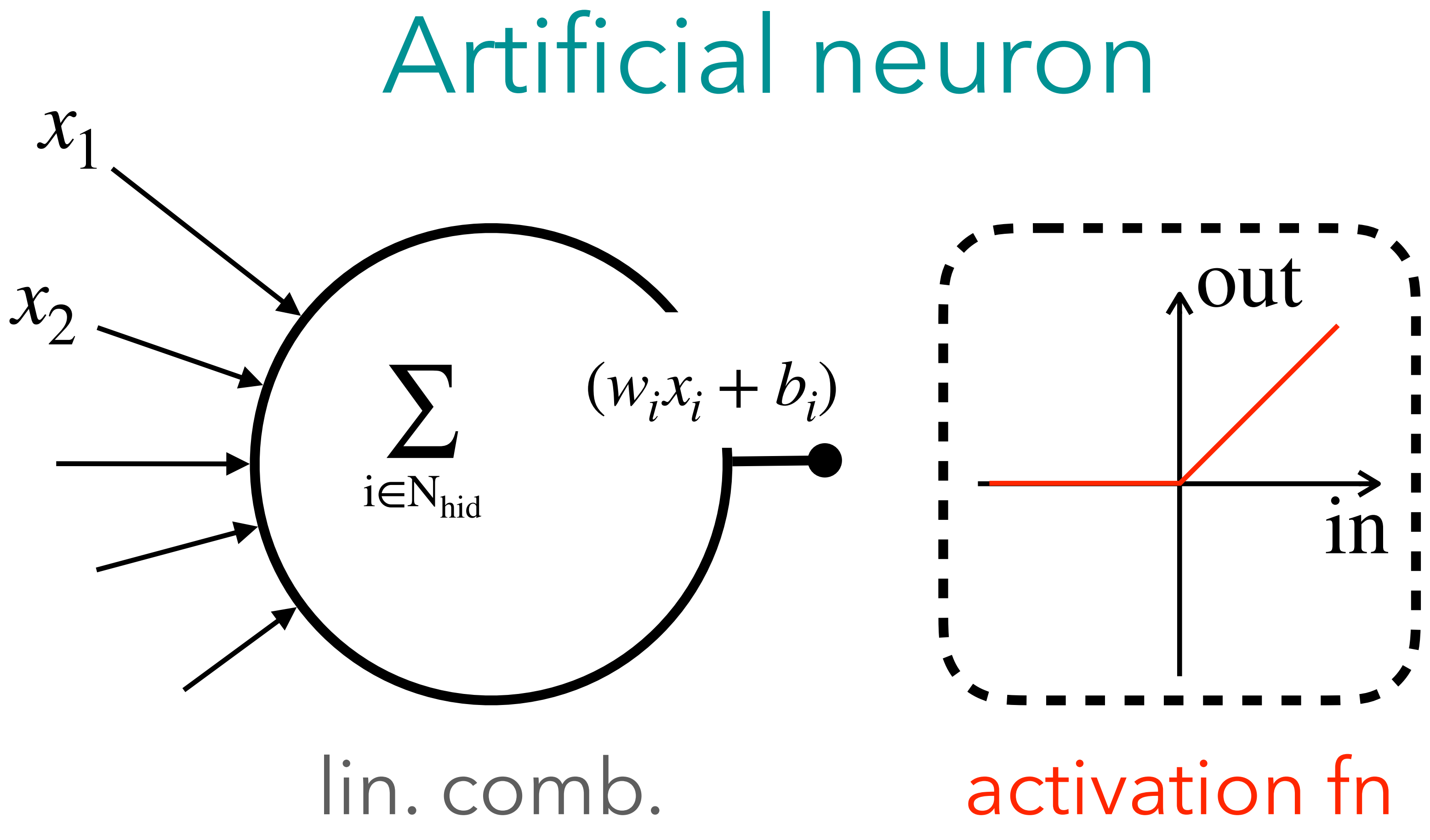


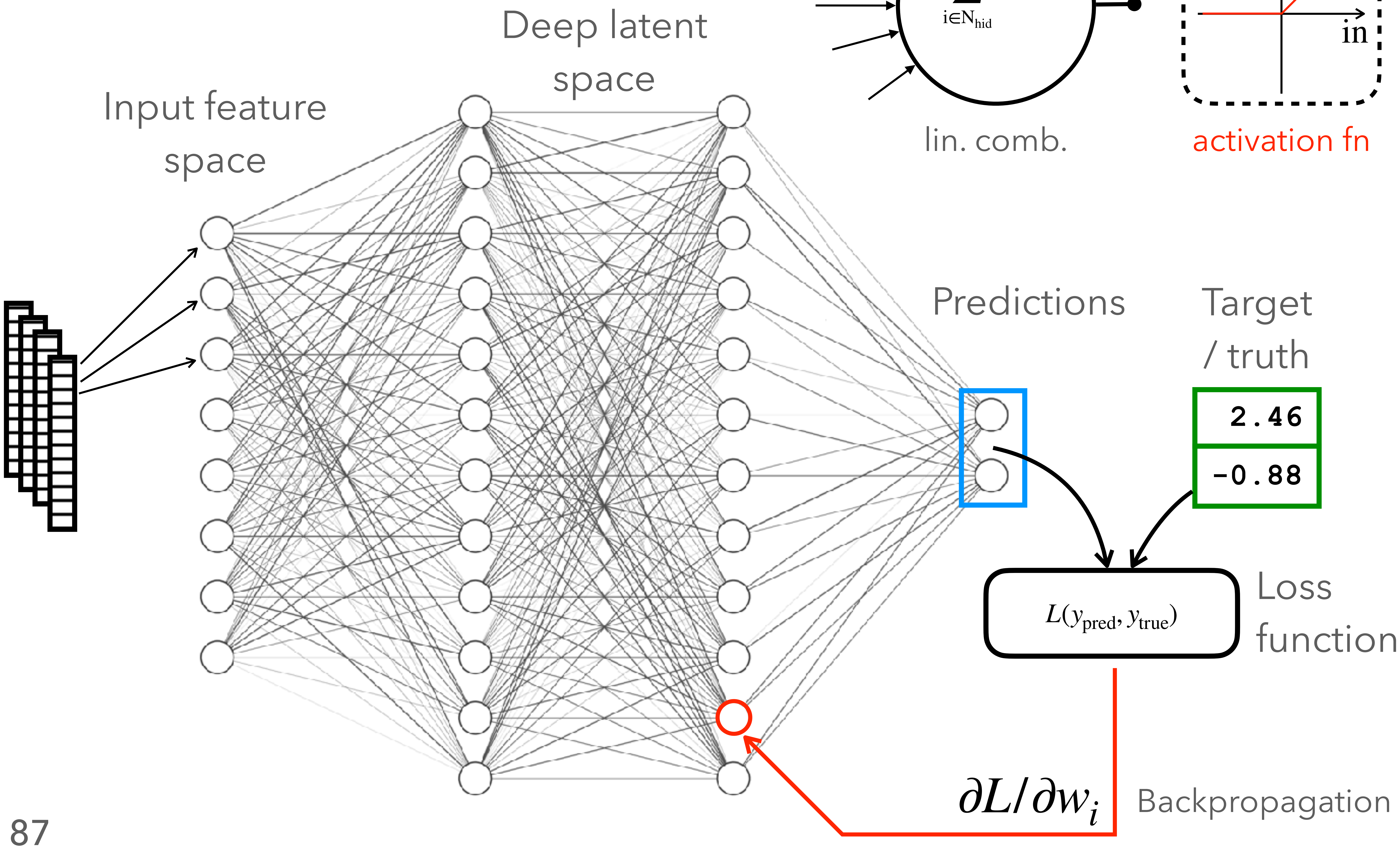
2010 – '20 rise of GPUs, computer vision, transformers, …

2020s LLMs, Nobel prizes in Physics and Chemistry

Neural networks

Multivariate classification, regression, …





Dataset

Single jets extracted from CMS Open Data in Phys. Rev. D 101, 034009

- Full CMS simulation and reconstruction
- Our goal is to mimic it

For each jet, extract sets of:

- Truth particles (input)
- Particle Flow Candidates (ground truth)

As reference, we run Delphes with:

- CMS Run-1 card
- Same truth particles as input
- appropriate pileup conditions

1M examples each

p_T^{min} - p_T^{max} [GeV]	Type	Training	Testing
470 - 600	Out-of-distribution		✓
600 - 800	Out-of-distribution		✓
800 - 1000	In-distribution	✓	✓
1000 - 1400	In-distribution	✓	✓
1400 - 1800	Out-of-distribution		✓
1800 - ∞	Out-of-distribution		✓