# Full event particle-level unfolding

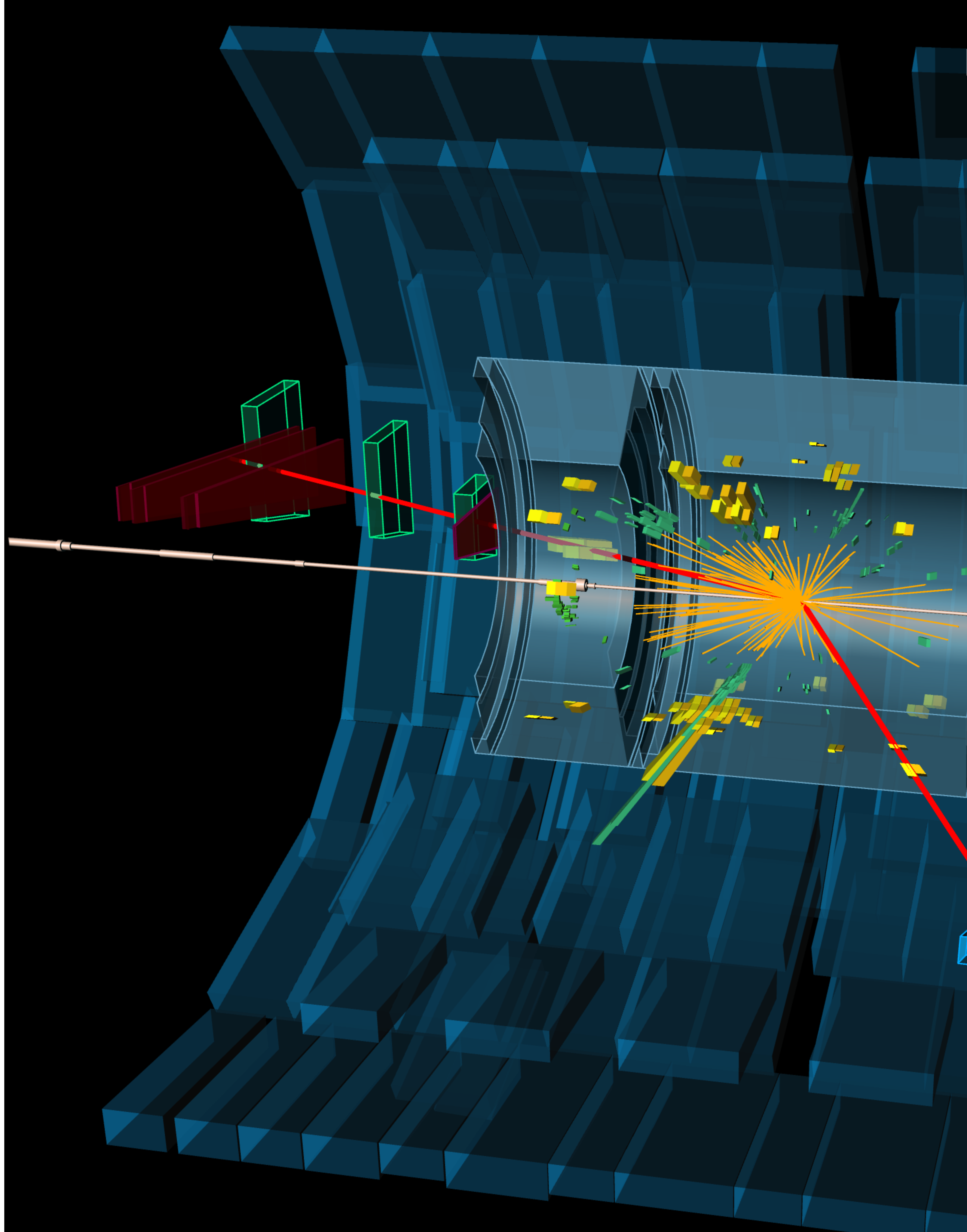
## with variable length variational latent diffusion (VL-VLD)

**Alexander Shmakov, Kevin Greif, Michael Fenton, Aishik Ghosh, Pierre Baldi, Daniel Whiteson**

November 11th, 2024
ML4Jets

# Why variable dimensions?

- Most unfolding at the LHC targets **particle-level**
  - Phase space is inherently variable dimensional
- No existing generative method for unfolding variable dimensions
  - Discriminative approaches (Omnifold) work well in this context
- Necessary for **full-event** unfolding at particle level

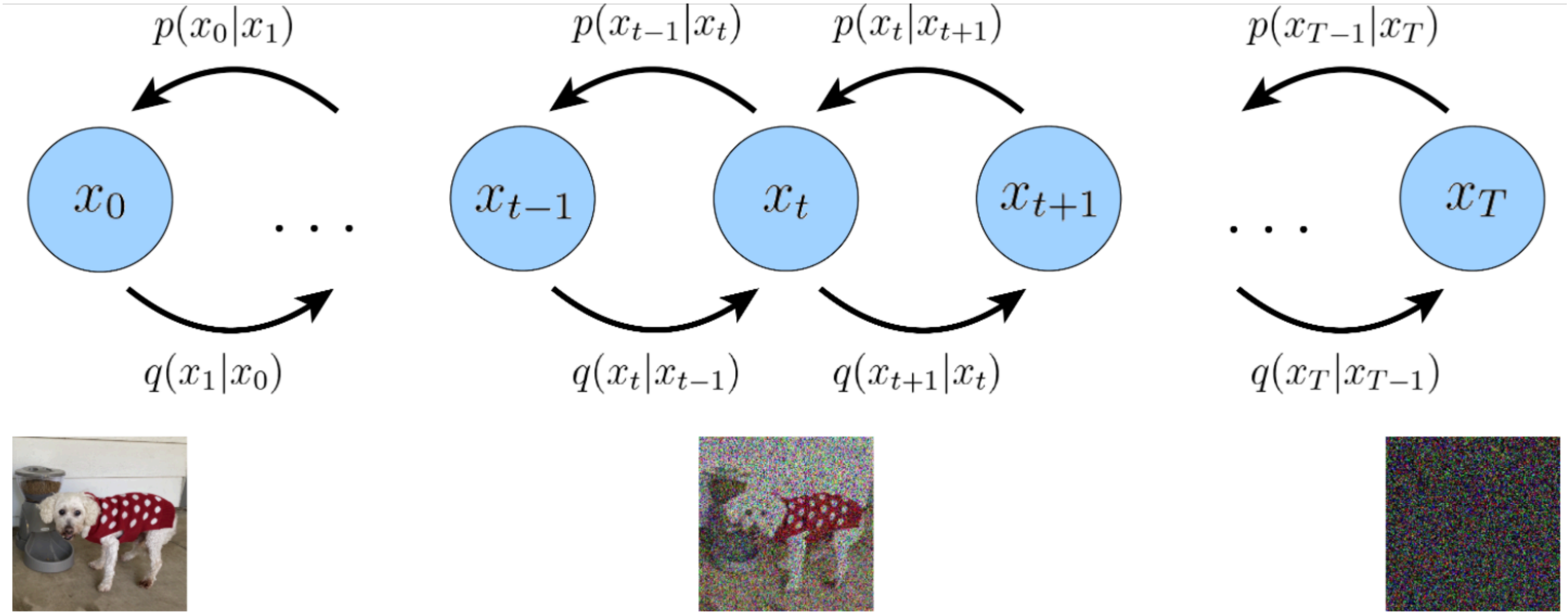


# Elements of latent variational diffusion

**Latent** diffusion model (2112.10752): perform the diffusion process in the latent space of a pre-trained variational autoencoder (VAE)

$$x \to z \sim VAE(x)$$

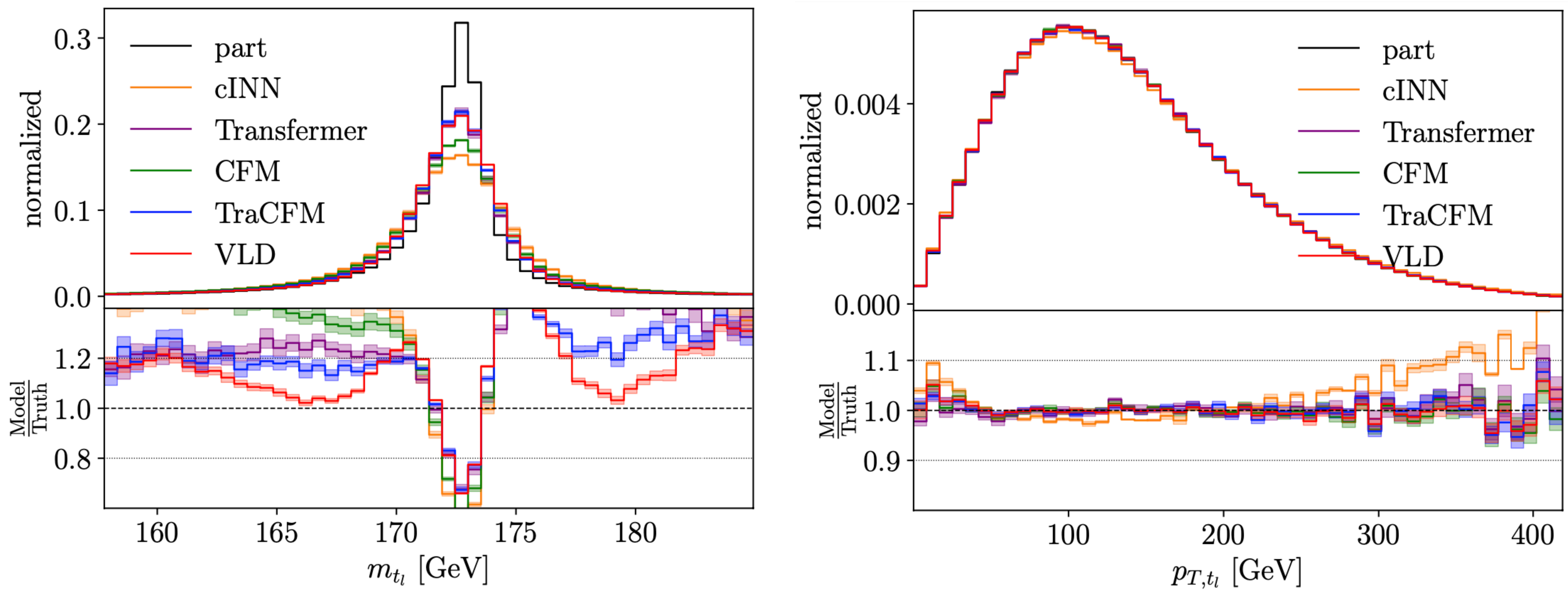
**Variational** diffusion model (2107.00630): interpretation of the diffusion model as an (infinitely deep) chain of VAEs




2208.11970

# On a parton level (fixed dimension) problem

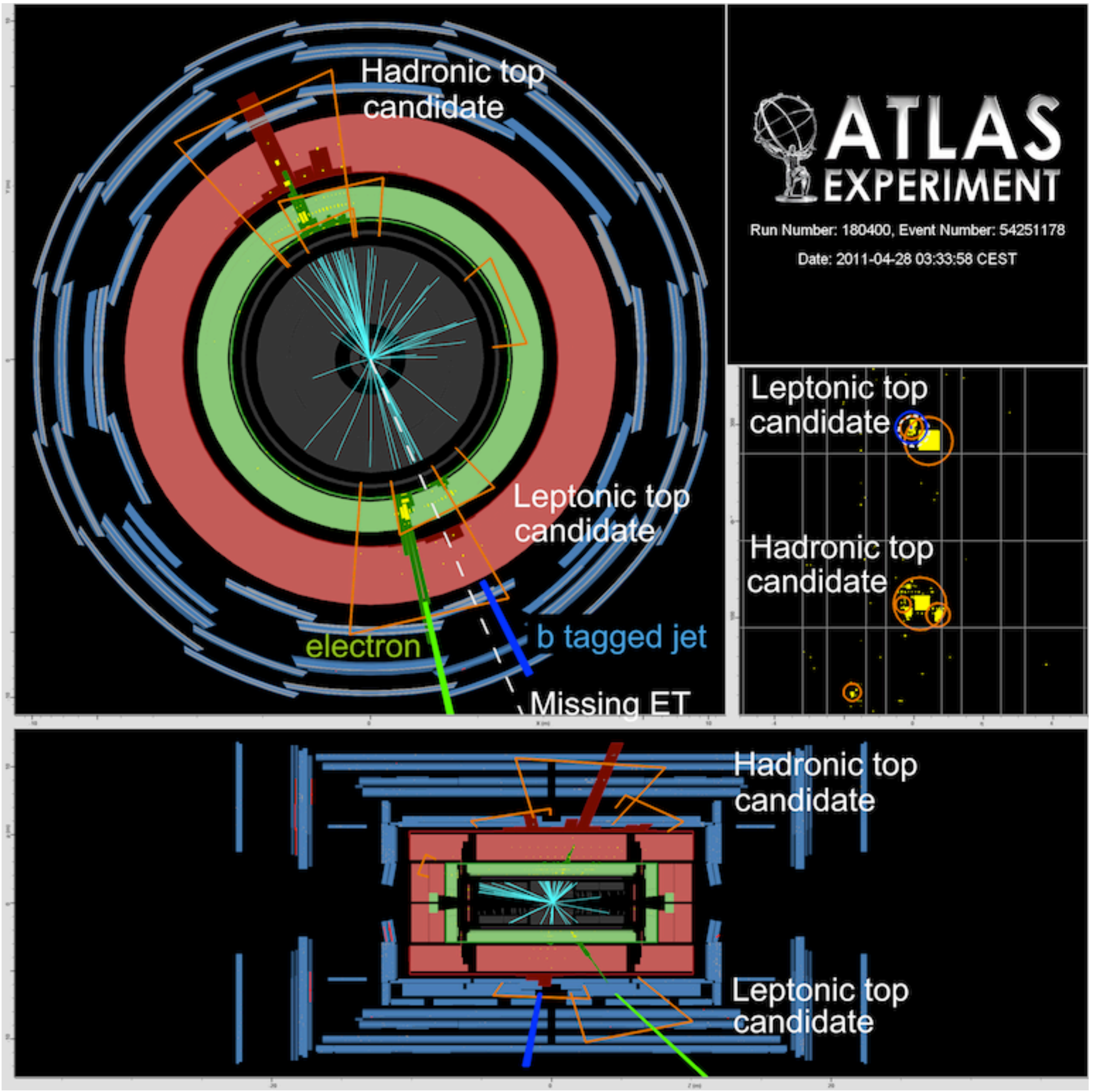
- Base model tested on parton-level $t\bar{t}$ unfolding: fixed dimensions
- Results included in comparison paper 2404.18807




**Results are shown without mass parametrization!**

# From partons to particles

## Particle-level unfolding: invert only the detector response




- Targets are particle-level objects:
  - Can be light quark jets, b tagged jets, electrons, or muons
  - Also interested in $E_T^{miss}$, $\phi^{miss}$, $\eta^{\nu}$
- **Do not always have 5 objects!**

# Variable length generative models

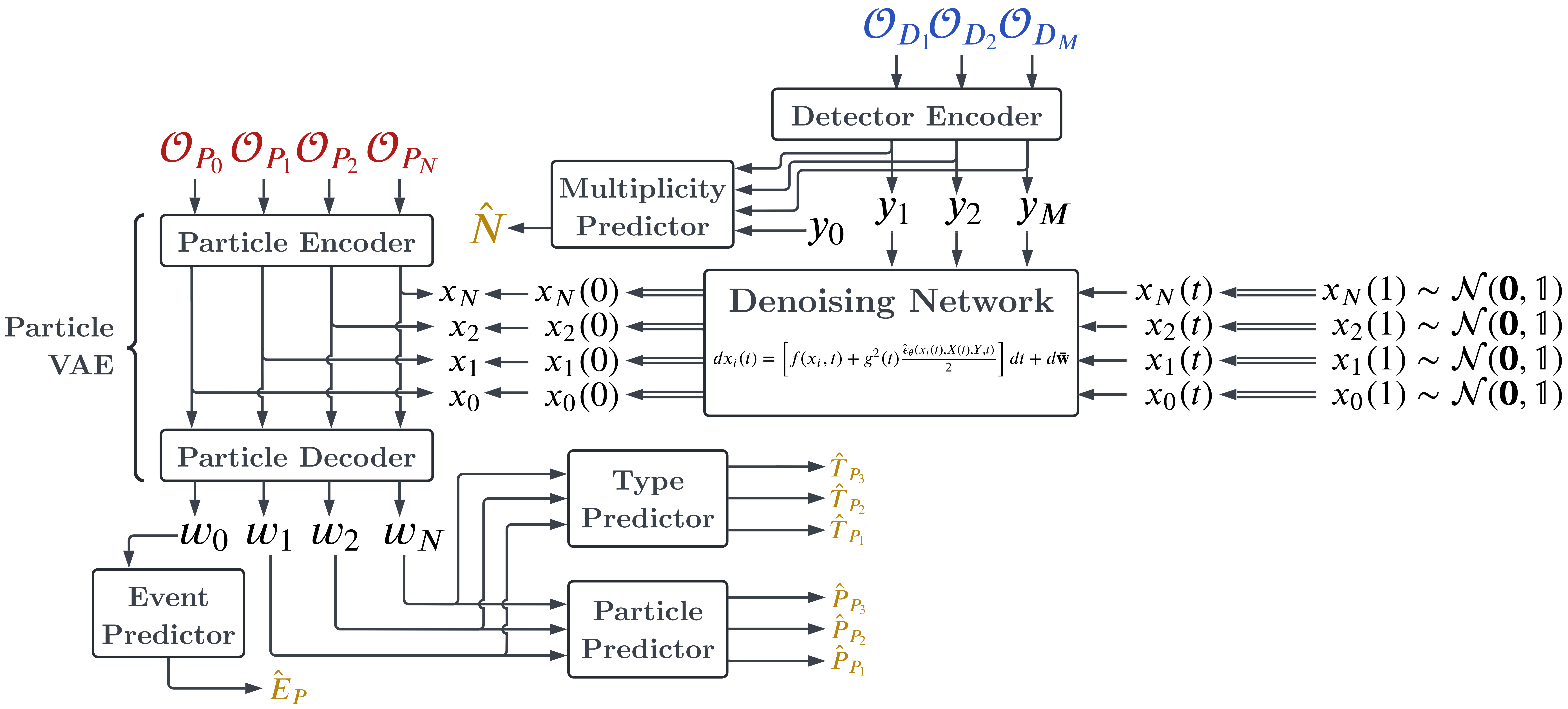
**Autoregressive approach:**

- Treat event as a sequence of objects, repeatedly run inference on model to generate sample object by object
- Output stop token to finish generation
- Approach used by ChatGPT, see 2403.05618 for a HEP example

**Multiplicity predictor approach:**

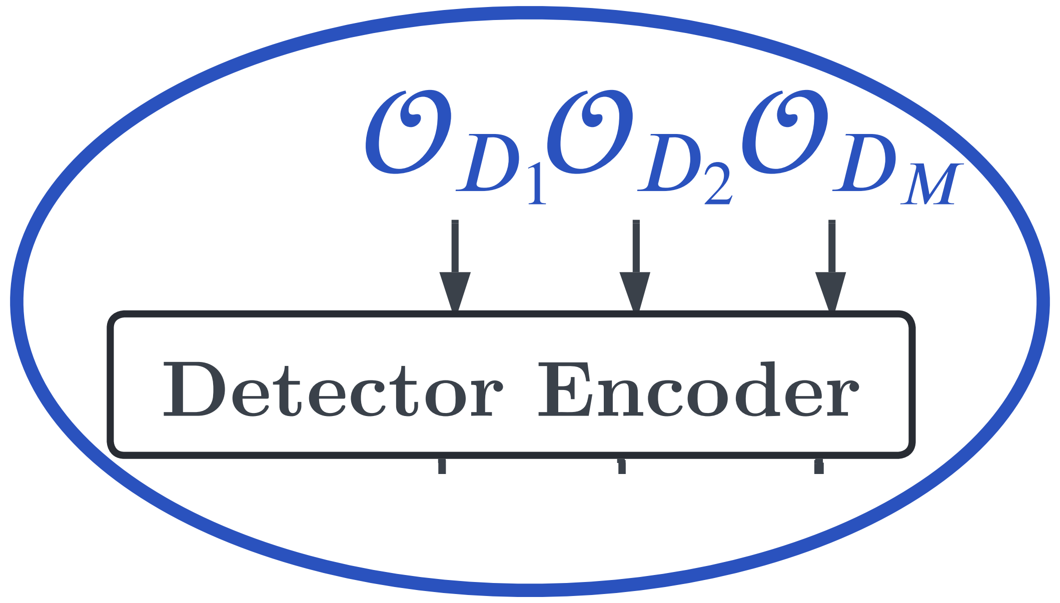
- Standard in HEP applications of point-cloud generative models (see backup)
- Use auxiliary network to predict particle multiplicity
- Generation is conditioned on the output of this model
- We take $\hat{N} = [N]; N \sim \Gamma(MLP_k(y), MLP_\theta(y))$

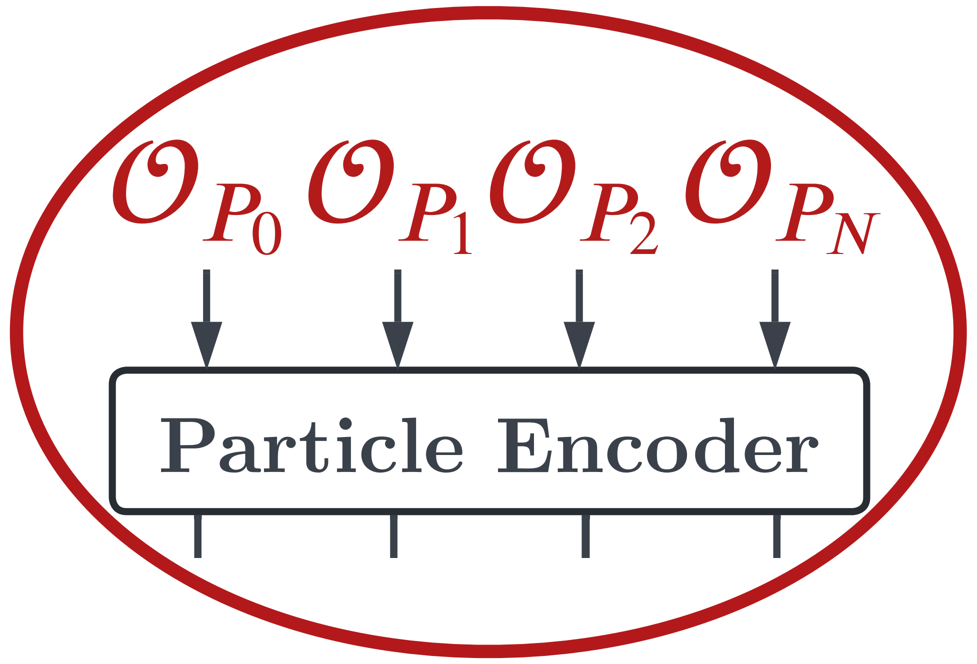
# Training variable length VLD (VL-VLD)



# Training variable length VLD (VL-VLD)

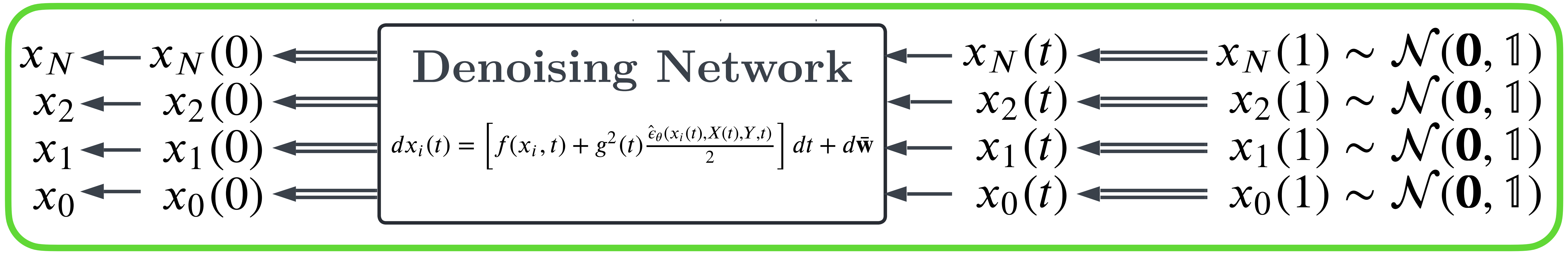
1. Encode particle-level and detector-level events into learned representations



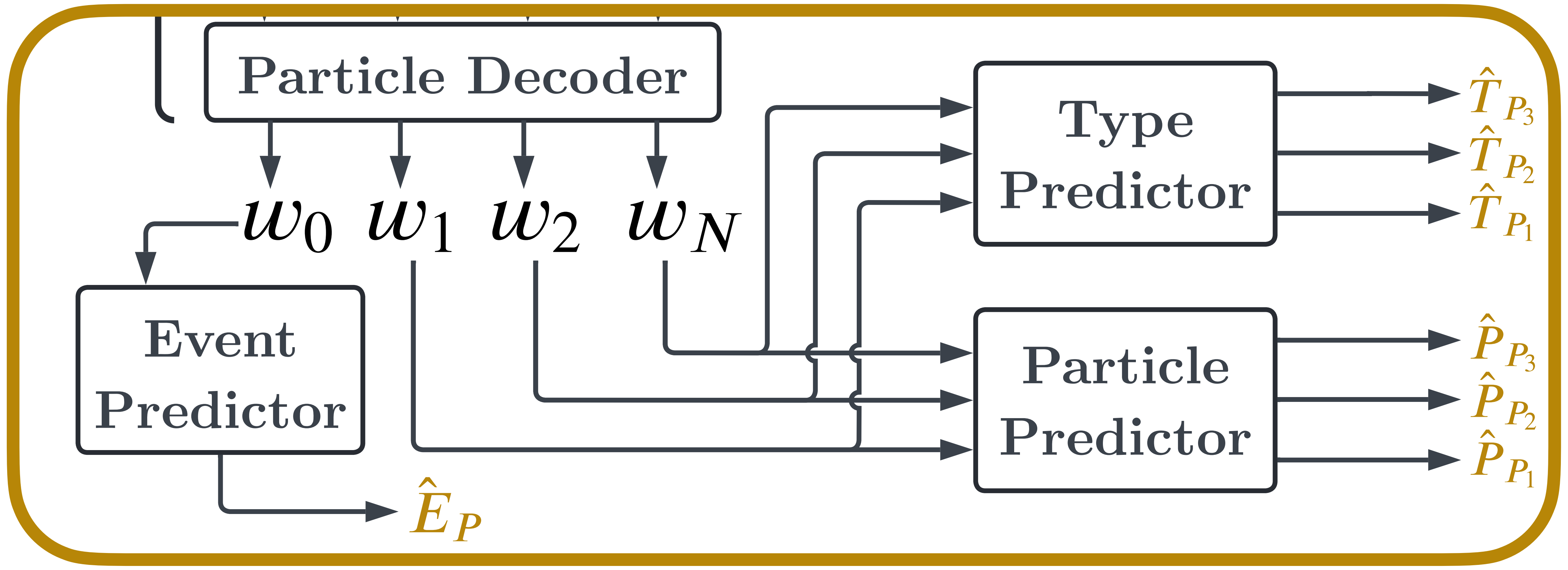



# Training variable length VLD (VL-VLD)

2. Train denoising network to remove noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}); \epsilon \in \mathbb{R}^{\mathbb{N}}$ from the encoded particle level event



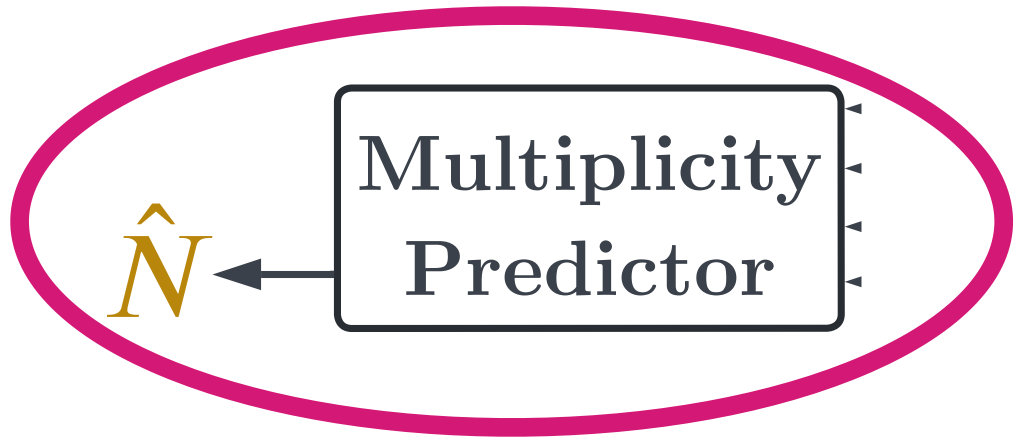
# Training variable length VLD (VL-VLD)




3. Train particle decoder and predictor MLPs to reconstruct the particle level event

# Training variable length VLD (VL-VLD)

4. Train multiplicity predictor to predict shape and width parameters of $\Gamma$ distribution



# VL-VLD loss function

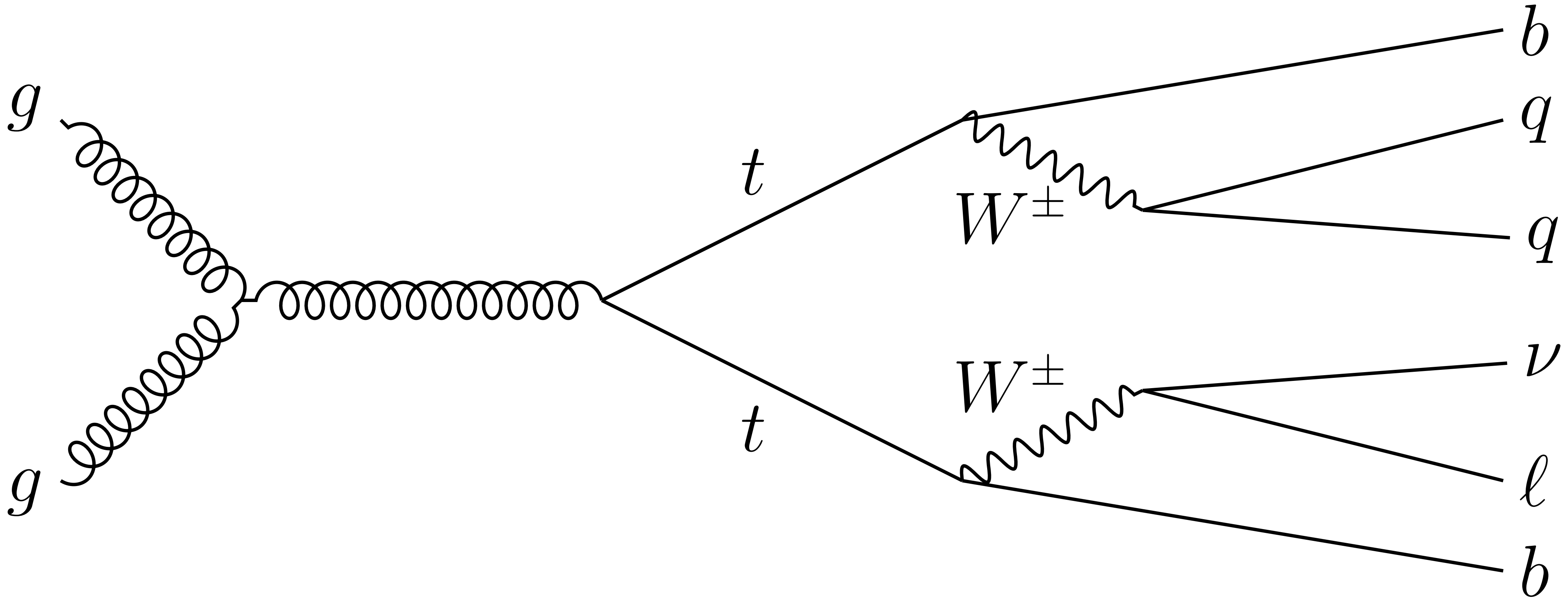
All networks are trained simultaneously to minimize a unified loss function:

$$
\begin{aligned}
\mathcal{L} = & \sum_{i \in \{0,1,\ldots N\}} D_{KL}[q(x_i(1)|\mathcal{O}_P, \mathcal{O}_D) \parallel p(x_i(1))] && \textsc{Prior Loss} \\
& + \sum_{i \in \{0,1,\ldots N\}} \mathbb{E}_{q(x_i(0)|\mathcal{O}_P)} \left[ -\log p(\hat{\mathcal{O}}_P | x_i(0)) \right] && \textsc{Reconstruction Loss} \\
& + \sum_{i \in \{0,1,\ldots N\}} \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}), t \sim \mathcal{U}(0,1)} \left[ \gamma'_\phi(t) \|\epsilon - \hat{\epsilon}_\theta(x_i(t), X(t), Y, t)\|_2^2 \right] && \textsc{Denoising Loss} \\
& - \log p(\hat{N} = N | \mathcal{O}_D). && \textsc{Multiplicity Loss} \quad (12)
\end{aligned}
$$

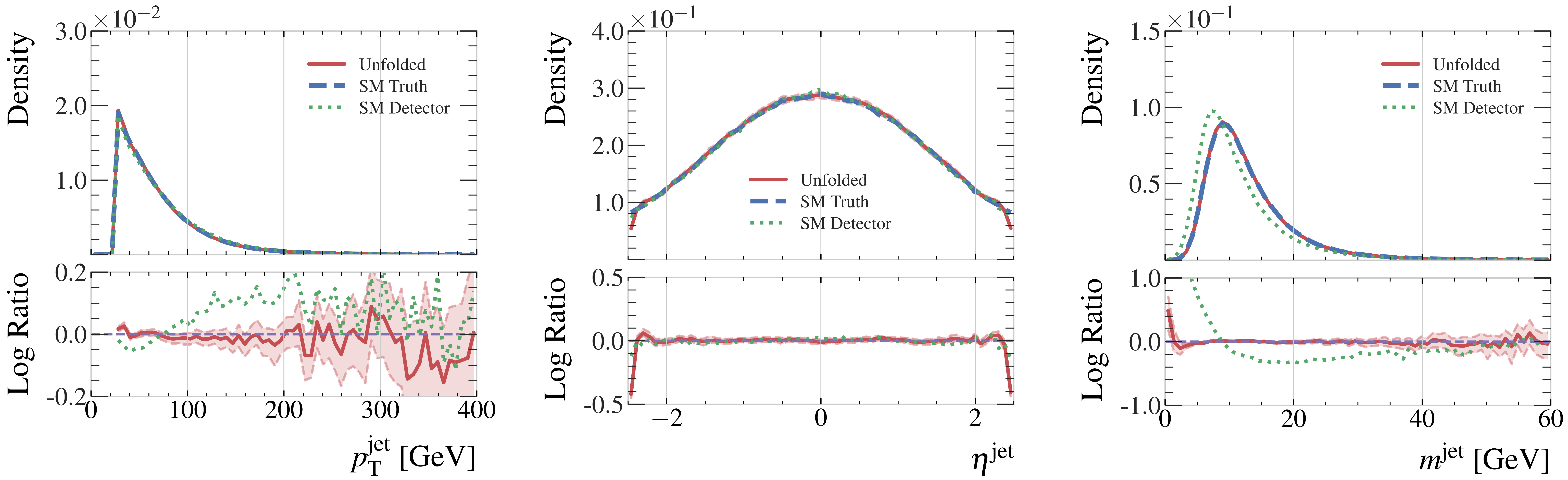
Transformer architectures ensure that network predictions are **position equivariant**

# Particle-level $t\bar{t}$ unfolding dataset

- Semi-leptonic decay mode: expect 2 light quark jets, 2 b jets, 1 lepton, MET
- Detector response simulated with Delphes
- **Identical detector and particle-level phase space requirements**:
  - Leptons and jets required to have $p_T > 25$ GeV, $|\eta| < 2.5$
  - Require 1 lepton and at least 4 jets (at least 2 b-tagged)
- Targets are object kinematics vectors: $P_i = (p_x, p_y, p_z, \log(E+1), \log(M+1))$
  - Also object type, encoded as one-hot vector
- Event-level targets: $E_T^{miss}, \phi^{miss}, p_x^\nu, p_y^\nu, p_z^\nu, E^\nu$

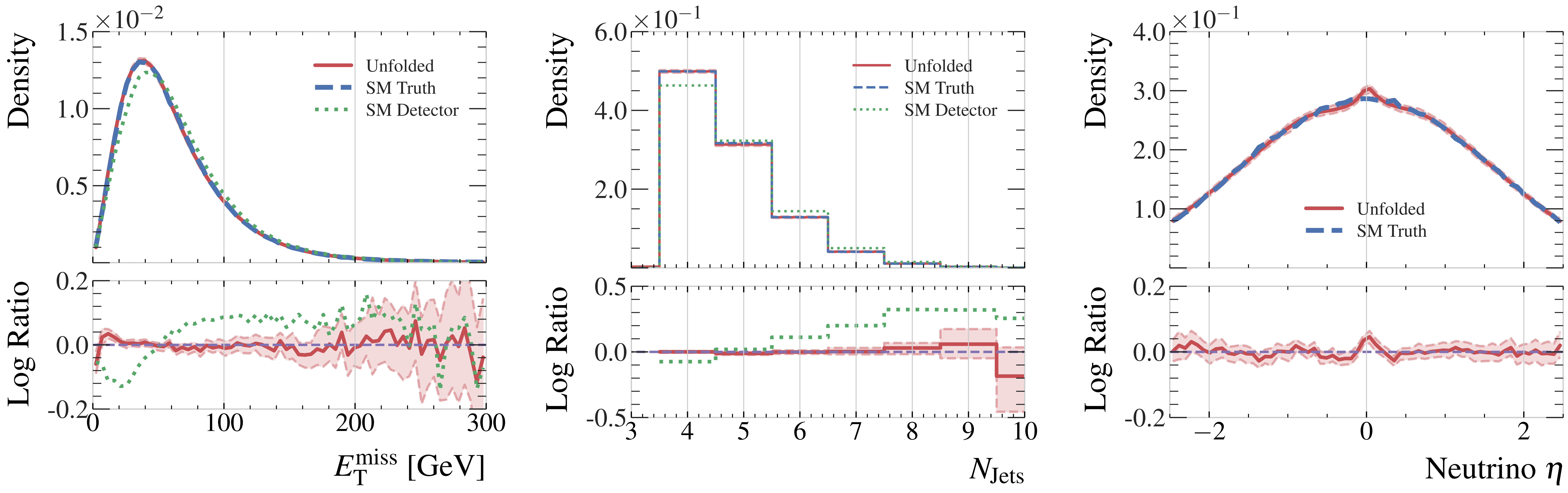


# Inclusive kinematic distributions for jets




- Kinematics of the particle level objects close well: these are directly optimized
- Struggle in edges of phase space where we lack training examples of events migrating across phase space boundaries

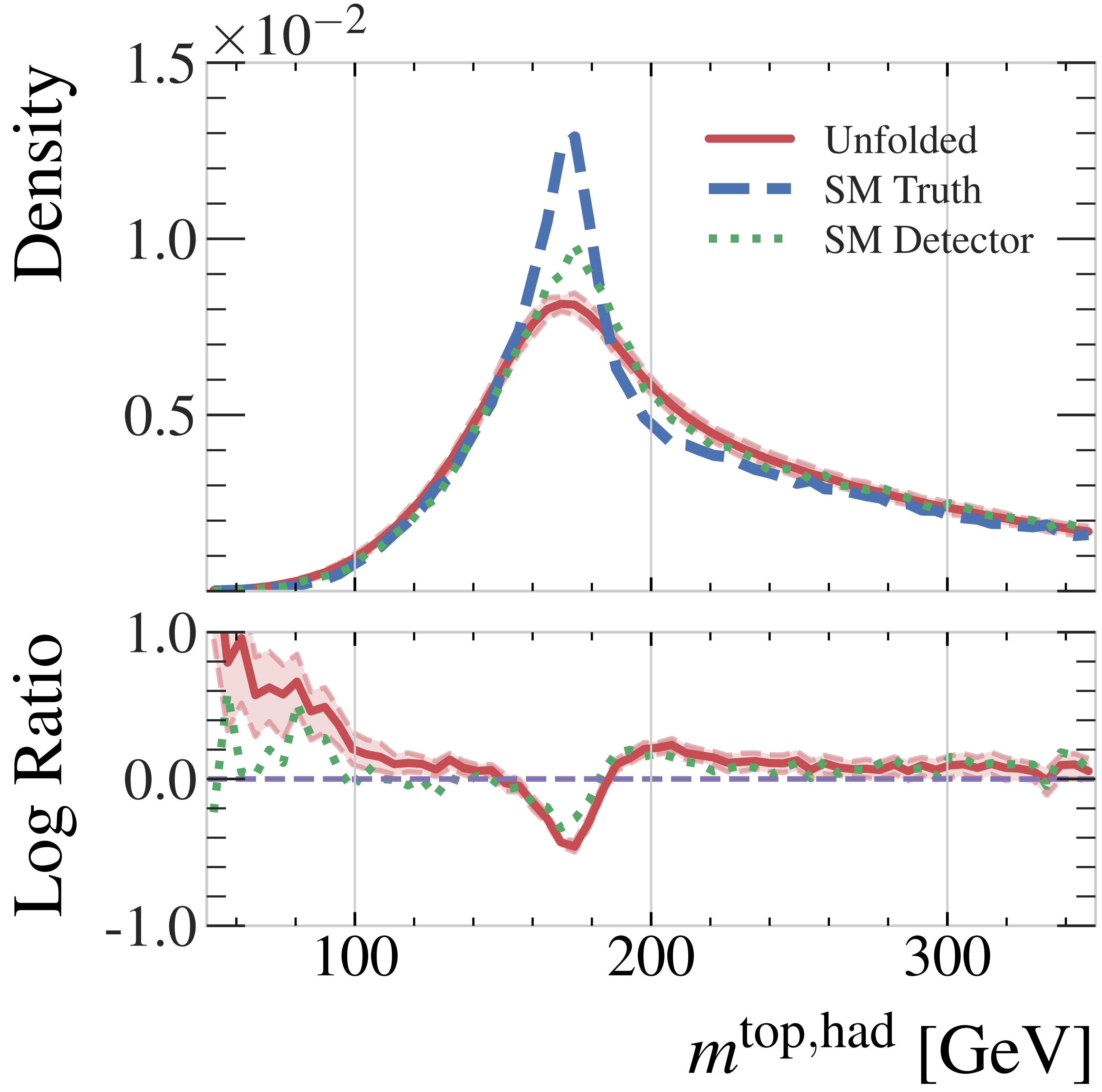
# Event-level distributions




- Event-level features also close well
- Neutrino $\eta$ is not constrained at detector-level, expect excess at 0 to result from model returning mean

# Particle-level top quark distributions

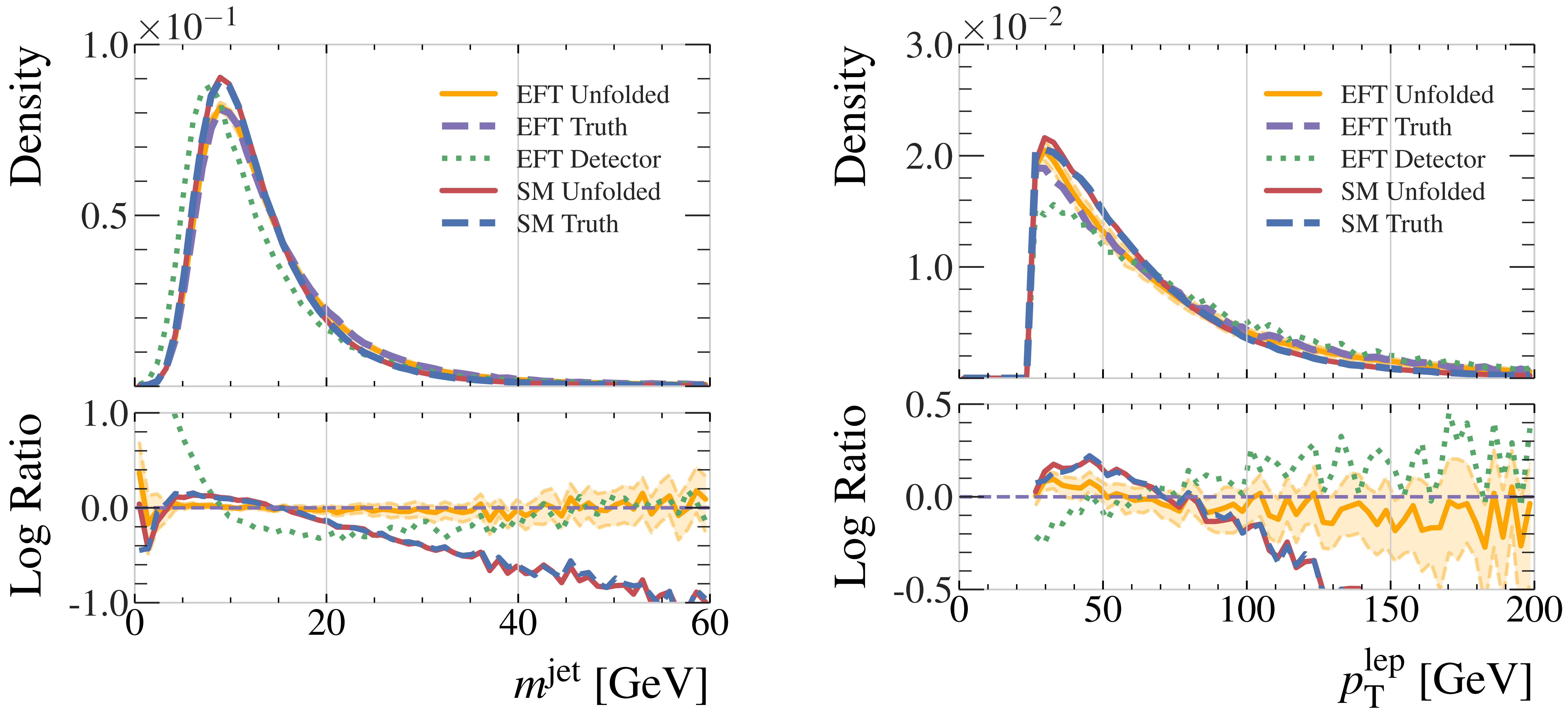
## Assumes pseudotop jet/parton assignment (see backup)




- Top kinematics: not directly optimized
- Sharply peaked distributions difficult to model without direct optimization
- Why not optimize?
  - Requires assumption of a jet/parton assignment in training
  - Assignment must be differentiable to optimize top kinematics calculated from particle-level objects

# EFT operator prior shift

- Generative models can suffer from prior-dependence
- Test by evaluating model over dataset generated with non-zero EFT operator



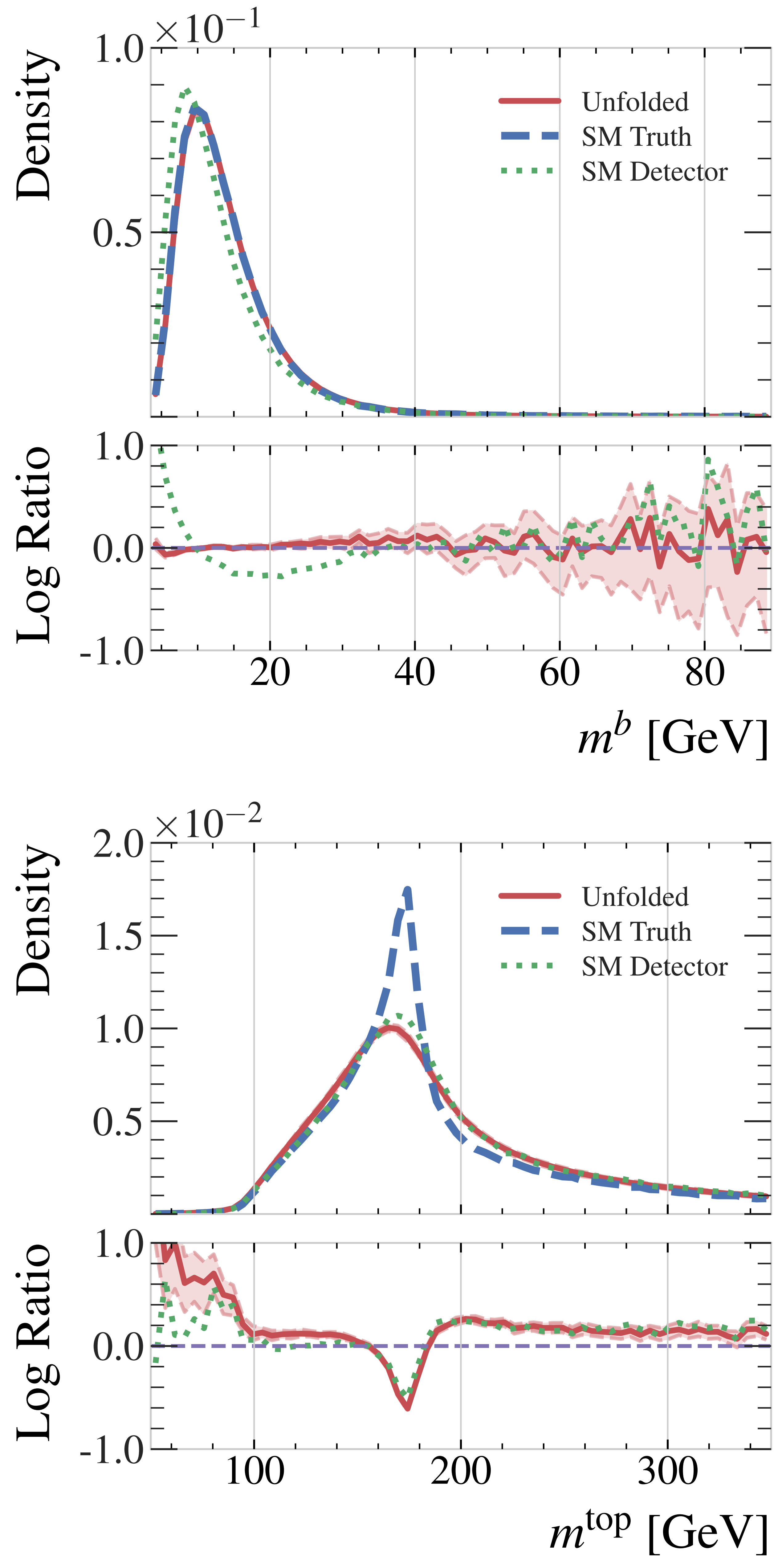

Clearly not just reproducing the SM distributions!

However iteration likely necessary in practice

# Conclusions

- First attempt at full event particle-level unfolding with a generative model
  - Method also applies to unfolding **all particles**, but order of magnitude higher dimensional problem
- Directly optimized quantities close well
- Derived quantities, like reconstructed top quark kinematics, are difficult
  - Can we improve? Data is public!

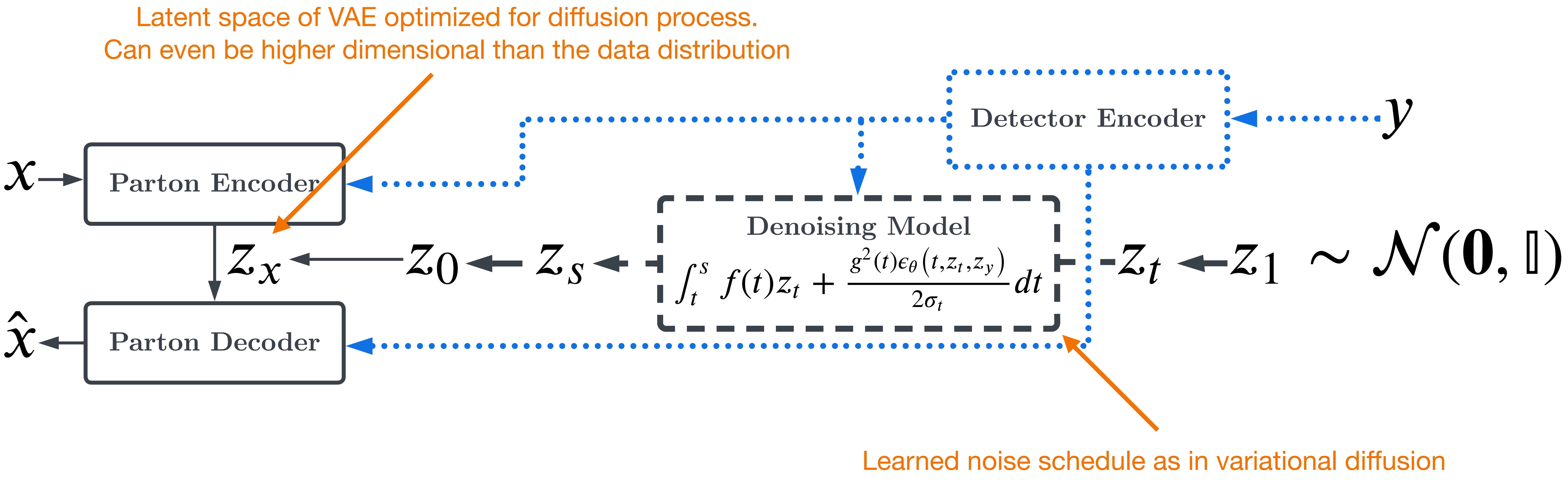
**Data:** https://zenodo.org/records/13364827
**Code:** https://github.com/Alexanders101/LVD
**Paper:** https://arxiv.org/abs/2404.14332



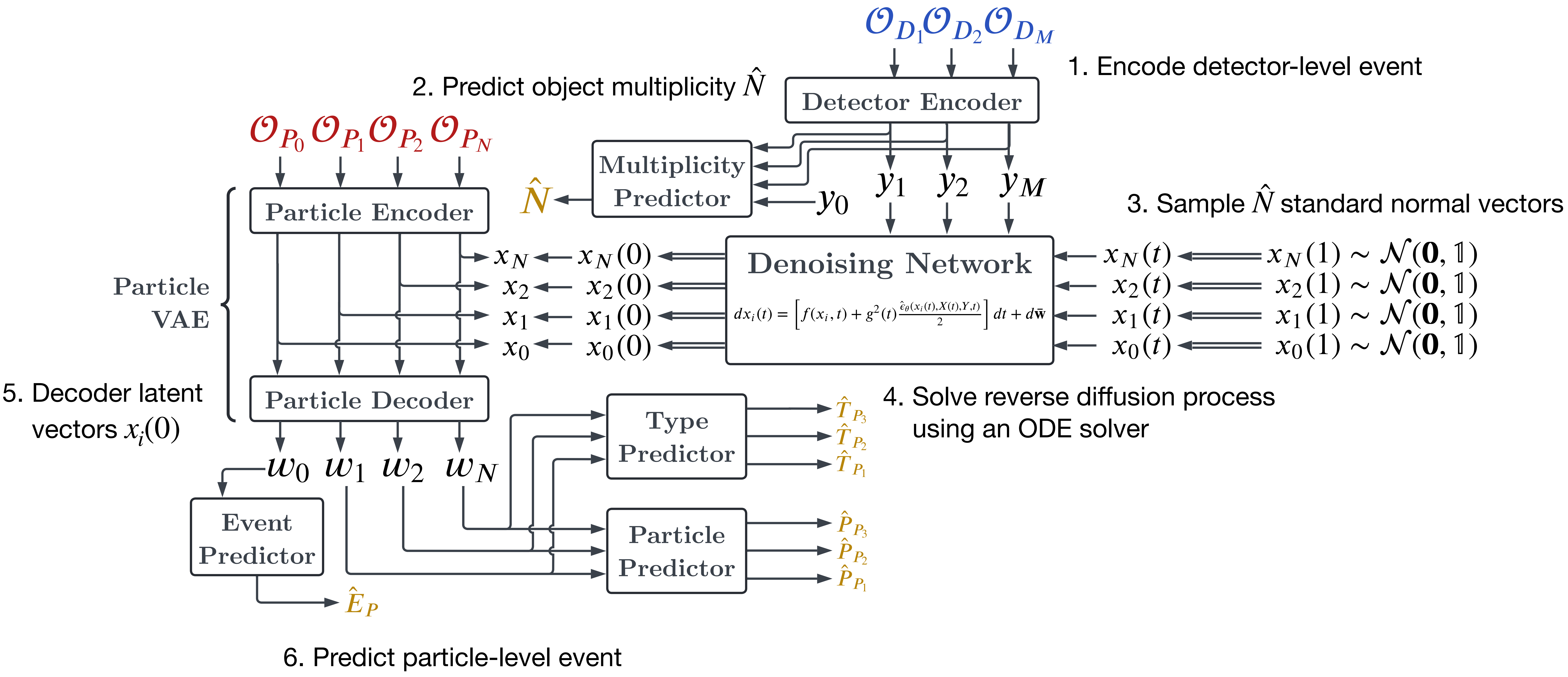
# Backup

# Variational Latent Diffusion (VLD)

Combine these ideas in an end-to-end model:



# Inference with VL-VLD



# VL-VLD loss function

All networks are trained at once to minimize a unified loss function:

$$
\begin{aligned}
\mathcal{L} = & \sum_{i\in\{0,1,\ldots N\}} D_{KL}[q(x_i(1)|\mathcal{O}_P,\mathcal{O}_D) \,\|\, p(x_i(1))] && \textsc{Prior Loss} \\
& + \sum_{i\in\{0,1,\ldots N\}} \mathbb{E}_{q(x_i(0)|\mathcal{O}_P)}\left[-\log p(\hat{\mathcal{O}}_P|x_i(0)\right] && \textsc{Reconstruction Loss} \\
& + \sum_{i\in\{0,1,\ldots N\}} \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbb{I}),t\sim\mathcal{U}(0,1)}\left[\gamma'_\phi(t)\,\|\epsilon-\hat{\epsilon}_\theta(x_i(t),X(t),Y,t)\|_2^2\right] && \textsc{Denoising Loss} \\
& - \log p(\hat{N}=N|\mathcal{O}_D). && \textsc{Multiplicity Loss} \quad (12)
\end{aligned}
$$

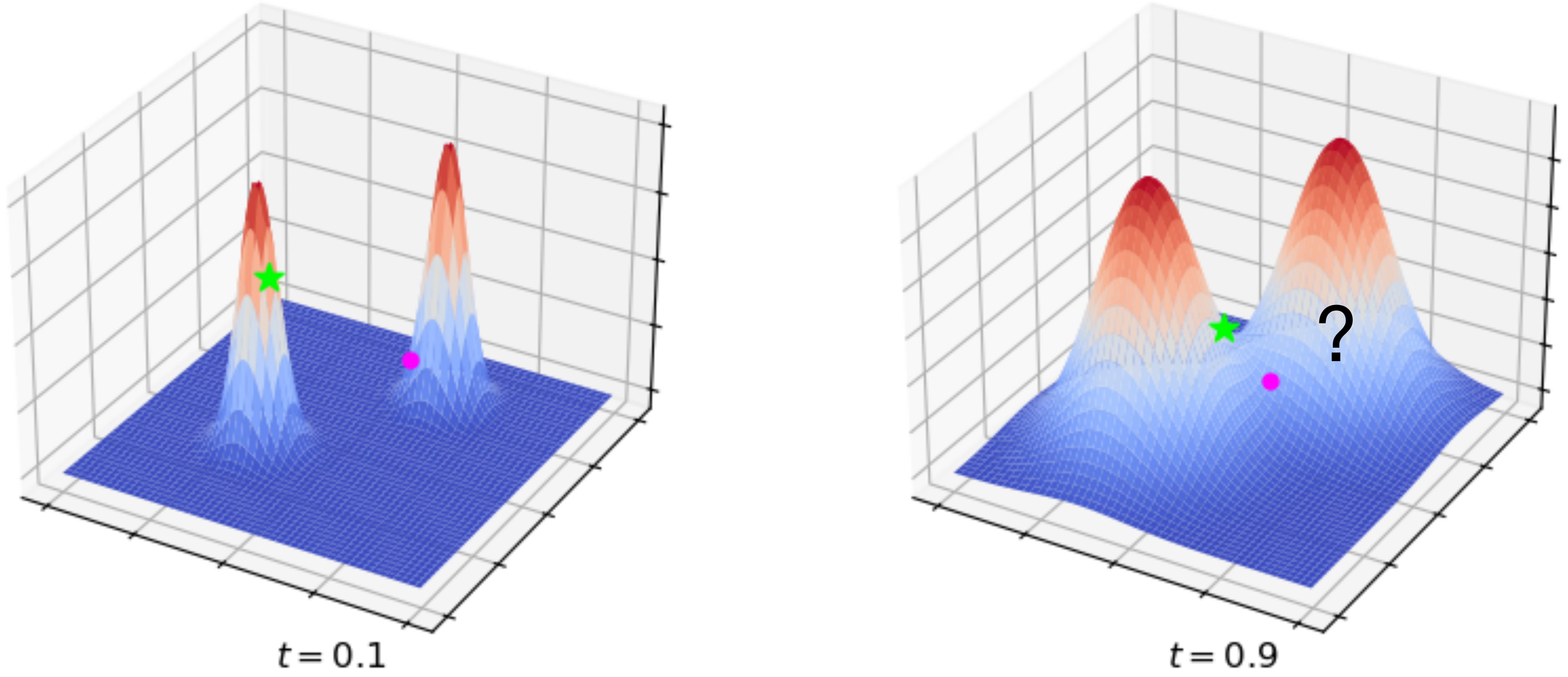
Transformer architectures ensure that network predictions are **position equivariant**

**Except there's a problem with the denoising loss**

# Ambiguous loss function

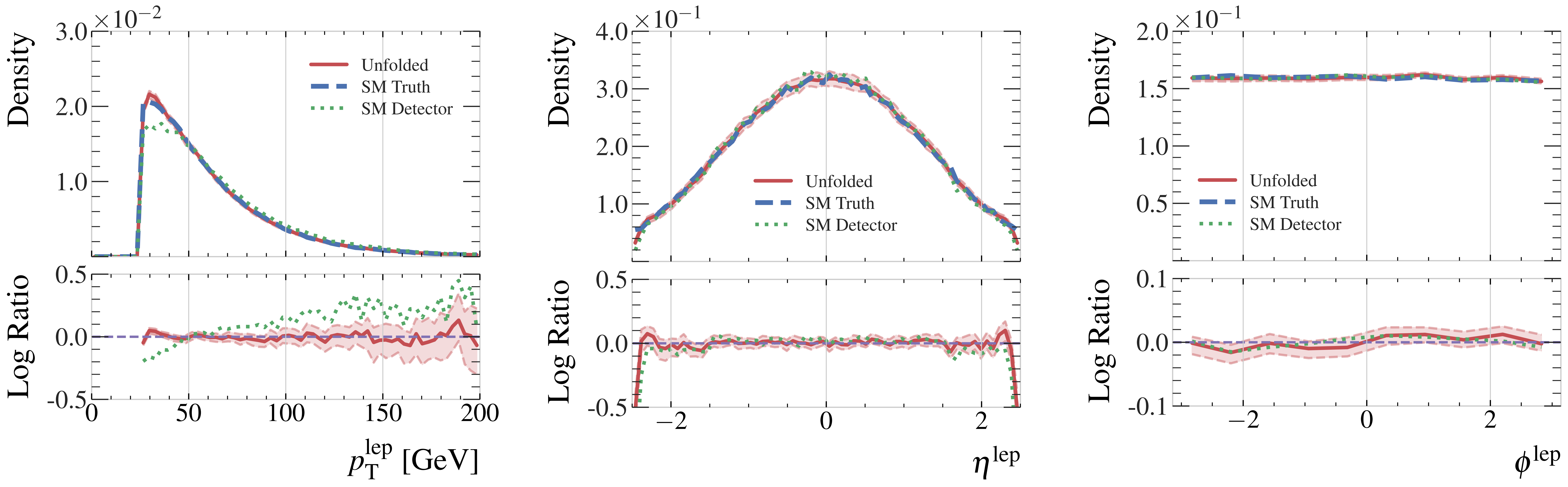
$$\sum_{i\in\{0,1,\ldots N\}} \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbb{I}),t\sim\mathcal{U}(0,1)}\left[\gamma'_{\phi}(t)\,\|\epsilon-\hat{\epsilon}_{\theta}(x_i(t),X(t),Y,t)\|_2^2\right]$$

**At high level of noise, the distinction between two objects can become ambiguous, making object-wise MSE loss undefined:**




**Solution: Impose ordering of objects by true particle-level $p_T$ when training denoising network**

# Inclusive kinematic distributions (leptons)

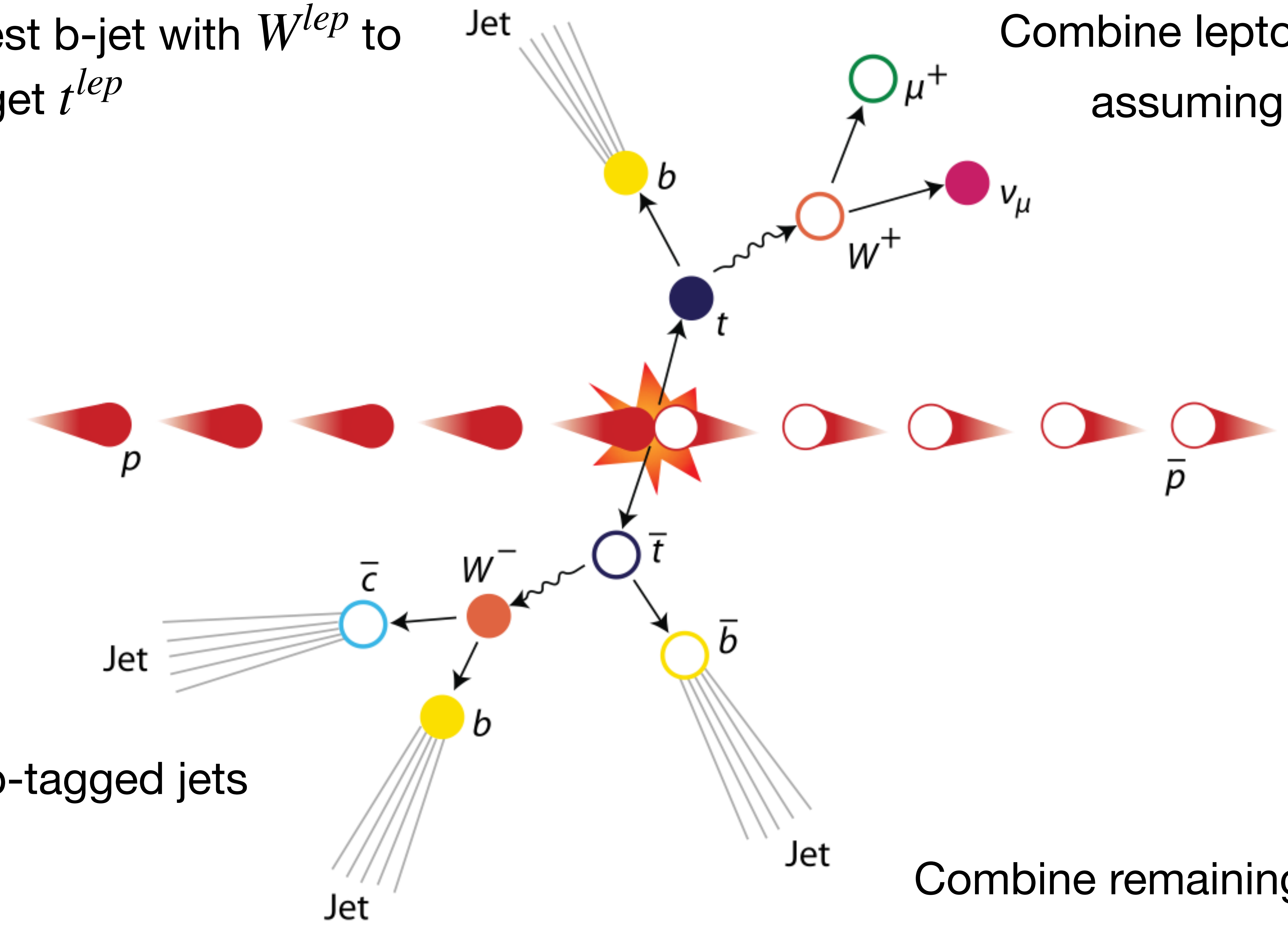



- Error bands estimated by sampling each particle-level configuration 128 times
- Kinematics of the directly optimized objects close well.
- Can struggle in edges of phase space where we lack training examples of events migrating across phase space boundaries from detector to particle-level

# Pseudotop algorithm

Combine closest b-jet with $W^{lep}$ to get $t^{lep}$

Combine lepton with $E_T^{miss}$ and solve for $p_z^{\nu}$ assuming the $W$ mass to get $W^{lep}$

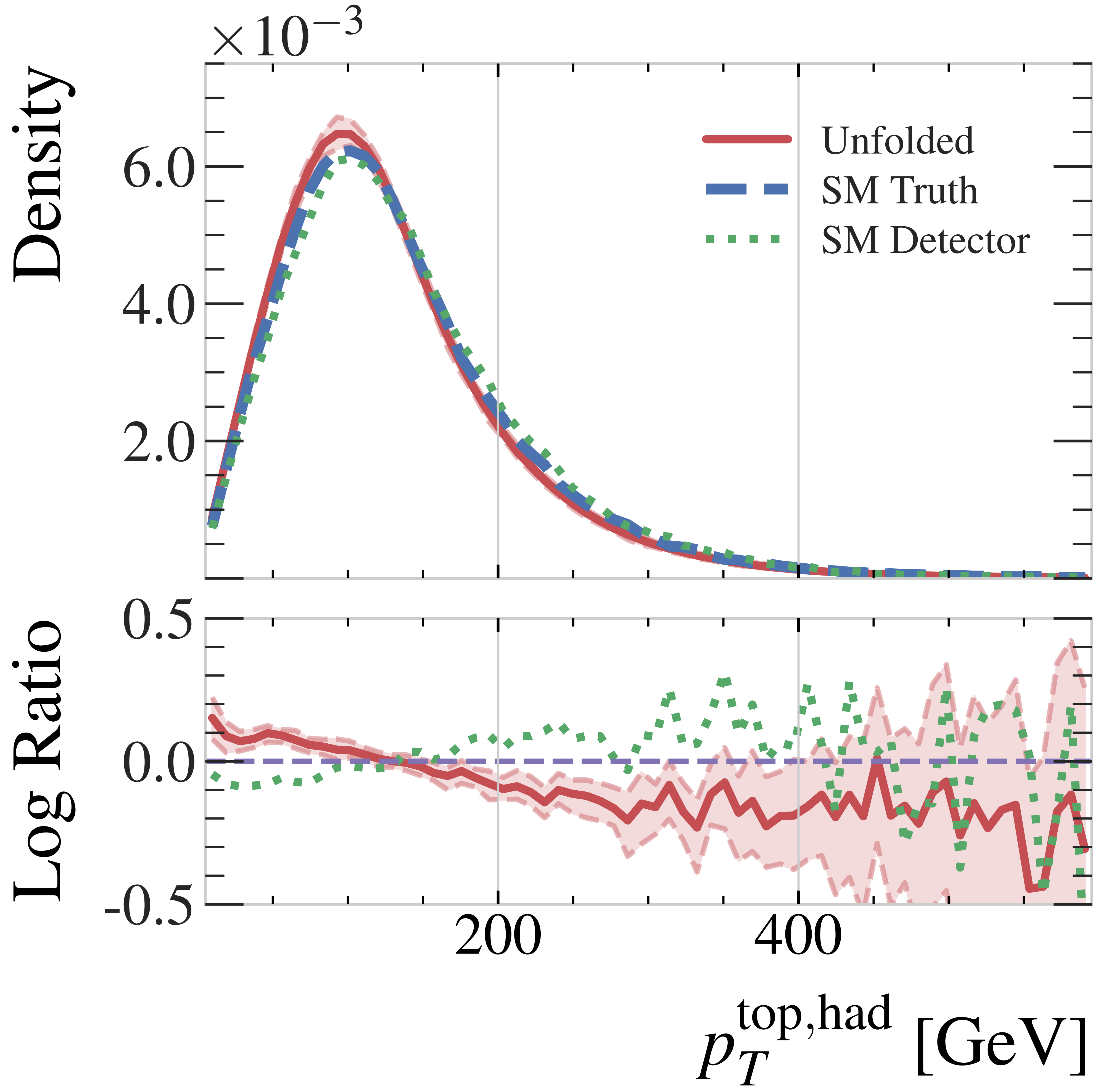


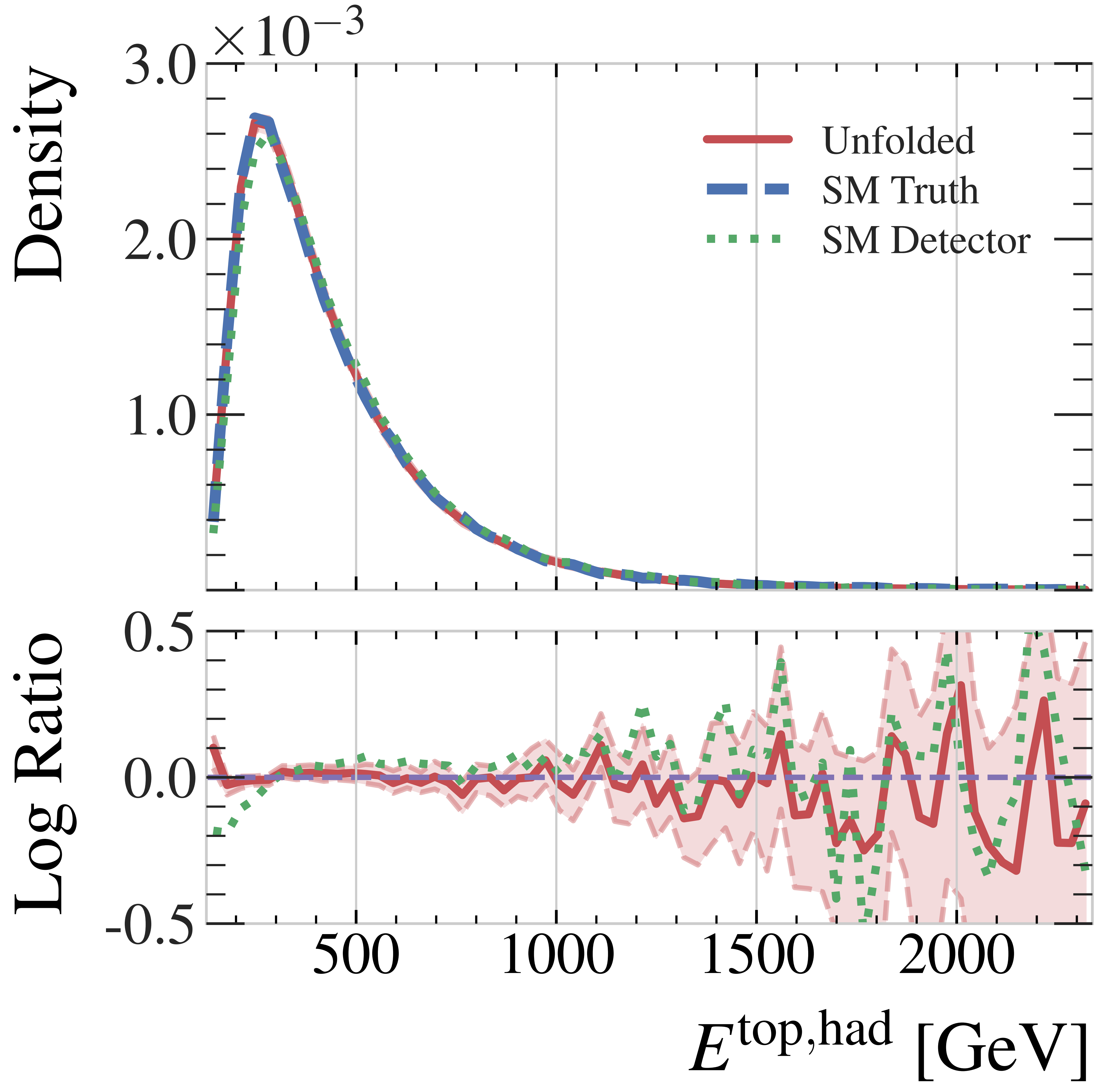

Combine two highest $p_T$ non b-tagged jets to get $W^{had}$

Combine remaining b-jet to get $t^{had}$

# Hadronic top kinematics

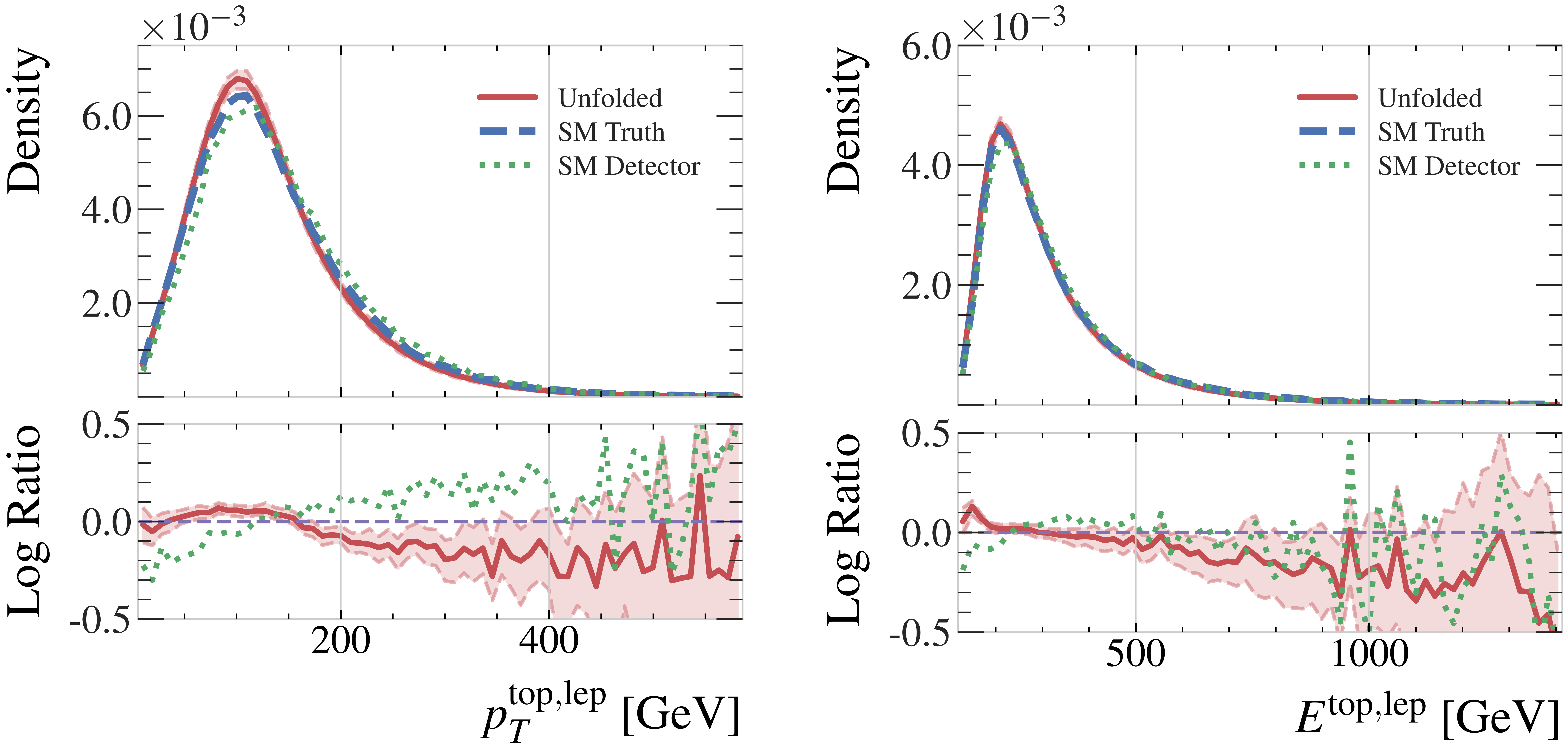
## Assumes pseudo-top jet/parton assignment



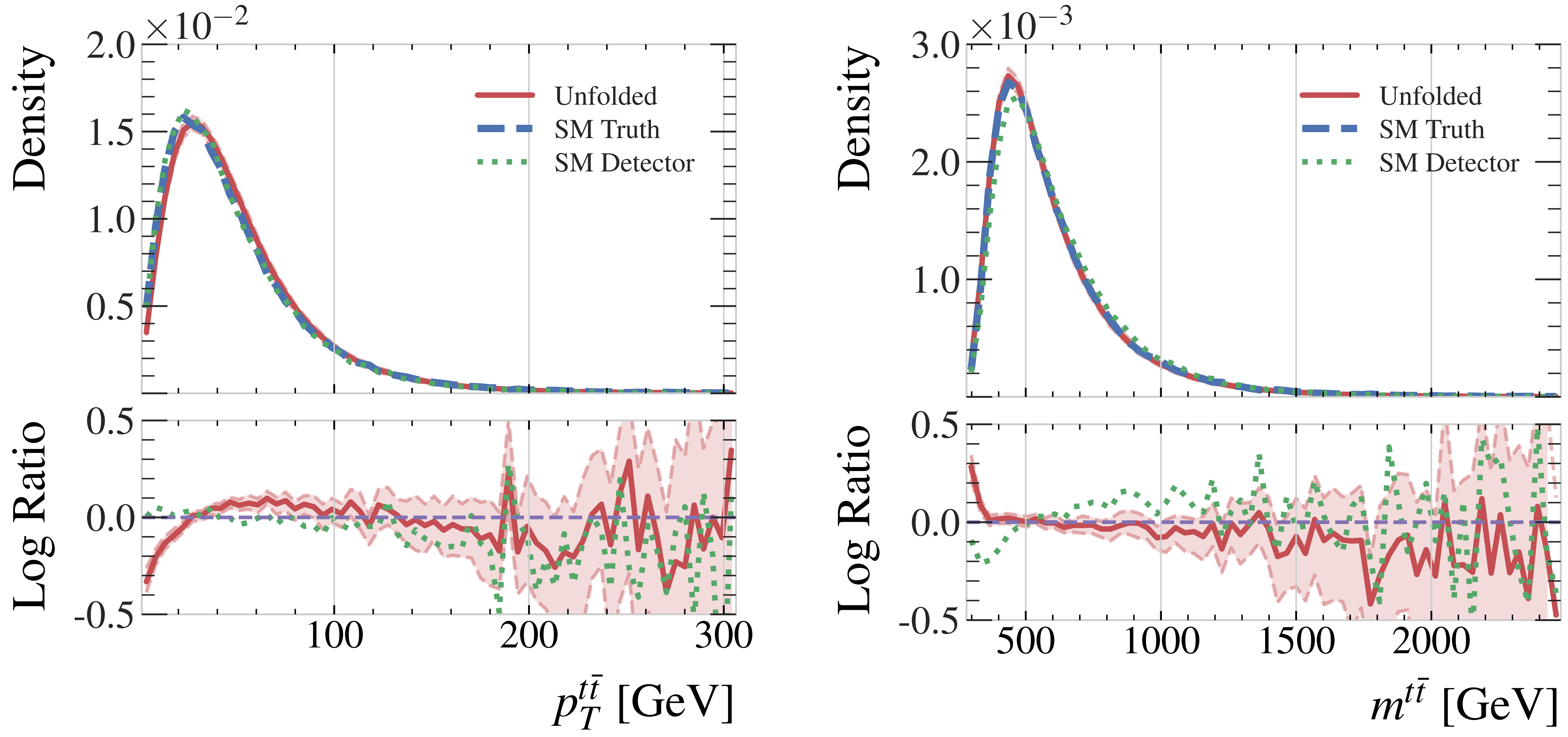
# Leptonic top kinematics

## Assumes pseudo-top jet/parton assignment



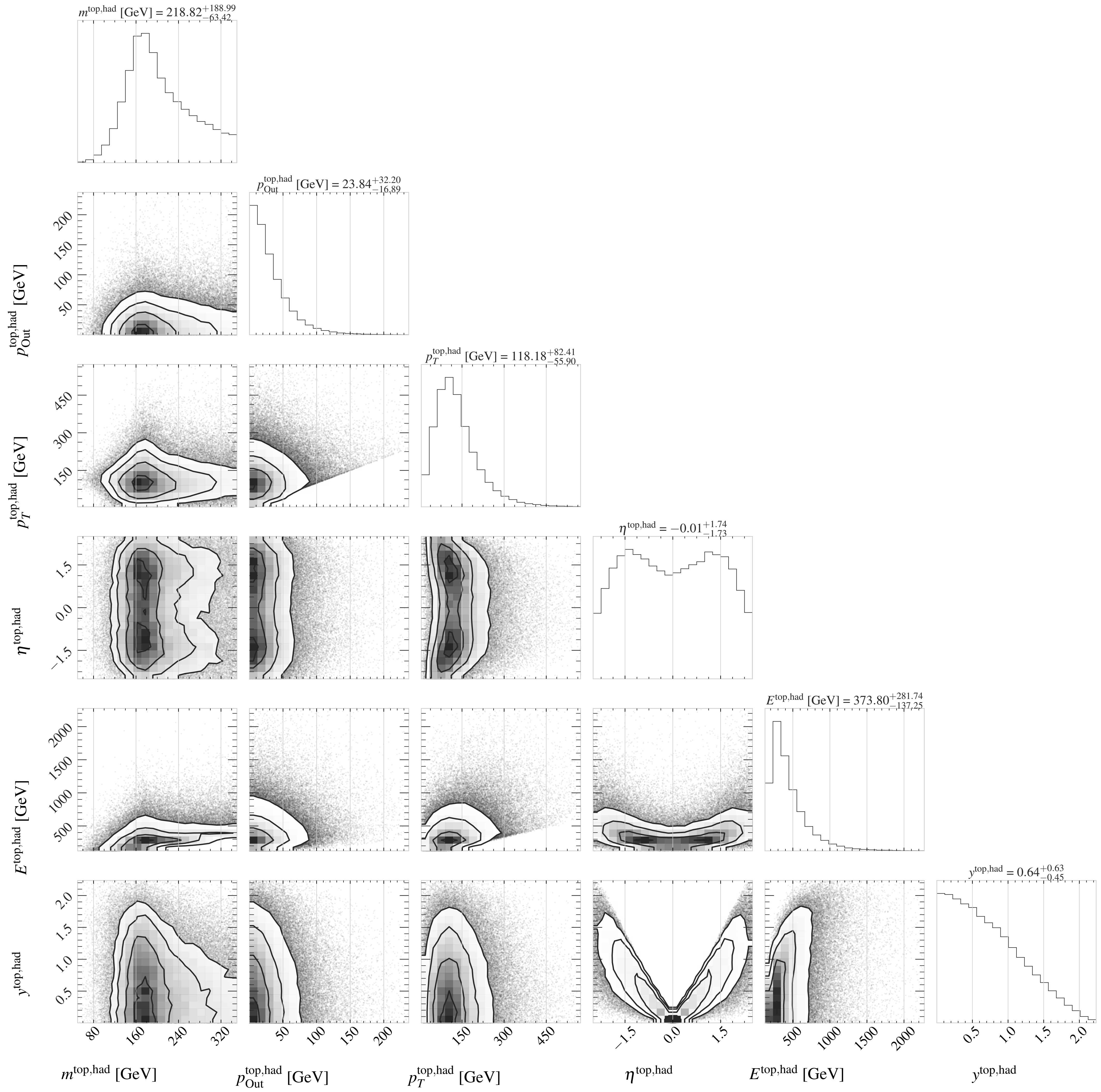
# $t\bar{t}$ system kinematics

## Assumes pseudo-top jet/parton assignment

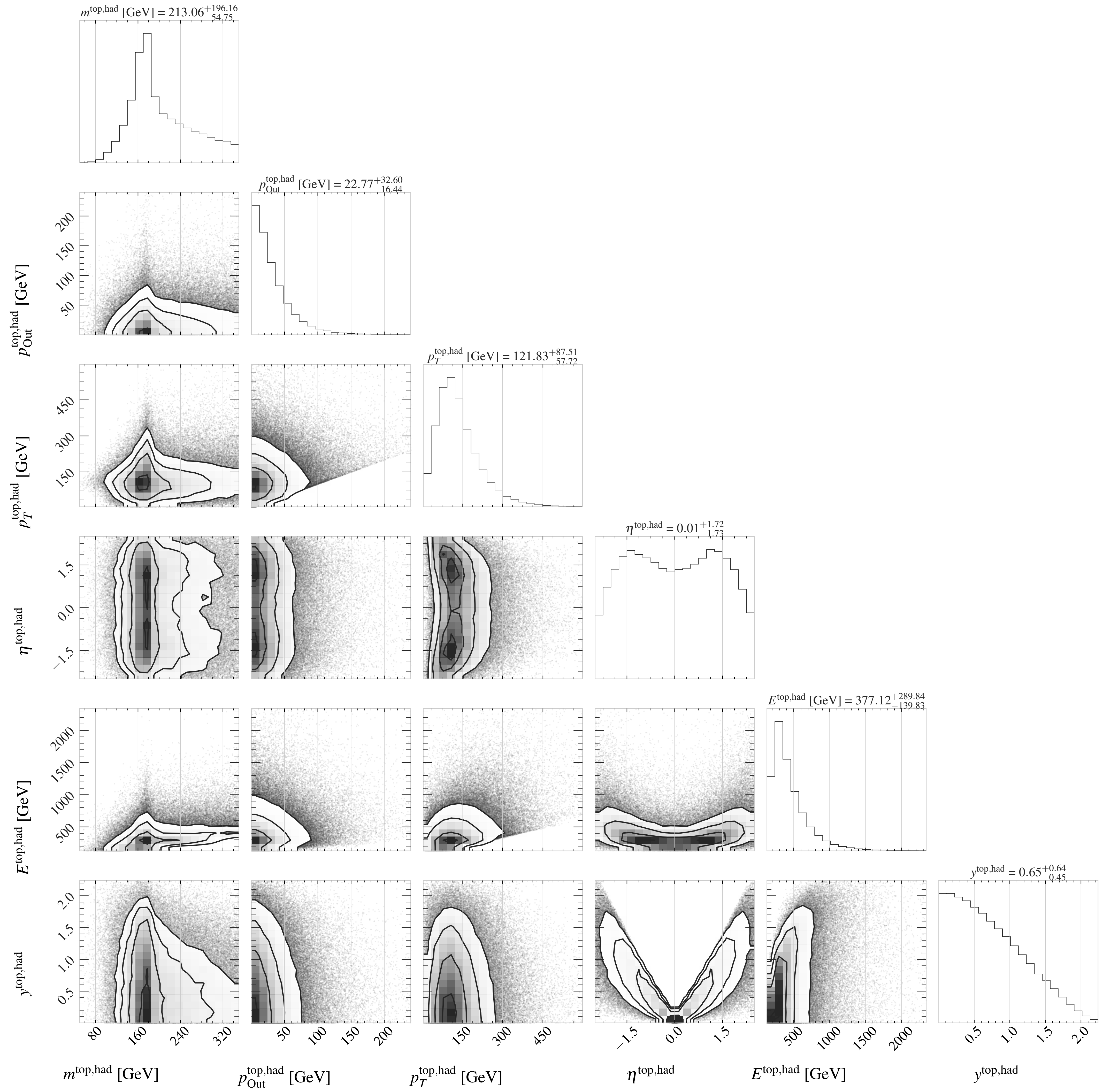



- These distributions are not directly optimized, but are less peaked than hadronic top mass
- Predictions are decent in high $p_T$ and mass events, but struggle in the low kinematic range

# Corner plots




Predicted

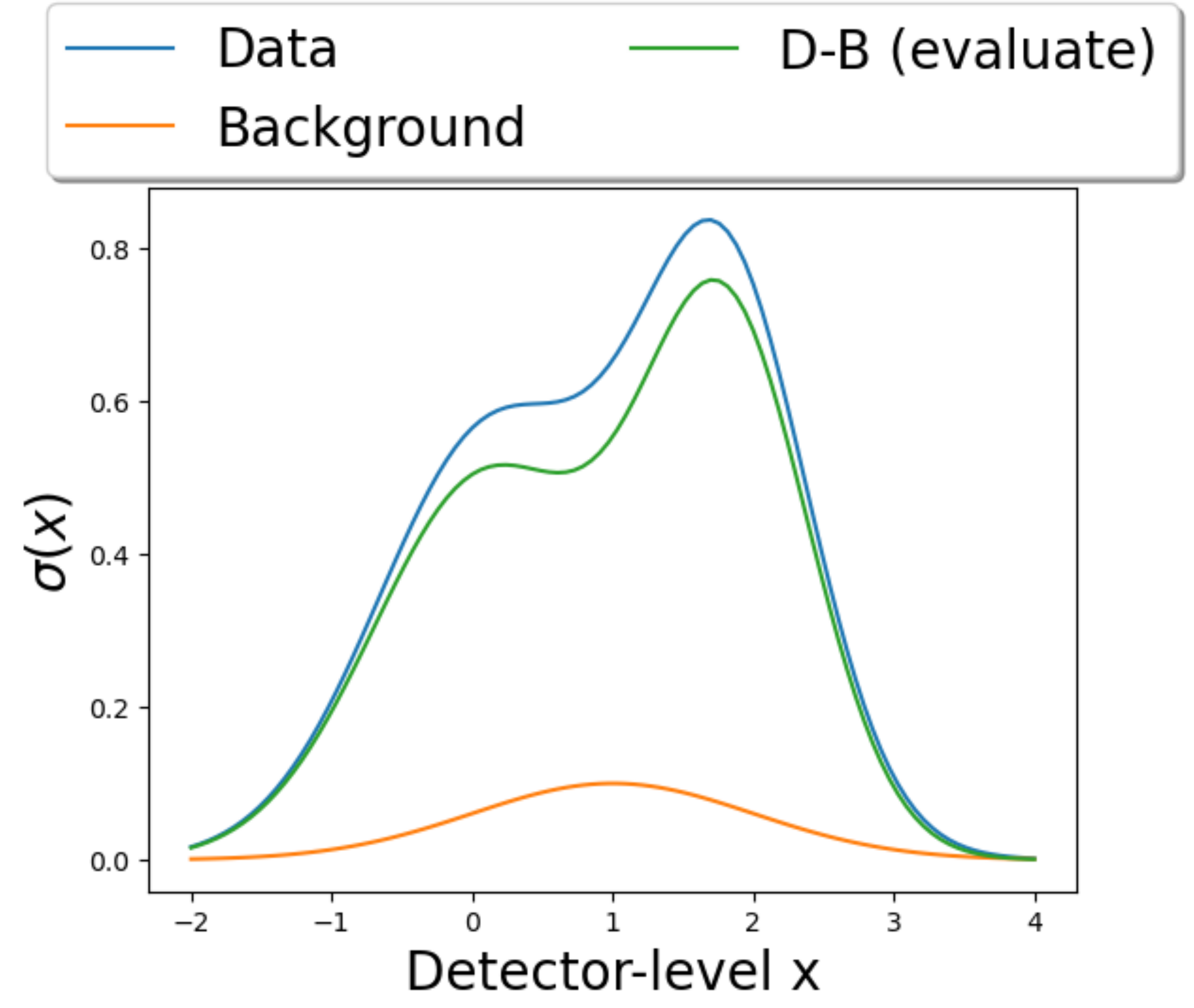



True

# Background subtraction

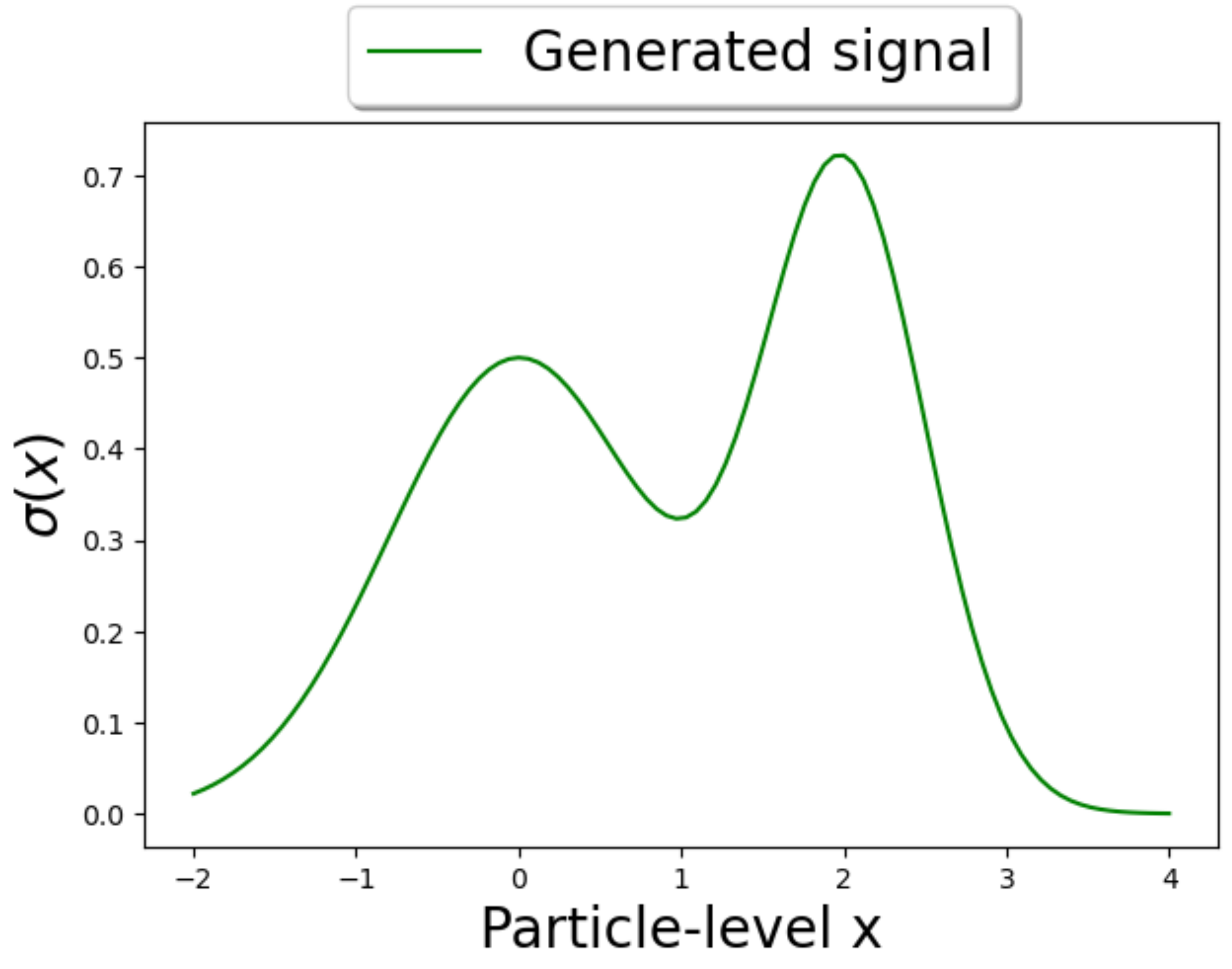
## Before Unfolding

Perform un-binned subtraction of background, then run inference on generative model trained with only signal



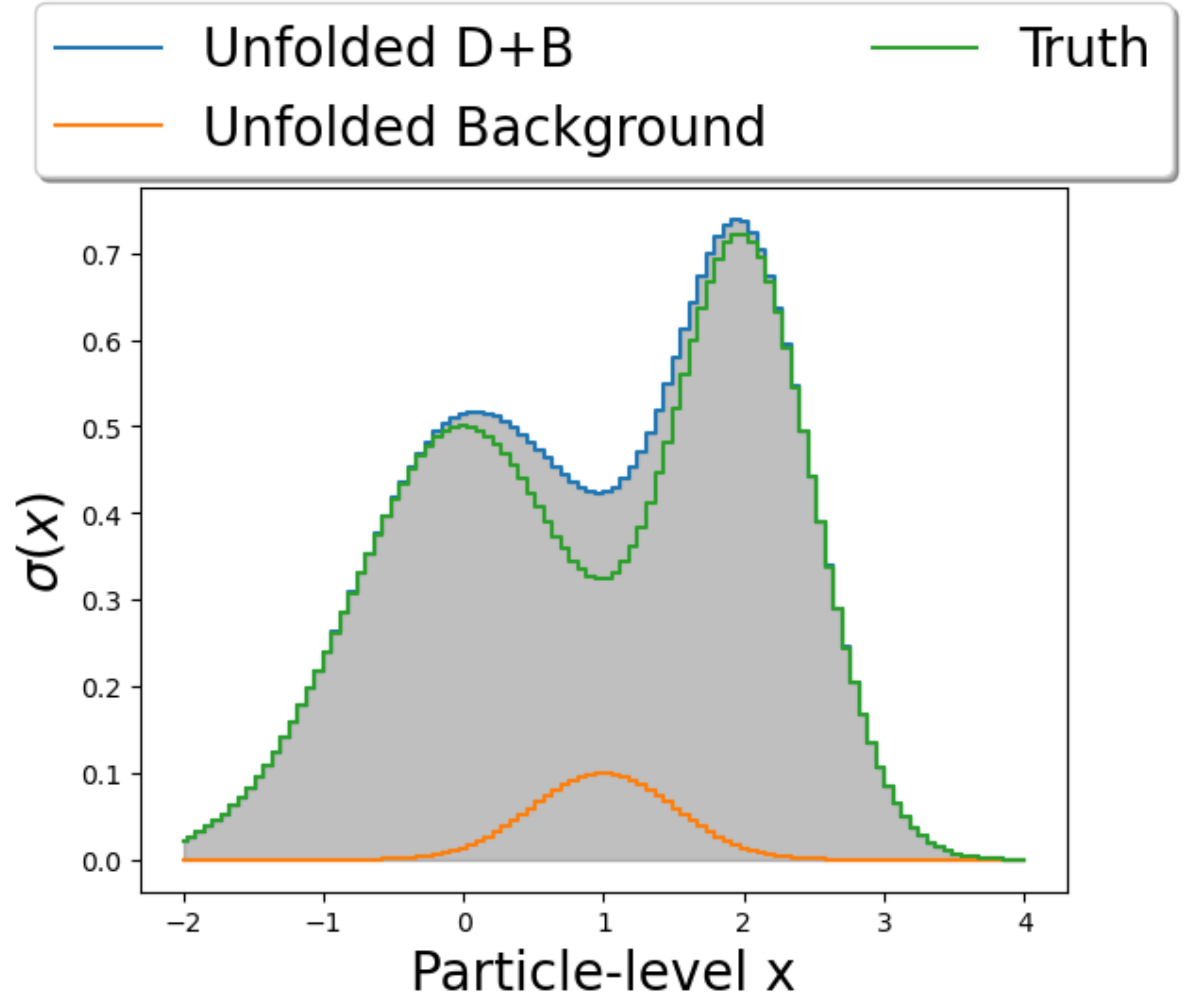

## During Unfolding

Train generative model with negative weights for background events, then run inference on data

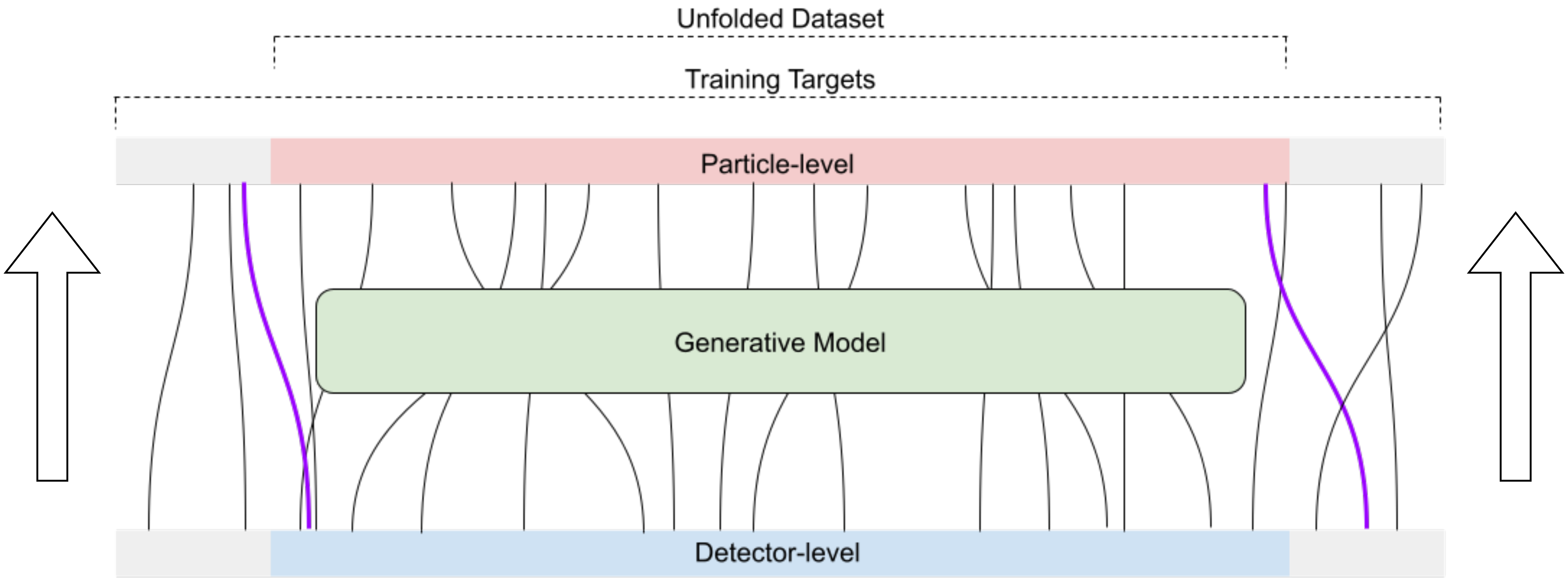



## After Unfolding

Train to unfold S+B together, then run inference on data and perform binned subtraction after unfolding



# Acceptance effects




**Fakes:**
Train generative model on large phase space region then place cuts on particle-level phase space after evaluation

**Inefficiencies:**
Challenging, since have no event to condition generation. Could reweight MC to match unfolded data at particle level.

# Other point-cloud conditional generative models

- The primary use case is fast generation / calorimeter simulation
- Set conditional set generation of jets: slot attention, graph diffusion
  - Generate reconstructed jet based on particle-level constituents
  - Note this is learning the detector simulation forward operator
- JetNet/JetClass datasets: mpgan, pc-jedi/droid, fpcd, mean-field gan, epic-gan, epic-jedi, deeptree gan, epic-fm
  - Fixed length conditions (jet $p_T$, mass, constituent multiplicity, particle type)
- ILD calorimeter simulation dataset: calocloads, calopointflow
  - Fixed length conditions (energy, number of shower points)
- Calo-challenge datasets: summary
  - Fixed length conditions (energy)
- I am likely missing more than a few!
- Conditioning is very different between unfolding and fast generation / calorimeter simulation