Learning Standard Model structure from LHC data with Riemannian Flow Matching

Based on [2607.16144]

Kevin A. Urquía Calderón

in collaboration with Midori Kato, Inar Timiryasov, and Oleg Ruchayskiy

Conference/Workshop
HAMLET – PHYSICS

KØBENHAVNS UNIVERSITET
UNIVERSITY OF COPENHAGEN

Website

Website of the paper: `https://hep-ssl-webapp.pages.dev`

Outline

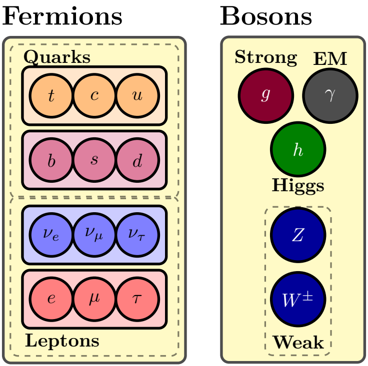
Introduction

KØBENHAVNS UNIVERSITET
UNIVERSITY OF COPENHAGEN

Introduction: particle physics

Particle physics deals with **fundamental particles**. The framework we use for particle physics is based on relativistic quantum mechanics (quantum field theory).

Interactions between particles are described by the **Standard Model**. Every interaction and particle the SM has predicted we have found, the last piece of the puzzle was the discovery of the Higgs boson.



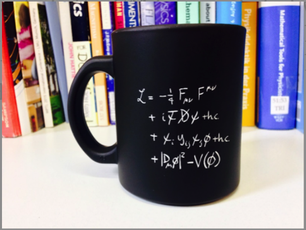
Introduction: particle physics

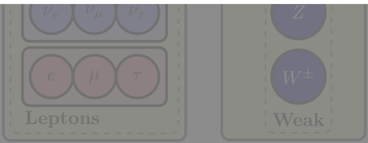
$$\mathcal{L}_{\rm SM} = -\frac{1}{2}\,{\rm tr}[\mathbf{W}_{\mu\nu}\,\mathbf{W}^{\mu\nu}] - \frac{1}{4}B_{\mu\nu}\,B^{\mu\nu} - \frac{1}{2}\,{\rm tr}[\mathbf{G}_{\mu\nu}\,\mathbf{G}^{\mu\nu}] + \sum_{\Psi}\bar{\Psi}\not{D}\Psi + |D_\mu H|^2$$

$$-\Big(\sum \bar{L}_L \cdot H\,Y_\ell\,\ell_R + \sum \bar{Q}_L \cdot H\,Y_D\,D_R + \sum \bar{Q}_L \cdot \tilde{H}\,Y_U\,U_R + {\rm H.c.}\Big)$$

$$+\,\mu^2|H|^2 - \lambda|H|^4\,,$$

Small enough to fit in a mug!

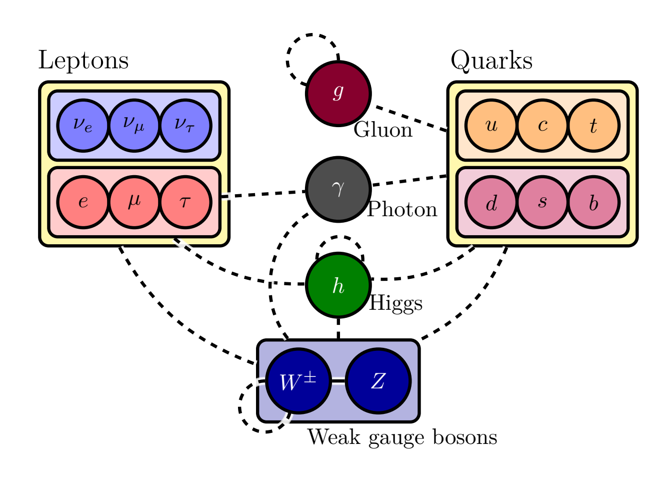




Particle physics deals with **fundamental particles**. The framew… physics… quantu… field th…

Interac… are des… **Model**… particle… have fo… puzzle… Higgs b…

Fermions Bosons



LHC experiment and ATLAS detector

The **Large Hadron Collider** is a 27 km radius proton – proton collider. It operates at a center of mass energy of 14 TeV.

It records tens of petabytes of data per year. It is currently on shutdown, but the next phase, the high luminosity LHC will record much (much!) more data.

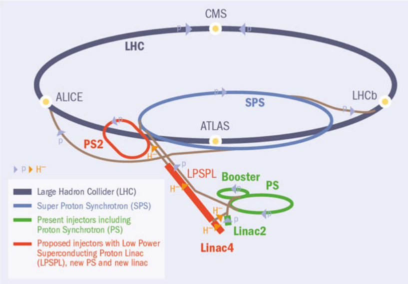


Image from CERN Courier

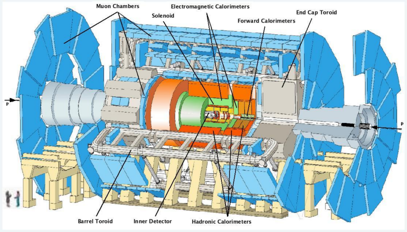


Image from UC Irvine

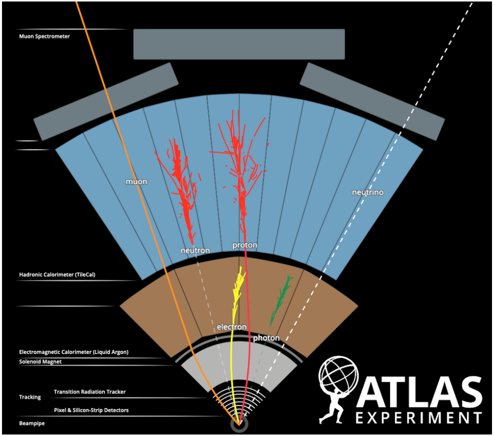
ATLAS

- A general-purpose detector
- Some objectives include: precision measurement, discovery of new physics...
- It's like an onion (or ogre), it has layers

Again, it's like an onion (or an ogre)

ATLAS detects:

- **Electrons**: interact in the tracker and ECAL
- **Muons**: travels the whole detector
- Jets: interact until the hadronic calorimeter. Types of *jets:*
  - **Small-R jets**
  - **Large-R jets**
  - **Hadronic τ**
- **Photons**: only interact ECAL



ATLAS Data Collection

ATLAS released some data from Run 2, for anyone to access it, and do any analysis with it. They claim it is mostly for education purposes, but researchers have done analyses on them (like us!)

Dataset	**Selection requirement**	**Size [GB]**	**Events**	**Frac. [%]**
Full release	No additional selection	1200	5,770,706,892	100.00
1LMET30	≥ 1 **lepton,** $p_T > 7$ **GeV;** $E_T^{\text{miss}} > 30$ **GeV**	**133.4**	**605,490,333**	**10.49**
2to4lep	**2–4 leptons, all with** $p_T > 7$ **GeV**	**74.2**	**283,951,733**	**4.92**
2bjets	2 jets $p_T > 20$ GeV, ≥ 1 b-tagged	36.2	141,259,550	2.45
3J1LMET30	≥ 3 jets $p_T > 20$, ≥ 1 lepton $p_T > 3$, $E_T^{\text{miss}} > 30$ GeV	36.6	76,407,431	1.32
2muons	≥ 2 muons $p_T > 10$ GeV	18.3	72,563,712	1.26
3lep	≥ 3 leptons $p_T > 7$ GeV	5.5	17,193,618	0.30
exactly3lep	exactly 3 leptons $p_T > 7$ GeV	4.8	15,677,770	0.27
2J2LMET30	≥ 2 jets $p_T > 20$, ≥ 2 leptons $p_T > 7$, $E_T^{\text{miss}} > 30$ GeV	1.9	6,242,521	0.11
4lep	≥ 4 leptons $p_T > 7$ GeV	0.627	1,515,848	0.03
exactly4lep	exactly 4 leptons $p_T > 7$ GeV	0.555	1,362,611	0.02

You can access the datasets in **this link**. We worked with the `2to4lep` and `1L30MET` datasets.

Working with these datasets was physics motivated

- The `2to4lep` dataset contains known resonances in the invariant mass spectrum: $\Upsilon, J/\psi, Z$
- The `1L30MET` dataset contains the cleanest channel for W and top quark

Generative models of continuous data

As of 2026, the dominant paradigm for generation of continuous data is **transport-based generative modeling**. These models construct a continuous path between two distributions (e.g. from noise to data).

Conditional flow matching

Model learns a velocity field u_t^θ whose flow transports the noise distribution into the data distribution:

$$\mathrm{d}X_t = u_t^\theta(X_t)\,\mathrm{d}t, \qquad X_0 \sim p_{\mathrm{init}}, \;\; 0 \le t \le 1\,,$$

so that $X_1 \sim p_{\mathrm{data}}$.

Our loss is:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,z,x}\left|\left|u_t^\theta(x) - u_t^{\mathrm{target}}(x \mid z)\right|\right|^2$$

Example of a conditional path: $x_t = (1-t)\,\epsilon + t\,z$ where $\epsilon \sim \mathcal{N}(0, I)$. The target velocity is $u_t(x \mid z) = z - \epsilon$.

Generative models of continuous data

As of 2026, the dominant paradigm for generation of continuous data is **transport-based generative modeling**. These models construct a continuous path between two distributions (e.g. from noise to data).

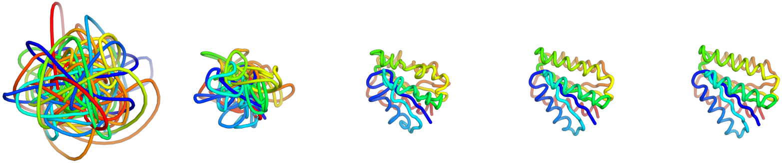
Applications in image generation:

Images generated with `FLUX.2 [klein] 4B`

Generative models of continuous data

As of 2026, the dominant paradigm for generation of continuous data is **transport-based generative modeling**. These models construct a continuous path between two distributions (e.g. from noise to data).

Applications in biology: protein generation



Protein generated with `FoldFlow`

Our motivation and novelty

Open question:
Can a single generative model successfully capture physics recorded at the Large Hadron Collider?

Why should we care?

- The next LHC runs will record an insurmountable amount of data, and we must take as much advantage of it
- Our model can provide a further step towards bigger analyses on LHC data

Method

Our goal: **generate collider physics events at reconstruction level**, using the ATLAS Open Data datasets as training examples. **Motivation**: can a single generative model learn to predict SM histograms?

Our first approach was: the model sees the kinematics distributions from ATLAS data $z = (p_T, \eta, \phi, E) \in \mathbb{R}^4$, and learns the velocity field that transports noise to ATLAS data.

		p_T	η	ϕ	E
Event 1	μ^-	45.16	0.71	1.79	57.20
	μ^+	35.80	2.41	3.10	201.55
	j	54.64	−0.05	−0.76	55.08
Event 2	μ^-	84.77	0.30	0.00	88.62
	μ^+	51.58	0.32	−1.44	54.24
	j	92.70	−2.37	2.04	502.28
	j	46.80	−1.09	−2.63	78.03
Event 3	μ^-	32.41	0.03	1.54	32.42
	μ^+	68.60	0.21	−0.97	70.13
	γ	45.31	−0.11	−3.09	45.57

Generative models for collider physics

Our goal: **generate collider physics events at reconstruction level**, using the ATLAS Open Data datasets as training examples. **Motivation**: can a single generative model learn to predict SM histograms?

Our first approach was: the model sees the kinematics distributions from ATLAS data $z = (p_T, \eta, \phi, E) \in \mathbb{R}^4$, and learns the velocity field that transports noise to ATLAS data.

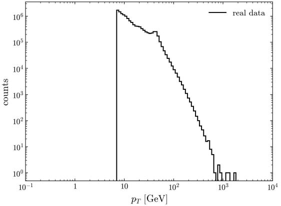
		p_T	η	ϕ	E
	μ^-	$\boxed{45.16}$	0.71	1.79	57.20
Event 1	μ^+	$\boxed{35.80}$	2.41	3.10	201.55
	j	54.64	−0.05	−0.76	55.08
	μ^-	$\boxed{84.77}$	0.30	0.00	88.62
	μ^+	$\boxed{51.58}$	0.32	−1.44	54.24
Event 2	j	92.70	−2.37	2.04	502.28
	j	46.80	−1.09	−2.63	78.03
	μ^-	$\boxed{32.41}$	0.03	1.54	32.42
Event 3	μ^+	$\boxed{68.60}$	0.21	−0.97	70.13
	γ	45.31	−0.11	−3.09	45.57

Generative models for collider physics

Our goal: **generate collider physics events at reconstruction level**, using the ATLAS Open Data datasets as training examples. **Motivation**: can a single generative model learn to predict SM histograms?

Our first approach was: the model sees the kinematics distributions from ATLAS data $z = (p_T, \eta, \phi, E) \in \mathbb{R}^4$, and learns the velocity field that transports noise to ATLAS data.

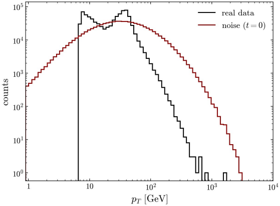
		p_T	η	ϕ	E
	μ^-	45.16	0.71	1.79	57.20
Event 1	μ^+	35.80	2.41	3.10	201.55
	j	54.64	−0.05	−0.76	55.08
	μ^-	84.77	0.30	0.00	88.62
Event 2	μ^+	51.58	0.32	−1.44	54.24
	j	92.70	−2.37	2.04	502.28
	j	46.80	−1.09	−2.63	78.03
	μ^-	32.41	0.03	1.54	32.42
Event 3	μ^+	68.60	0.21	−0.97	70.13
	γ	45.31	−0.11	−3.09	45.57

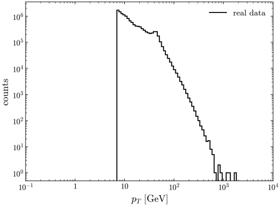


Our goal: **generate collider physics events at reconstruction level**, using the ATLAS Open Data datasets as training examples. **Motivation**: can a single generative model learn to predict SM histograms?

Our first approach was: the model sees the kinematics distributions from ATLAS data $z = (p_T, \eta, \phi, E) \in \mathbb{R}^4$, and learns the velocity field that transports noise to ATLAS data.

		p_T	η	ϕ	E
	μ^-	$\boxed{45.16}$	0.71	1.79	57.20
Event 1	μ^+	$\boxed{35.80}$	2.41	3.10	201.55
	j	54.64	−0.05	−0.76	55.08
	μ^-	$\boxed{84.77}$	0.30	0.00	88.62
Event 2	μ^+	$\boxed{51.58}$	0.32	−1.44	54.24
	j	92.70	−2.37	2.04	502.28
	j	46.80	−1.09	−2.63	78.03
	μ^-	$\boxed{32.41}$	0.03	1.54	32.42
Event 3	μ^+	$\boxed{68.60}$	0.21	−0.97	70.13
	γ	45.31	−0.11	−3.09	45.57



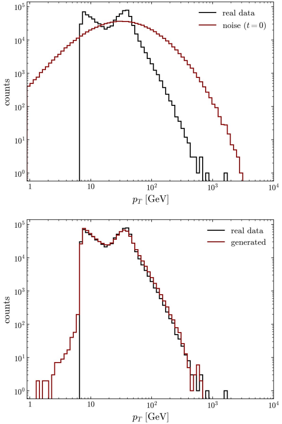


Generative models for collider physics

Our goal: **generate collider physics events at reconstruction level**, using the ATLAS Open Data datasets as training examples. **Motivation**: can a single generative model learn to predict SM histograms?

Our first approach was: the model sees the kinematics distributions from ATLAS data $z = (p_T, \eta, \phi, E) \in \mathbb{R}^4$, and learns the velocity field that transports noise to ATLAS data.

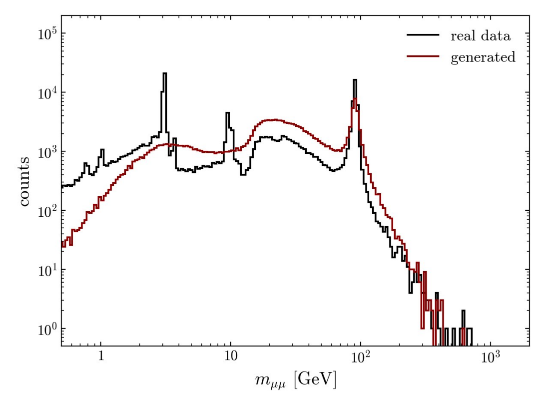
		p_T	η	ϕ	E
Event 1	μ^-	45.16	0.71	1.79	57.20
	μ^+	35.80	2.41	3.10	201.55
	j	54.64	−0.05	−0.76	55.08
Event 2	μ^-	84.77	0.30	0.00	88.62
	μ^+	51.58	0.32	−1.44	54.24
	j	92.70	−2.37	2.04	502.28
	j	46.80	−1.09	−2.63	78.03
Event 3	μ^-	32.41	0.03	1.54	32.42
	μ^+	68.60	0.21	−0.97	70.13
	γ	45.31	−0.11	−3.09	45.57



Consider now the invariant mass spectrum of opposite sign-same flavor muons:

$$m_{\mu\mu}^2 = 2\,m_\mu^2 + 2(E_1 E_2 - \vec{p}_1 \cdot \vec{p}_2)\,,$$

encodes known physics: peaks at $m_{\mu\mu} = m_Z, m_{J/\Psi}, \ldots$. The same model failed to replicate the whole spectrum



Riemannian Flow Matching

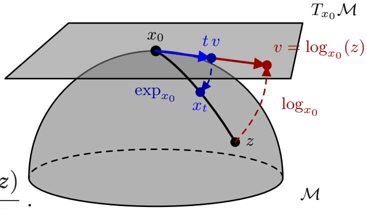
We want to change our approach so that it keeps all of its particles on-shell:

$$E^2 = |\vec{p}|^2 + m^2 \,,$$

all particles must follow this relationship. We can enforce it by using **Riemannian Flow Matching**.

Given a Riemannian manifold $\mathcal{M}$ with metric g, where we want the noise, data, and the path in-between to be in. In this case, the velocity vector, $u_t(x) \in T\,\mathcal{M}$ in the tangent space of the manifold $\mathcal{M}$.

The path from noise to data now must follow **geodesic paths**. The geodesics connecting ϵ and z, and the velocity vector that transports between both; can be expressed in terms of the exp and log mappings



$$x_t = \exp_\epsilon\left(t\,\log_\epsilon(z)\right)\,, \qquad u_t(x \mid z) = \frac{\log_{x_t}(z)}{1-t}\,.$$

We want to change our approach so that it keeps all of its particles on-shell:

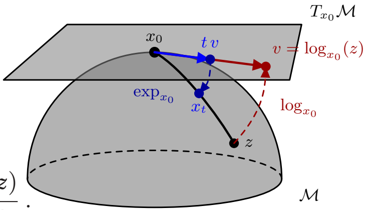
$$E^2 = |\vec{p}|^2 + m^2 \,,$$

all particles must follow this relationship. We can enforce it by using **Riemannian Flow Matching**.

Given a Riemannian manifold $\mathcal{M}$ with metric g, where we want the noise, data, and the path in-between to be in. In this case, the velocity vector, $u_t(x) \in T\,\mathcal{M}$ in the tangent space of the manifold $\mathcal{M}$.

The path from noise to data now must follow **geodesic paths**. The geodesics connecting ϵ and z, and the veloc... between ... terms of the exp and log mappings

Contrast with vanilla CFM:
$x_t = (1-t)\,\epsilon + t\,z \,, u_t = z - \epsilon$



$$x_t = \exp_\epsilon\left(t\,\log_\epsilon(z)\right)\,, \qquad u_t(x \mid z) = \frac{\log_{x_t}(z)}{1-t}\,.$$

Manifold choice

The data we have is written in terms of (p_T, η, ϕ, E, m). Converting it to Euclidean momentum:

$$z = (E, \vec{p}, m)\,,$$

.

This is then encoded to two types of manifolds:

Massless manifold (e, μ, γ)

$$\mathcal{M}_{\text{massless}} = \mathbb{R} \times S^2\,,$$
$$z = (y_E, \hat{n})\,,$$

Massive manifold (jets, τ)

$$\mathcal{M}_{\text{massive}} = \mathbb{R} \times S^2 \times \mathbb{R}\,,$$
$$z = (y_E, \hat{n}, z_m)\,,$$

Chart	**Encode**	**Decode**
Direction (S^2)	$\hat{n} = \mathbf{p}/\lvert\mathbf{p}\rvert$	$\mathbf{p} = \lvert\mathbf{p}\rvert\,\hat{n}$
Log-energy	$y_E = \log\!\big(E/E^k_{\text{ref}}\big)$	$E = E^k_{\text{ref}}\exp(y_E)$
Boost-rapidity χ	$\chi = \operatorname{arccosh}(E/m)$	$E = m\cosh\chi$
On-shell mass*	$m = \sqrt{E^2 - \lvert\mathbf{p}\rvert^2}$, floored	$\lvert\mathbf{p}\rvert = \sqrt{E^2 - m^2}$
Log-mass	$z_m = \log\!\big(m/m^k_{\text{floor}}\big)$	$m = m^k_{\text{floor}}\exp(z_m)$

Manifold choice

The data we have is written in terms of (p_T, η, ϕ, E, m). Converting it to Euclidean momentum:

Explicit form for $\exp_z$ and $\log_z$ maps for S^2 are shown in backup slides

.

This is then encoded to two types of manifolds:

Massless manifold (e, μ, γ)

$$\mathcal{M}_{\text{massless}} = \mathbb{R} \times S^2\,, \quad z = (y_E, \hat{n})\,,$$

Massive manifold (jets, τ)

$$\mathcal{M}_{\text{massive}} = \mathbb{R} \times S^2 \times \mathbb{R}\,, \quad z = (y_E, \hat{n}, z_m)\,,$$

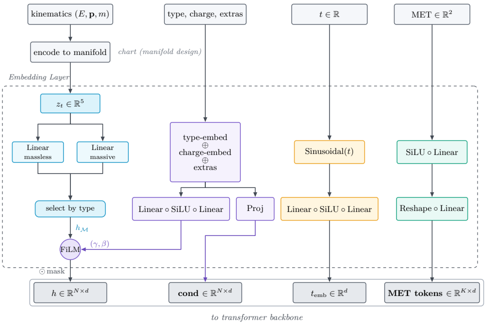
Chart	**Encode**	**Decode**
Direction (S^2)	$\hat{n} = \mathbf{p}/\lvert\mathbf{p}\rvert$	$\mathbf{p} = \lvert\mathbf{p}\rvert\,\hat{n}$
Log-energy	$y_E = \log\big(E/E^k_{\text{ref}}\big)$	$E = E^k_{\text{ref}}\exp(y_E)$
Boost-rapidity χ	$\chi = \operatorname{arccosh}(E/m)$	$E = m\cosh\chi$
On-shell mass*	$m = \sqrt{E^2 - \lvert\mathbf{p}\rvert^2}$, floored	$\lvert\mathbf{p}\rvert = \sqrt{E^2 - m^2}$
Log-mass	$z_m = \log\big(m/m^k_{\text{floor}}\big)$	$m = m^k_{\text{floor}}\exp(z_m)$

Input:

- Kinematics to generate: $(E, \vec{p}, m)$
- Conditioned variables: particle-type, MET, t, detector-features

Output:

- Kinematics embedded in higher dimension: h
- Higher dimensional conditions variables: t_{emb}, cond, MET tokens



Model architecture: backbone model

Input: output of the embedding layer

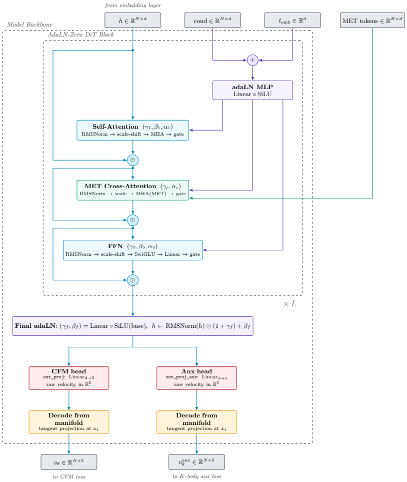
Backbone layer: six transformer adaLN-Zero layers:

- attention
- MET-cross attention
- feed-forward network

each one wrapped in a residual connection (ResNet-style)

Output: two velocity field: $u(t), u(t)^{\text{aux}} \in \mathrm{R}^5$

Two losses: main CFM loss, and an auxiliary loss that focuses on K-invariant masses



Model architecture: backbone model

Input: output of the embedding layer

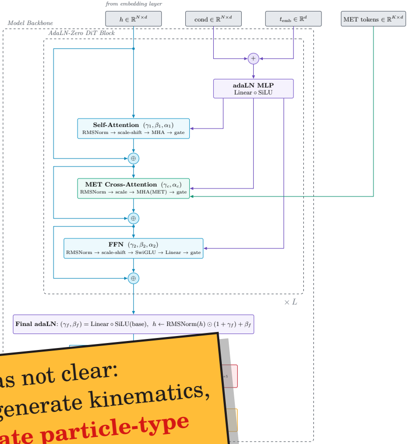
Backbone layer: six transformer adaLN-Zero layers:

- attention
- MET-cross attention
- feed-forward network

each one wrapped in a residual connection (ResNet-style)

Output: two velocity field: $u(t), u(t)^{\text{aux}} \in \mathrm{R}^5$

Two losses: ma
an auxiliary loss
K-invariant mass



In case it was not clear:
the model will only generate kinematics,
it will not generate particle-type
or particle multiplicities

Input: output of the embedding layer

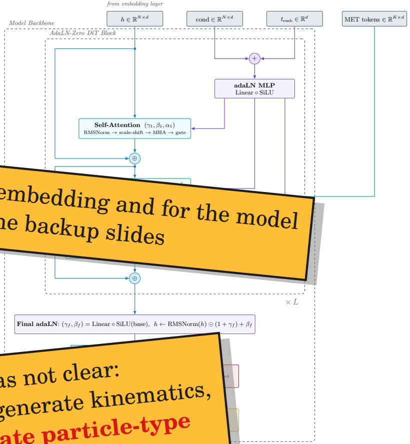
Backbone layer: six transformer adaLN-Zero layers:

- atten
- MET
- feed-forward ne

each one wrapped in a residual connection (ResNet-style)

Output: two velocity field: $u(t), u(t)^{\text{aux}} \in \mathrm{R}^5$

Two losses: ma
an auxiliary loss
K-invariant mass



Relevant functions for both embedding and for the model are defined in the backup slides

In case it was not clear:
the model will only generate kinematics,
it will not generate particle-type
or particle multiplicities

Loss designs

CFM loss: modification from original CFM loss

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}\left[q_i^{\text{eff}} \cdot H_\delta\left(u_i(x_t, t) - u_i(x_t, t \mid z)\right)\right]$$

H_δ: Huber loss
q_i^{eff}: quality-weighting

Loss designs

CFM loss: modification from original CFM loss

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}\left[q_i^{\text{eff}} \cdot H_\delta\left(u_i(x_t, t) - u_i(x_t, t \mid z)\right)\right]$$

Defined in backup slides

H_δ: Huber loss
q_i^{eff}: quality-weighting

Loss designs

CFM loss: modification from original CFM loss

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}\left[q_i^{\mathrm{eff}} \cdot H_\delta\left(u_i(x_t, t) - u_i(x_t, t \mid z)\right)\right]$$

Defined in backup slides

H_δ: Huber loss
q_i^{eff}: quality-weighting

Auxiliary loss: ensures that the model becomes aware of the invariant mass

$$\mathcal{L}_{\mathrm{aux}} = \sum_K \left\langle q_{\mathcal{I}}\, w_I^* \left[\mathbb{I}(t > t_{\mathrm{gate}})\, H_\delta(\Delta_{\mathcal{I}}^{\mathrm{pos}}) + H_\delta\left(\Delta_I^{\mathrm{vel}}\right)\right]\right\rangle ,$$

with

$$\Delta_I^{\mathrm{pos}} = \log m_K^{\mathrm{pred}}(\mathcal{I}) - \log m_K^{\mathrm{truth}}(\mathcal{I}) ,$$
$$\Delta_I^{\mathrm{vel}} = \log \dot{m}_K^{\mathrm{pred}}(\mathcal{I}) - \log \dot{m}_K^{\mathrm{truth}}(\mathcal{I}) ,$$

where $m_K^2 = (E_1 + \cdots + E_K)^2 - (\vec{p}_1 + \cdots + \vec{p}_K)^2$. Predicted kinematics are computed from *one Euler step*. w^* is as KDE distribution-matching weight (defined in backup slides)

CFM loss: modification from original CFM loss

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}\left[q_i^{\text{eff}} \cdot H_\delta\left(u_i(x_t, t) - u_i(x_t, t \mid z)\right)\right]$$

Defined in backup slides

H_δ: Huber loss
q_i^{eff}: quality-weighting

Auxiliary loss: ensures that the model becomes aware of the invariant mass

$$\mathcal{L}_{\text{aux}} = \sum_K \left\langle q_{\mathcal{I}}\, w_I^* \left[\mathbb{I}(t > t_{\text{gate}})\, H_\delta(\Delta_{\mathcal{I}}^{\text{pos}}) + H_\delta\left(\Delta_I^{\text{vel}}\right)\right]\right\rangle ,$$

Teaches the model where the invariant mass should arrive

CFM loss: modification from original CFM loss

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}\left[q_i^{\text{eff}} \cdot H_\delta\left(u_i(x_t, t) - u_i(x_t, t \mid z)\right)\right]$$

Defined in backup slides

H_δ: Huber loss
q_i^{eff}: quality-weighting

Auxiliary loss: ensures that the model becomes aware of the invariant mass

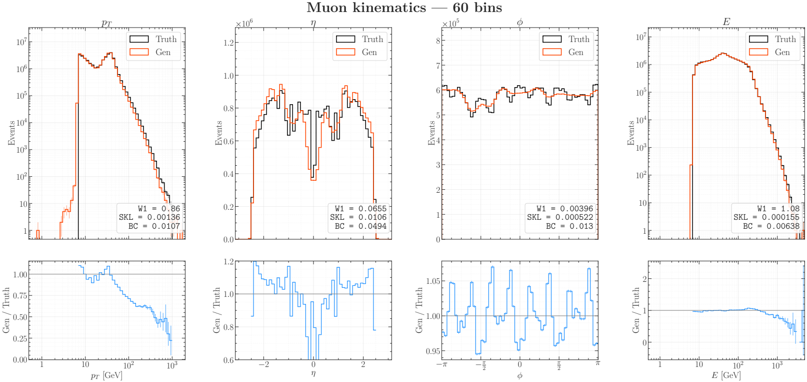
$$\mathcal{L}_{\text{aux}} = \sum_K \left\langle q_{\mathcal{I}}\, w_I^* \left[\mathbb{I}(t > t_{\text{gate}})\, H_\delta(\Delta_{\mathcal{I}}^{\text{pos}}) + H_\delta\left(\Delta_I^{\text{vel}}\right)\right]\right\rangle ,$$

Teaches the model where the invariant mass should arrive

Teaches the direction of travel of the invariant mass

Results

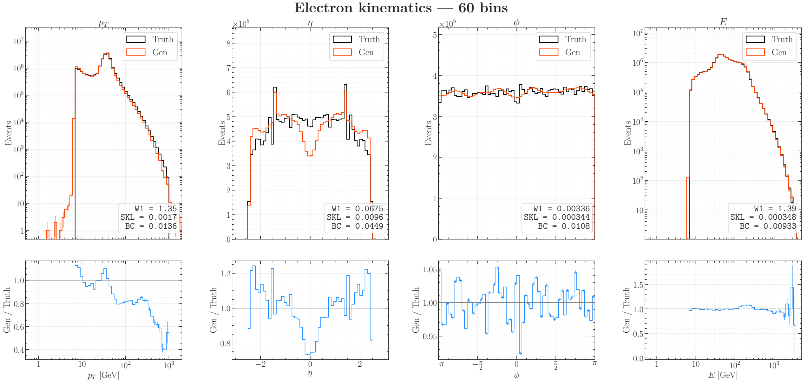
Single-particle kinematics (for example muon)



We can immediately notice

- $p_T < 7\,\text{GeV}$ cut not learned (hard to learn strict cuts like this with any generative model)
- $\eta = 0$ deficit: collider artifact (for some reason also replicated for other particles despite not present in data)
- ϕ relative uniformity is learned

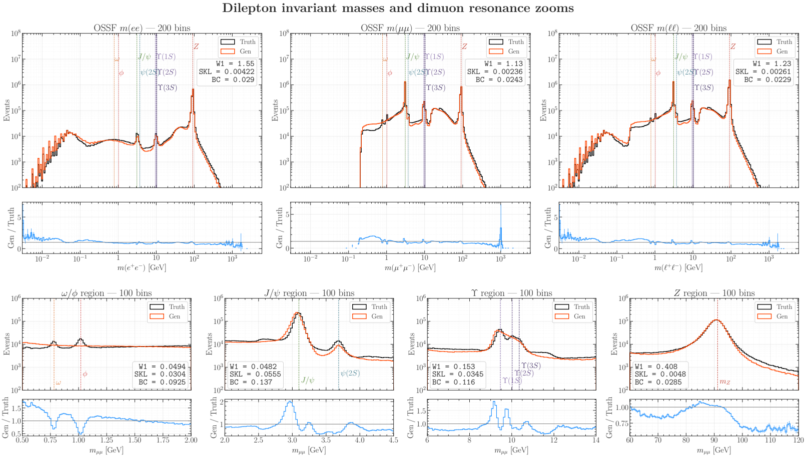
Single-particle kinematics (for example electron)



We can immediately notice

- $p_T < 7\,\text{GeV}$ cut not learned (hard to learn strict cuts like this with any generative model)
- $\eta = 0$ deficit, unkown reason
- ϕ relative uniformity is learned

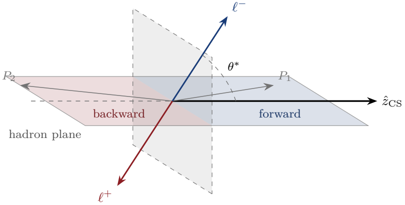
Dilepton spectrum



Plot of $m_{\ell\ell}^2 = 2\,(E_1 E_2 - \vec{p}_1 \cdot \vec{p_2})$.

The model could reproduce very accurately several peaks in the invariant mass spectrum: $Z, J/\Psi, \Psi(2S)$. It merged the Υ resonances.

Measurement of the Weinberg Angle

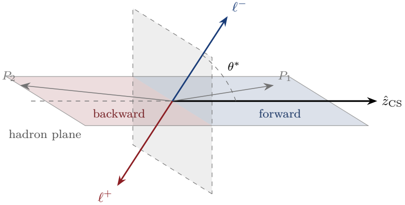


Forward-backwards asymmetry:

$$A_{\mathrm{FB}} = \frac{N_F - N_B}{N_F + N_B}$$

N_F: number of ℓ^- in the forward direction

N_B: number of ℓ^- in the backward direction



Forward-backwards asymmetry:

$$A_{\mathrm{FB}} = \frac{N_F - N_B}{N_F + N_B}$$

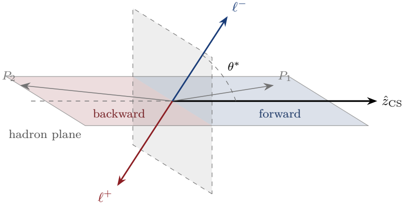
N_F: number of ℓ^- in the forward direction
N_B: number of ℓ^- in the backward direction

Theory predicts A_{FB} depends on the $\sin\theta_W$, the sine of the Weinberg Angle.

$$A_{FB}^q = \frac{3(1-4\ \sin^2\theta_W)(1-4|Q_q|\ \sin^2\theta_W)}{[1+(1-4\sin^2\theta_W)]\,[1+(1-4|Q_q|\sin^2\theta_W)^2]}\,,$$

where Q_q is the charge of the initial quark.

Measurement of the Weinberg Angle



Forward-backwards asymmetry:

$$A_{\mathrm{FB}} = \frac{N_F - N_B}{N_F + N_B}$$

N_F: number of ℓ^- in the forward direction

N_B: number of ℓ^- in the backward direction

Theory predicts A_{FB} depends on the $\sin\theta_W$, the sine of the Weinberg Angle.

$$A^q_{FB} = \frac{3(1 - 4\ \sin^2\theta_W)(1 - 4|Q_q|\ \sin^2\theta_W)}{[1 + (1 - 4\sin^2\theta_W)]\,[1 + (1 - 4|Q_q|\sin^2\theta_W)^2]}\ ,$$

where Q_q is the charge of the initial quark.

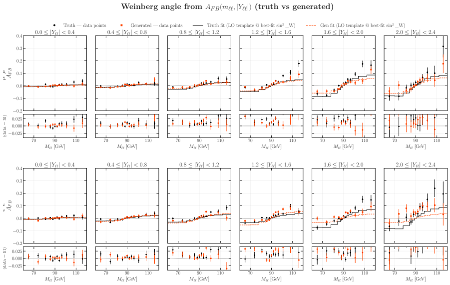
The model was never encouraged to reproduce the forward-backward asymmetry spectrum. If it could reproduce it, it is a small sign it learned the physics.

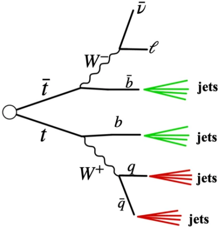
Forward-backward asymmetry histograms at different total rapidities, $Y_{\ell\ell}$ and invariant masses $m_{\ell\ell}$:

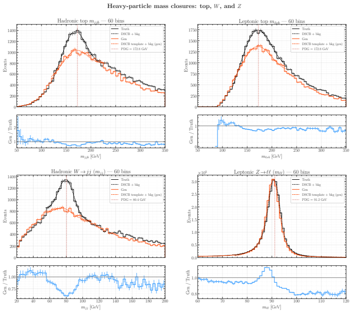
We do a fit using the leading-order template fits of the A_{FB}. We find from the fits that $\sin^2\theta_W$ are

Source	$\sin^2\theta_w$	σ_{fit}	σ_{gen}
$\mu^+\mu^-$ truth	0.23690	±0.00084	—
$\mu^+\mu^-$ generated	0.23657	±0.00082	±0.00055
e^+e^- truth	0.23792	±0.00103	—
e^+e^- generated	0.23627	±0.00092	±0.00073
PDG eff. leptonic	0.23122	—	—
CMS Run-1 NLO	0.23140	—	—

The "generation uncertainty", σ_{gen} was computed using five different seeds during generation.







Semileptonic $t\bar{t}$ process:
$t\bar{t} \to WWbb \to jj\ell\nu bb$.

Observable	**Truth μ**	**Gen μ**	**PDG**
Hadronic top m_{jjb}	165.1 ± 1.2	$167.5 \pm 2.3 \pm 3.6$	172.76
Leptonic top $m_{\ell\nu b}$	166.9 ± 1.3	$166.8 \pm 1.6 \pm 1.9$	172.76
Hadronic W m_{jj}	80.0 ± 0.4	$77.3 \pm 1.8 \pm 3.5$	80.38
Leptonic Z $m_{\ell\ell}$	90.9 ± 0.1	$90.5 \pm 0.1 \pm 0.05$	91.19

Conclusions

Conclusions

What we did:

- We built `ShellFlow`, a generative model whose only physical priors are the **on-shell condition** of particles, and knowledge of **invariant masses**.
- `ShellFlow` managed to capture a substantial part of the Standard Model structure that can be seen in colliders. This includes non-trivial **multi-particle correlations**, positions of heavy-resonances in different multi-particle spectrum of particles, and a reconstruction of the **Weinberg angle** that agrees with PDG values.

Limitations:

Not all the particle correlations are perfect, known issues include: deficit at $\eta = 0$ for all particles, no low-mass resonances (ω, ϕ), no peaks from 4-lepton invariant mass (Z, H), and W, t peaks are not sharp

What is next:

Train with MC data, mechanistic interpretability analysis (can we find *physical* features inside `ShellFlow`?), changing `ShellFlow` to also predict particle multiplicity or PID,...

Acknowledgments

I would like to that our co-authors, since none of this work would have been possible without either of them

- Midori Kato
- Inar Timiryasov
- Oleg Ruchayskiy

We thank **NVIDIA** for providing us access to 8 A100s used during the writing of this project.

We acknowledge the work of the **ATLAS Collaboration** to record or simulate, reconstruct, and distribute the Open Data used in this paper, and to develop and support the software with which it was analysed.

Backup slides

KØBENHAVNS UNIVERSITET
UNIVERSITY OF COPENHAGEN

Collider coordinate system

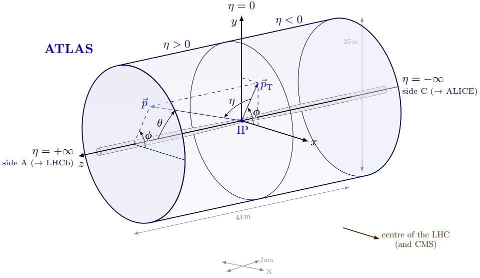
Detector geometry is cylindrically symmetric about the beam, so we work in $(\theta, \phi, p_T, \eta)$ derived from the 4-momentum (p_x, p_y, p_z, E):

$$\theta = \arctan \frac{\sqrt{p_x^2 + p_y^2}}{p_z}$$

$$\phi = \arctan \frac{p_y}{p_x}$$

$$p_T = \sqrt{p_x^2 + p_y^2}$$

$$\eta = -\ln \tan \frac{\theta}{2} = \frac{1}{2} \ln \frac{|\mathbf{p}| + p_z}{|\mathbf{p}| - p_z}$$



Geodesics of different manifolds

Spheres

For $x, y \in S^{d-1}$, the exp and log maps have the form:

$$\log_x(y) = \frac{\theta}{\sin\theta}\big(y + \langle x, y\rangle\, x\big), \qquad \exp_x(v) = \cos\|v\|\, x + \frac{\sin\|v\|}{\|v\|}\, v\,,$$

where $\theta = \arccos(\langle x, y\rangle)$ is the angle between x and y, $\langle x, y\rangle = \sum_i x_i\, y_i$ is the usual dot product, and $\|v\| = \langle v, v\rangle$.

Hyperboloids

For $x, y \in \mathbb{H}^d$, the exp and log maps have the form:

$$\log_x(y) = \frac{\chi}{\sinh\chi}\big(y - \langle x, y\rangle_{\mathbb{M}}\, x\big), \qquad \exp_x(v) = \cosh\|v\|_{\mathbb{M}}\, x + \frac{\sinh\|v\|_{\mathbb{M}}}{\|v\|_{\mathbb{M}}}\, v\,,$$

where $\theta = \operatorname{arccosh}(\langle x, y\rangle_{\mathbb{M}})$ is the hyperbolic angle between x and y, $\langle x, y\rangle_{\mathbb{M}} = \sum_{i,j} g_{ij}\, x_i\, y_j = x_0\, y_0 - \sum_{i>0} x_i\, y_i$ is the Minkowski product, and $\|v\|_{\mathbb{M}} = \langle v, v\rangle_{\mathbb{M}}$.

Here we can see why we chose not to embed the data into a hyperbolic manifold, cosh and sinh would present instabilities during training as they are not bounded functions, unlike cos or sin.

Definition of functions in model layers

Embedding layer functions

$$
\begin{aligned}
\mathrm{SiLU}(x) &= x\,\sigma(x)\,, & \mathrm{FiLM}(h;\gamma,\beta) =& (1+\gamma)\odot h+\beta\,,\\
\mathrm{Sinusoidal}(t)_{2k} &= \sin(1000\,t\,\omega_k)\,, & \mathrm{Sinusoidal}(t)_{2k+1} =& \cos(1000\,t\,\omega_k)\,,\\
\omega_k &= 10000^{-2k/d}\,,
\end{aligned}
$$

where $\sigma(x)$ is a sigmoid function, and $\odot$ is the Hadamard matrix product.

Model functions

$$
\mathrm{FFN(SwiGLU)} = [\mathrm{SiLU}(x\,W_1)\odot x\,W_3]\,W_2\,.
$$

Training details

Hyperparameter	Value
Architecture	
DiT blocks L	6
model width d_{model}	128
attention heads	4
feed-forward expansion ratio	4
MET cross-attention tokens	4
dropout	0
trainable parameters	3.0×10^6
Optimiser (AdamW)	
peak learning rate	6×10^{-4}
weight decay	0.01
(β_1, β_2)	$(0.9, 0.999)$
ε	10^{-8}
gradient clip (global norm)	1.0
EMA decay ρ	0.9995
Learning-rate schedule	
type	cosine
warm-up	10^4 steps, linear
minimum learning rate	10^{-6}
epochs	30
total optimiser steps	717,180

Hyperparameter	Value
Data and batching	
batch size per GPU	4,096
global batch size	32,768
train / validation split	0.95 / 0.05
split seed	42
Loss	
Huber scale δ	1.0
K-body orders	$K \in \{2, 3, 4\}$, equal wt.
position gate t_{gate}	0.99
exponent α	1.0
sharpness k	5
truth-prior mix η	$1/3$ ($w^{\star}_{\max} = 3$)
quality floor $q^{\text{cfm}}_{\text{floor}}$	0.5
Flow-time sampling	
distribution	$t \sim \mathcal{U}[t_{\min}, t_{\max}]$
$(t_{\min}, t_{\max})$	$(0.01, 1 - 10^{-6})$
Compute	
GPUs	8, data-parallel
numerical precision	`bf16-mixed`
compilation	`torch.compile`

Hardware and software specifications

Component	**Specification**
GPU	$8\times$ NVIDIA A100 (80 GB), NVLink
CPU	$2\times$ AMD EPYC 7J13 (128 cores)
System memory	2 TB
Storage	19 TB NVMe RAID array
Software	PyTorch 2.10 (CUDA 12.8), Linux

Functions in loss functions

Huber loss:

$$H_\delta(x) = \begin{cases} \frac{1}{2}x^2\,, & |x| < \delta\,, \\ \delta \cdot \left(|x| - \frac{1}{2}\delta\right), & \text{otherwise}\,, \end{cases}$$

Quality weighting:

$$q_i^{\text{eff}} = q_{\text{floor}}^{\text{cfm}} + (1 - q_{\text{floor}}^{\text{cfm}})\, q_i\,, \qquad q_i^{\text{eff}} \in \left[q_{\text{floor}}^{\text{cfm}}, 1\right]\,,$$

with

$$q_k = \begin{cases} \sigma\big(\alpha_{\text{iso}}^{\text{lep}}(c_{\text{iso}}^{\text{lep}} - r_{\text{iso}})\big)\,\sigma\big(\alpha_{\text{ip}}^{\text{lep}}(c_{\text{ip}}^{\text{lep}} - |d_0/\sigma_{d_0}|)\big) & k \in \{e, \mu\}, \\ \sigma\big(\alpha_{\text{iso}}^{\gamma}(c_{\text{iso}}^{\gamma} - r_{\text{iso}}^{\gamma})\big) & k = \gamma, \\ \sigma\big(\alpha_{\text{jvt}}(\text{JVT} - c_{\text{jvt}})\big) & k = \text{jet}, \\ \sigma\big(\alpha_{\text{rnn}}(s_{\text{rnn}} - c_{\text{rnn}})\big) & k = \tau_{\text{had}}, \\ \sigma\big(\alpha_{D_2}(c_{D_2} - D_2)\big) & k = \text{large-}R\text{ jet}, \\ 1 & \text{otherwise (MET, padding)}. \end{cases}$$

with $\sigma(x) = (1 + e^{-x})^{-1}$, $r_{\text{iso}} = \min(\texttt{ptvarcone30}, \texttt{topoetcone20})/p_T$ and $r_{\text{iso}}^{\gamma} = \min(\texttt{topoetcone40}, \texttt{ptcone20})/p_T$. The values of c and α are shown in the next slide.

More on quality features

Type	**Feature**	**cut** c	**slope** α
Leptons (e, μ)	$\min(\texttt{ptvarcone30}, \texttt{topoetcone20})/p_T$	0.15	10.0
	$\lvert\texttt{d0sig}\rvert$	3.00	1.00
Photon	$\min(\texttt{topoetcone40}, \texttt{ptcone20})/p_T$	0.15	10.0
Jet	`jvt`	0.50	5.00
Tau	`RNNJetScore`	0.50	5.00
Large-R jet	`D2`	1.50	5.00

Functions in loss functions

KDE distribution-matching weight:

$$w^*(r) = \Big[(1-\eta)\, r^k + \eta\Big]^{-\alpha}\,, \qquad r = \frac{\hat{p}_{\text{gen}}}{\hat{p}_{\text{truth}}}\,,$$

gaussian KDE of the K-body mass density,

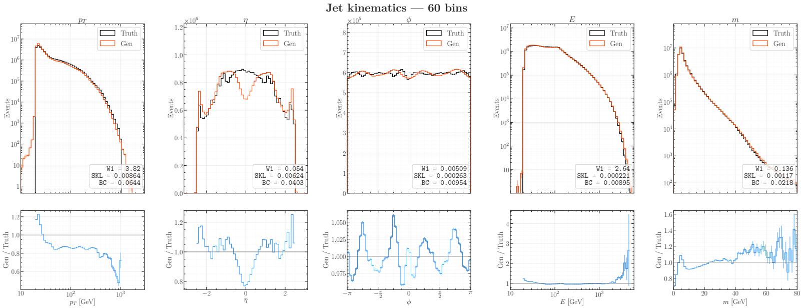
$$\hat{p}(\log m) = \frac{1}{h}\sum \kappa\left(\frac{\log m - \log m_j}{h}\right)\,, \quad \kappa(u) = \frac{1}{\sqrt{2\pi}}\exp\big(-u^2/2\big)\,,$$

Evaluation metrics

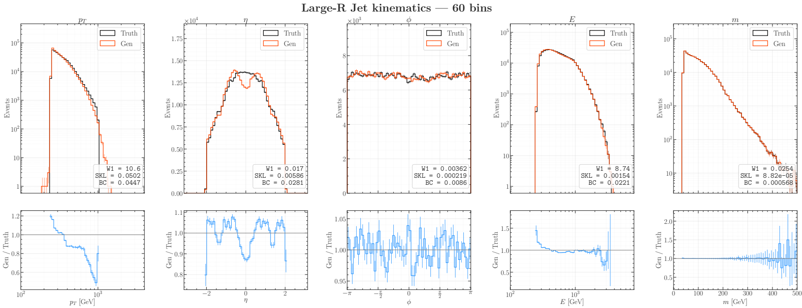
Metrics evaluation distances between distributions x, y used in evaluating the quality of generation:

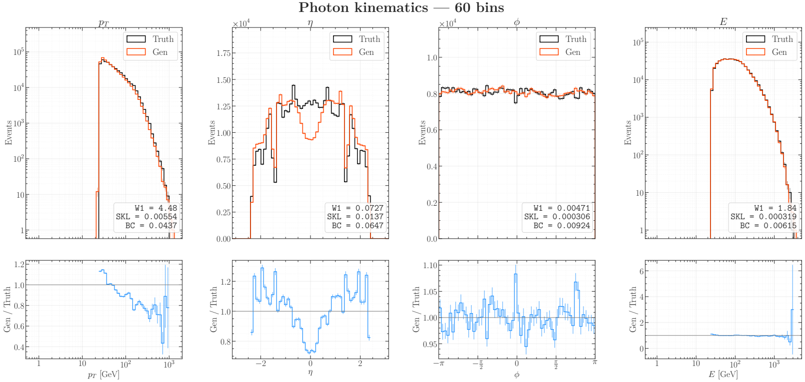
W_1	Wasserstein distance	$\hat{W}_1 = \frac{1}{n}\sum_{i=1}^{N} \lvert x_i - y_i \rvert$
SKL	Symmetric KL divergence	$\mathrm{SKL} = \frac{1}{2}\sum_{i=1}^{n} x_i \log\frac{x_i}{y_i} + y_i \log\frac{y_i}{x_i}$
BC	Bray-Curtis Dissimilarity	$\mathrm{BC} = \frac{\sum_{i=1}^{n} \lvert x_i - y_i \rvert}{\sum_{i=1}^{n} x_i + y_i}$

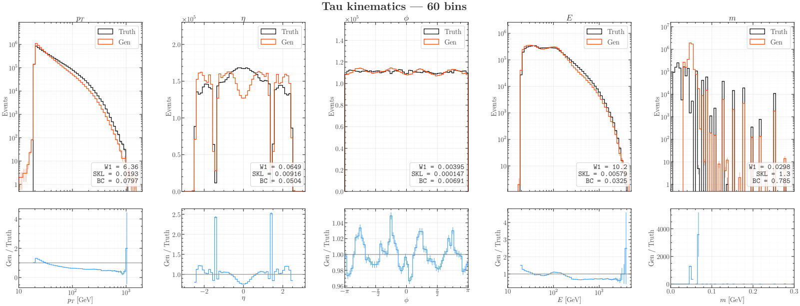
Jet kinematics



Large-R jet kinematics

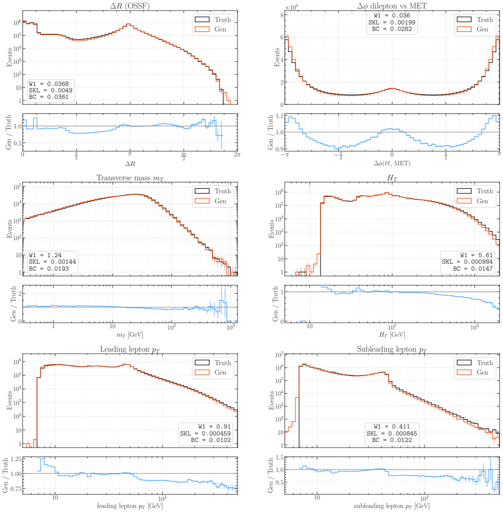



Photon kinematics — 60 bins
p_T
η
ϕ
E
Truth
Gen
Events
W1 = 4.48
SKL = 0.00554
BC = 0.0437
W1 = 0.0727
SKL = 0.0137
BC = 0.0647
W1 = 0.00471
SKL = 0.000306
BC = 0.00924
W1 = 1.84
SKL = 0.000319
BC = 0.00615
Gen / Truth
p_T [GeV]
η
ϕ
E [GeV]

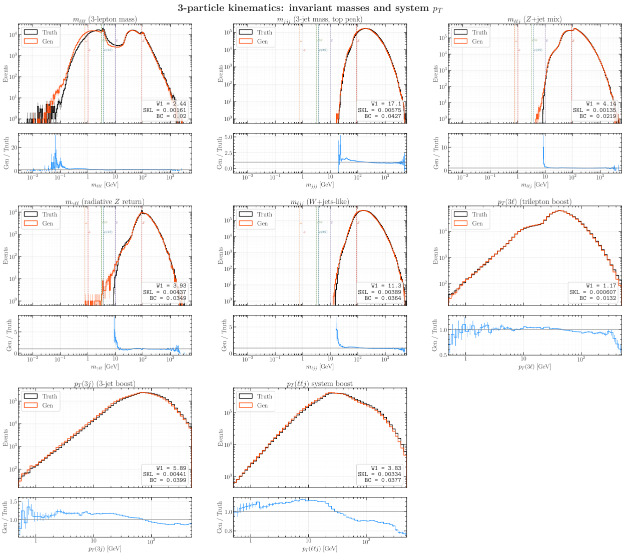

Tau kinematics — 60 bins
p_T
η
ϕ
E
m
Truth
Gen
W1 = 6.36
SKL = 0.0193
BC = 0.0797
W1 = 0.0649
SKL = 0.00916
BC = 0.0504
W1 = 0.00395
SKL = 0.000147
BC = 0.00691
W1 = 10.2
SKL = 0.00579
BC = 0.0325
W1 = 0.0298
SKL = 1.3
BC = 0.785
Events
Gen / Truth
p_T [GeV]
η
ϕ
E [GeV]
m [GeV]

Additional evaluation metrics

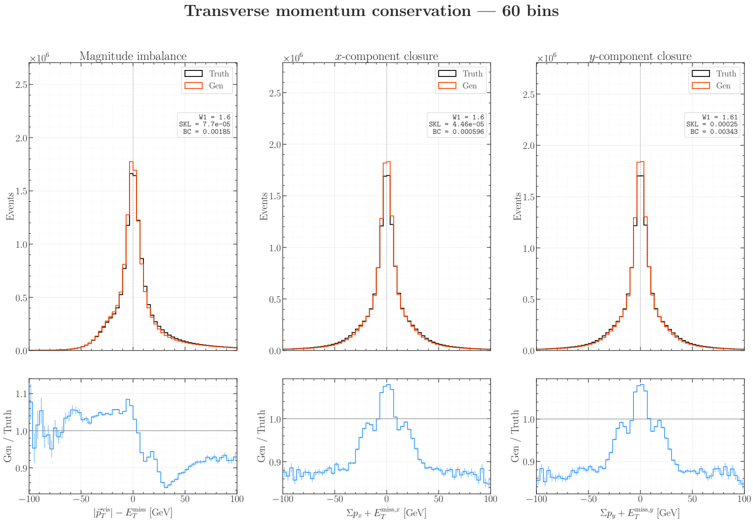
Inter-particle kinematics: ΔR, $\Delta\phi$, m_T, H_T, lepton p_T — 60 bins



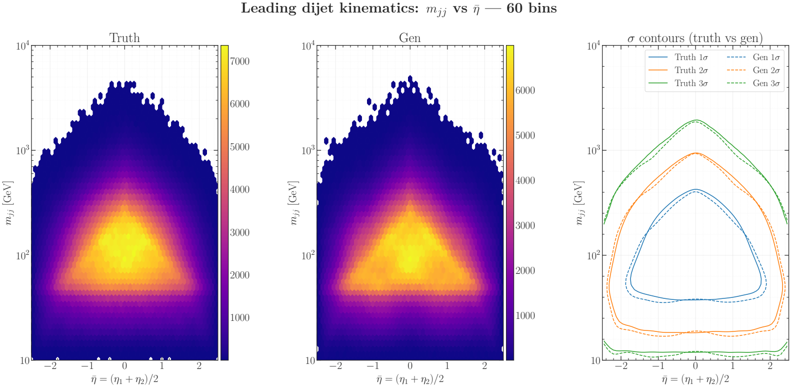
Three-particle features



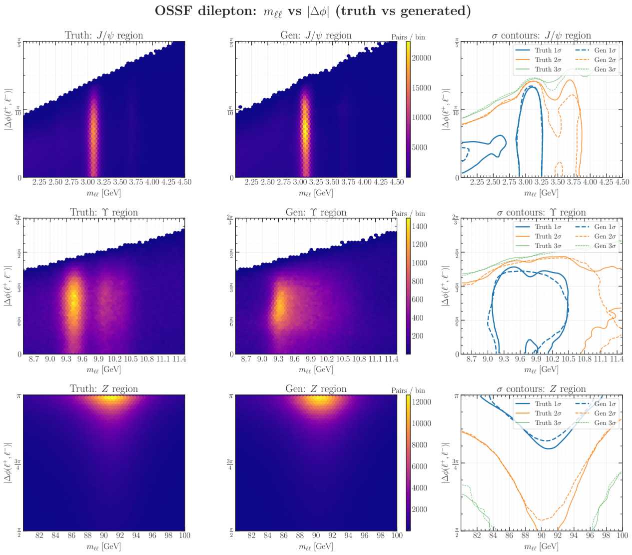
Transverse momentum conservation



Leading di-jet kinematics



OSSF dilepton pairs



Four-lepton invariant mass