EPiC-ly Fast Particle Cloud Generation with Flow Matching and Diffusion

Erik Buhmann, **Cedric Ewen***, Darius Faroughy, Tobias Golling, Gregor Kasieczka, Matthew Leigh, Guillaume Quétant, John Andrew Raine, Debajyoti Sengupta, David Shih

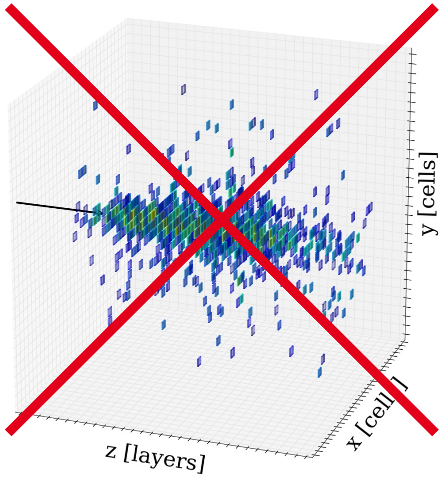
07/11/2023 - ML4Jets2023

arxiv: 2310.00049

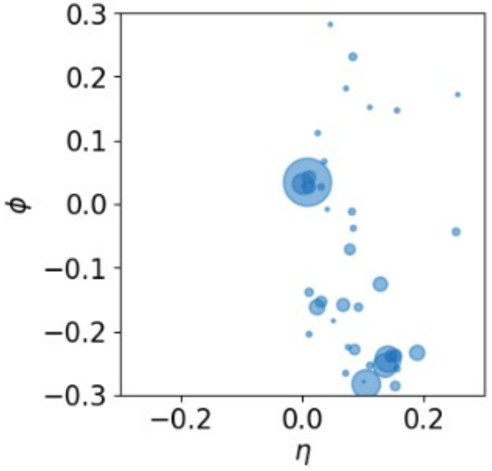
* cedric.ewen@studium.uni-hamburg.de

Generative Simulation in Particle Physics

- Monte Carlo Simulations are time-consuming
- Generative machine learning
  - Train on a small dataset
  - Sample from model
  - ➢ Significantly faster
  - ➢ More data
- Goal: Move from image structures to point clouds
- Point clouds: natural representation for many systems
  - Unordered
  - Variable set cardinalities
- Simulation of jets
  - Complex structure
  - ➢ Good for benchmarking generative models



Photon Shower in a Calorimeter



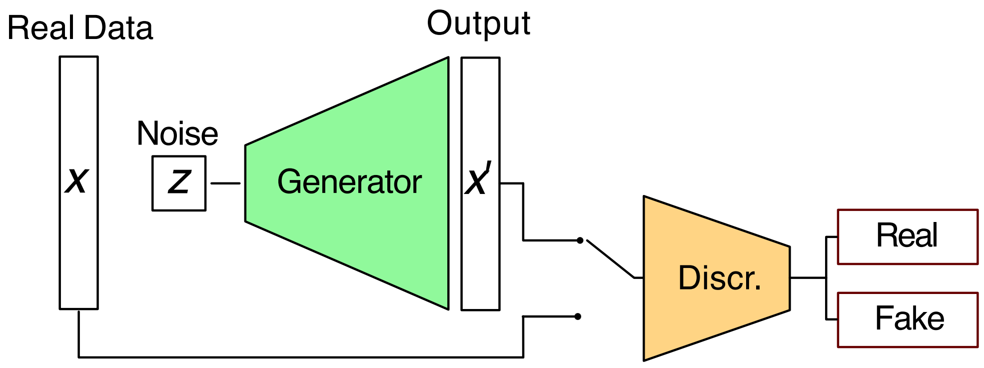
Pythia Generated Top Quark Jet

Butter et al.; GANplifying Event Samples; arxiv:2008.06545

Generative Approaches for Jet Generation

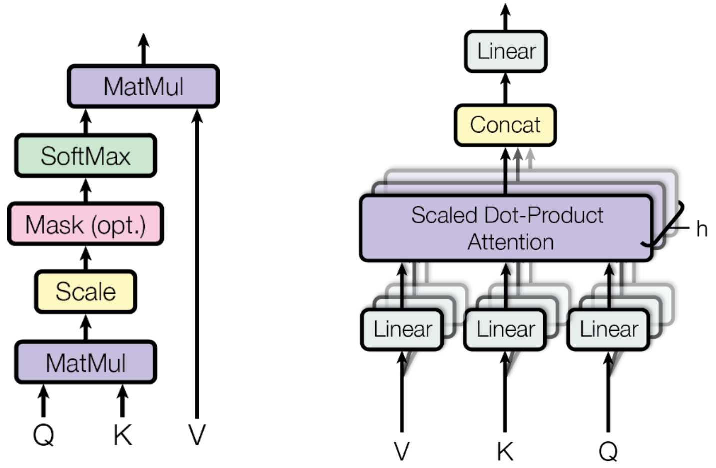
- Requirements for generative models:
  - Permutation equivariance
  - Handle variable set cardinalities

- Previous approaches:
  - MP-GAN [2106.11535]
  - EPiC-GAN (DeepSets based GAN) [2301.08128]
    - Fast but unstable training behaviour
  - PC-JeDi (Transformer based diffusion model) [2303.05376]
    - Slow but powerful training objective



Generative Adversarial Network (GAN)

S. Diefenbacher, PhD thesis



Transformer [1706.03762]

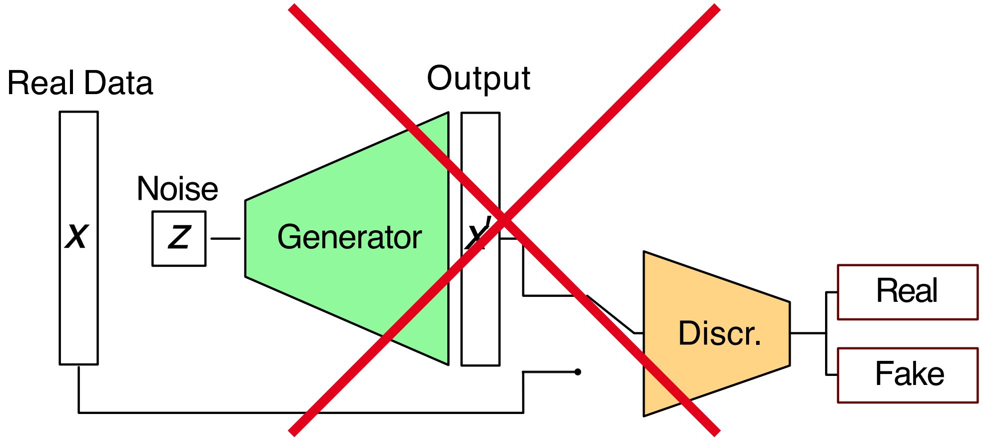
Generative Approaches for Jet Generation

- Requirements for generative models:
  - Permutation equivariance
  - Handle variable set cardinalities

- Previous approaches:
  - MP-GAN [2106.11535]
  - EPiC-GAN (DeepSets based GAN) [2301.08128]
    - Fast but unstable training behaviour
  - PC-JeDi (Transformer based diffusion model) [2303.05376]
    - Slow but powerful training objective

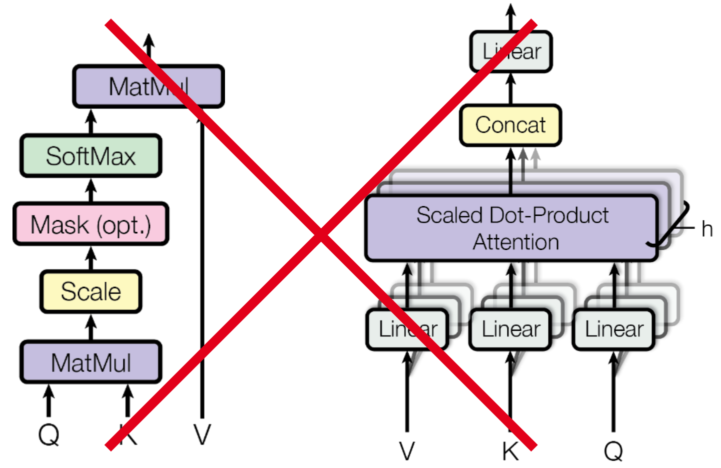
➢ Combine powerful diffusion objective with more scalable and fast EPiC layers

➢ Compare to flow-based Flow Matching objective



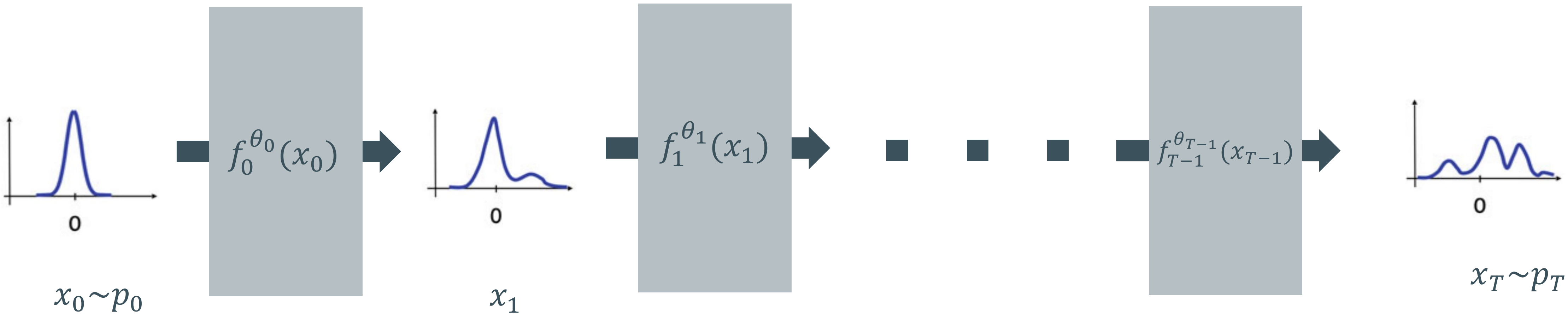
Generative Adversarial Network (GAN)

S. Diefenbacher, PhD thesis



Transformer [1706.03762]

Normalizing Flows



Normalizing Flow (NF)

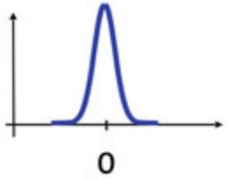
Training: $$\log p_T(x_T) = \log p_0(x_0) - \log \left| \frac{\partial f_t^{\theta}}{\partial x_t} \right|$$

Sampling: $$x_T = f_{T-1} \circ \cdots \circ f_0(x_0)$$

- f must be invertible
- Determinant computationally expensive
  - ➢ Restricted transformations needed

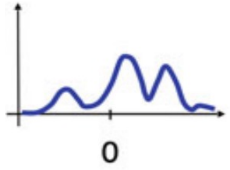
Rezende et al.; Variational Inference with Normalizing Flows; arxiv:1505.05770

Continuous Normalizing Flows



$x_0 \sim p_0$

$$x_t := f(x_t, t) \qquad \frac{\partial x_t}{\partial t} = v_\theta(x_t, t)$$



$x_1 \sim p_1$

	Normalizing Flow (NF)	**Continuous Normalizing Flow (CNF)**
Training:	$\log p_T(x_T) = \log p_0(x_0) - \log \left\lvert \frac{\partial f_t^\theta}{\partial x_t} \right\rvert$	$\log p_1(x_1) = \log p_0(x_0) - \int_{t_0}^{t} Tr\left(\frac{\partial v_\theta}{\partial x_t}\right) dt$
Sampling:	$x_T = f_{T-1} \circ \cdots \circ f_0(x_0)$	Solve ODE (ordinary differential equation)

Normalizing Flow (NF):

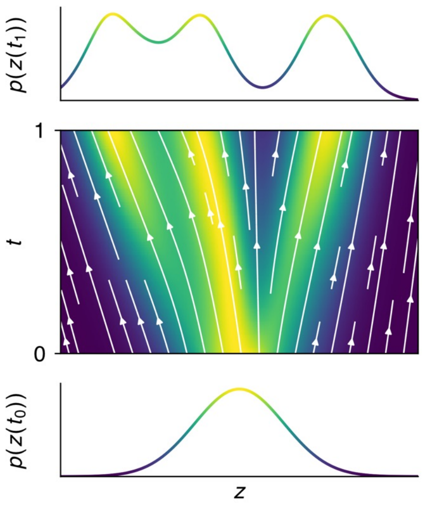
- f must be invertible
- Determinant computationally expensive
  - ➢ Restricted transformations needed

Continuous Normalizing Flow (CNF):

- f has no restrictions
- Trace is easier to calculate
- Still computationally expensive

Chen et al.; Neural Ordinary Differential Equations; arxiv:1806.07366

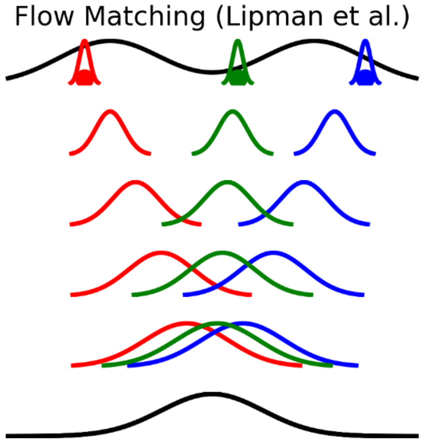
Flow Matching



[1810.01367]

$$\log p_1(x_1) = \cancel{\log p_0(x_0) - \int_{t_0}^{t} Tr\left(\frac{\partial v_\theta}{\partial x_t}\right) dt}$$

$$L_{FM} = \left|\left|v_\theta(x_t) - u_t(x_t|x_0)\right|\right|^2$$



$$x_t = \gamma_t x_0 + \sigma_t \epsilon$$

[2302.00482]

Continuous Normalizing Flow (CNF)

Training:
- Training is difficult because ODE needs to be solved

$$\frac{\partial x_t}{\partial t} = v_\theta(x_t, t)$$

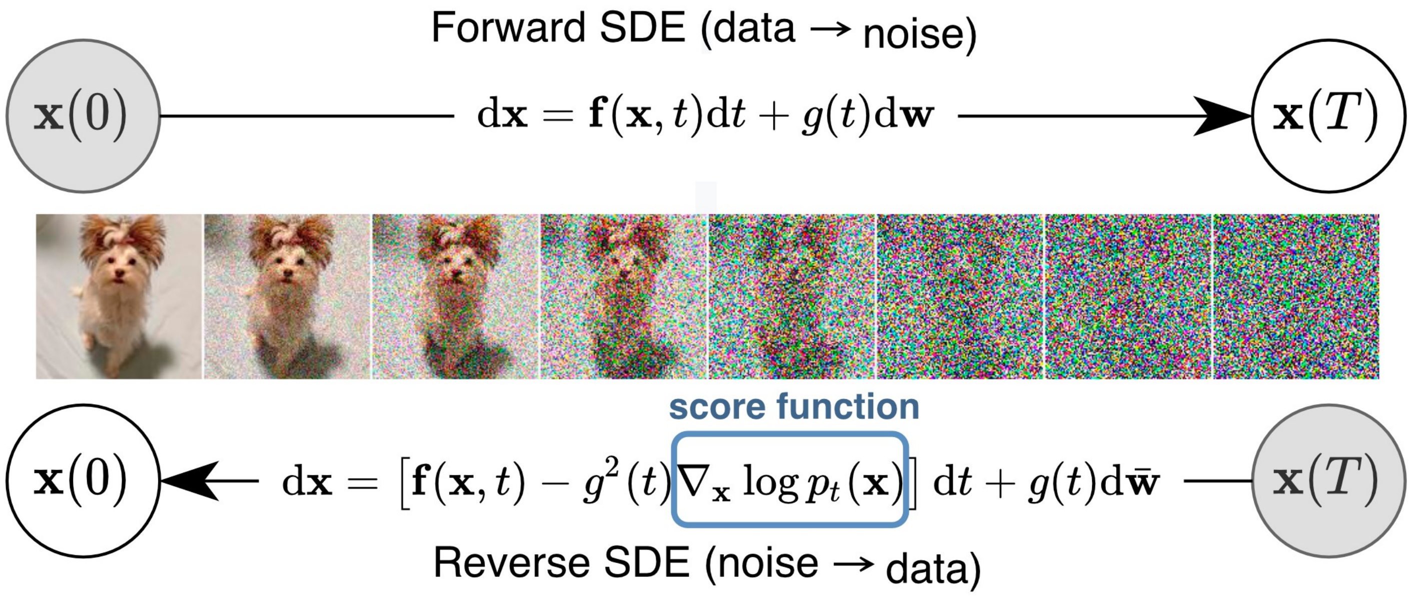
Flow Matching (FM)

Training:
- Simulation-free training objective (no ODE solving during training)
- Regressing against conditional flows
- ➢ Much faster training

Lipman et al.; Flow Matching for Generative Modeling; arxiv:2210.02747

Diffusion Models

- Adding noise to perturb data
- Description as stochastic differential equation (SDE)
- Sample by solving reverse SDE
- Train model by approximating score function with conditional probability paths



Probability Flow ODE:

- Remove stochasticity
- SDE → ODE

➢ A CNF describable with FM

"Continuous Time Generative Models"

$$L = ||s_\theta(x_t) - \nabla_x \log p_t(x|x_0)||$$

Loss Function

$$dx = \left[f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x)\right] dt$$

Probability flow ODE

Song et al.; Score-Based Generative Modeling through Stochastic Differential Equations; arxiv:2011.13456

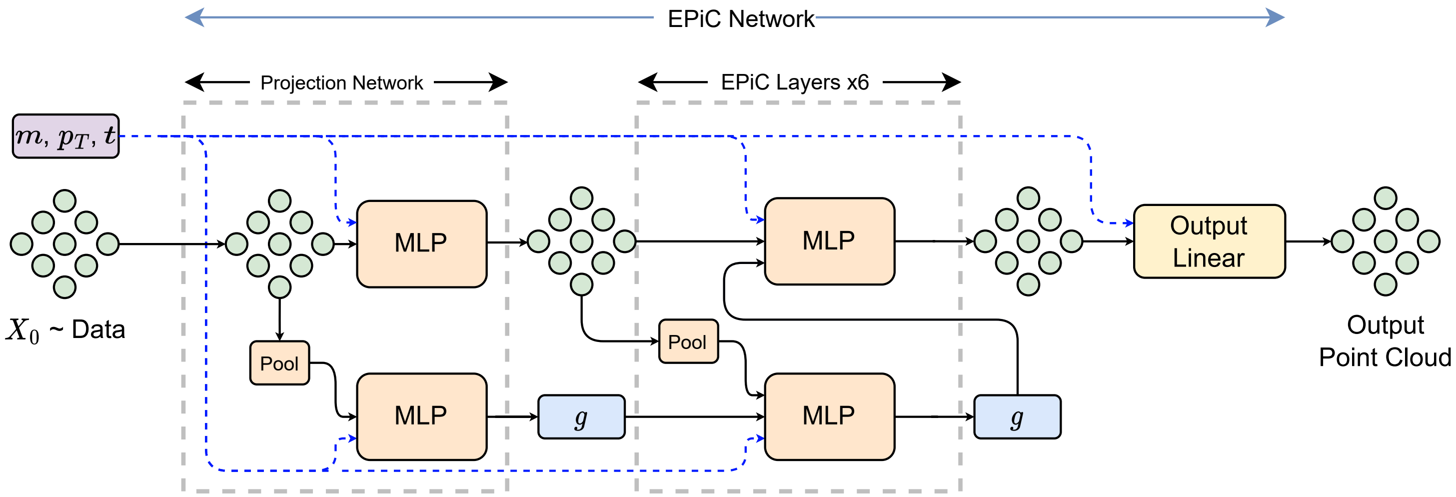
EPiC-FM & EPiC JeDi

- EPiC-FM: EPiC Architecture with Flow Matching

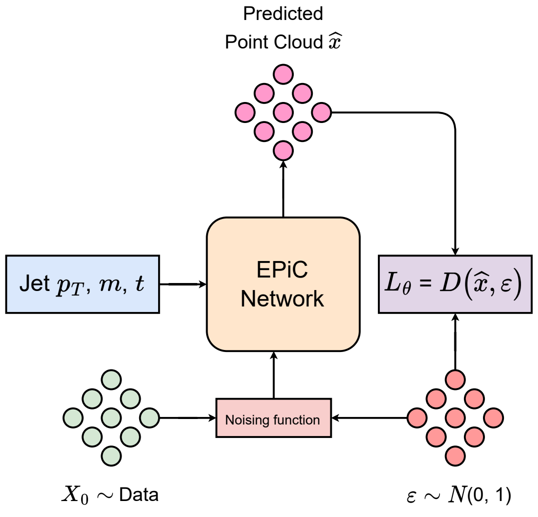
$$L_{FM}(v_\theta, u_t(x|x_0)) = \left|\left| v_\theta(x_t, t) - ((1-\sigma_{min})\epsilon - x_0) \right|\right|^2$$

- EPiC-JeDi: EPiC Architecture with JeDi diffusion

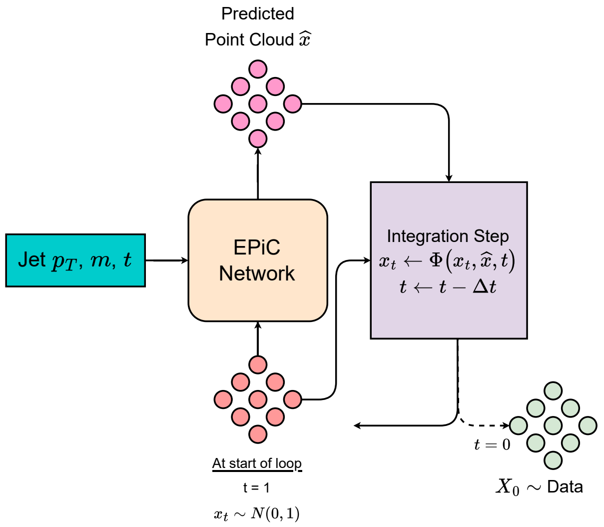
$$L_{JeDi}(v_\theta, s_t(x|x_0)) = \left(1 - \alpha \frac{\beta(t)}{\sigma(t)^2}\right) \left|\left| v_\theta(x_t, t) - \epsilon \right|\right|^2$$



EPiC Network



Training



Sampling

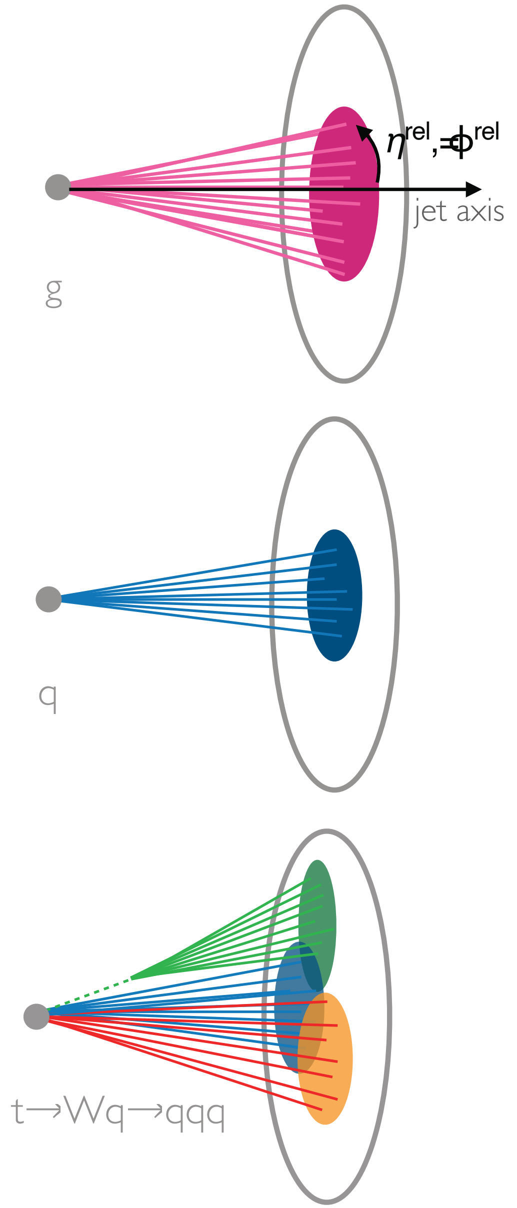
Benchmark on Pythia Jets

- Benchmark dataset: JetNet [2106.11535]
- Pythia simulated jets from proton-proton collisions
- Anti-k_T clustered with $R = 0.4$
- Maximum particle multiplicity of 30 and 150
- 5 jet classes (gluons, light-quarks, top quarks, W, Z)
- ~200k events per class
- Focus on top quarks (most challenging)
- Relative jet constituents

$$\eta^{rel} = \eta - \eta^{jet}$$

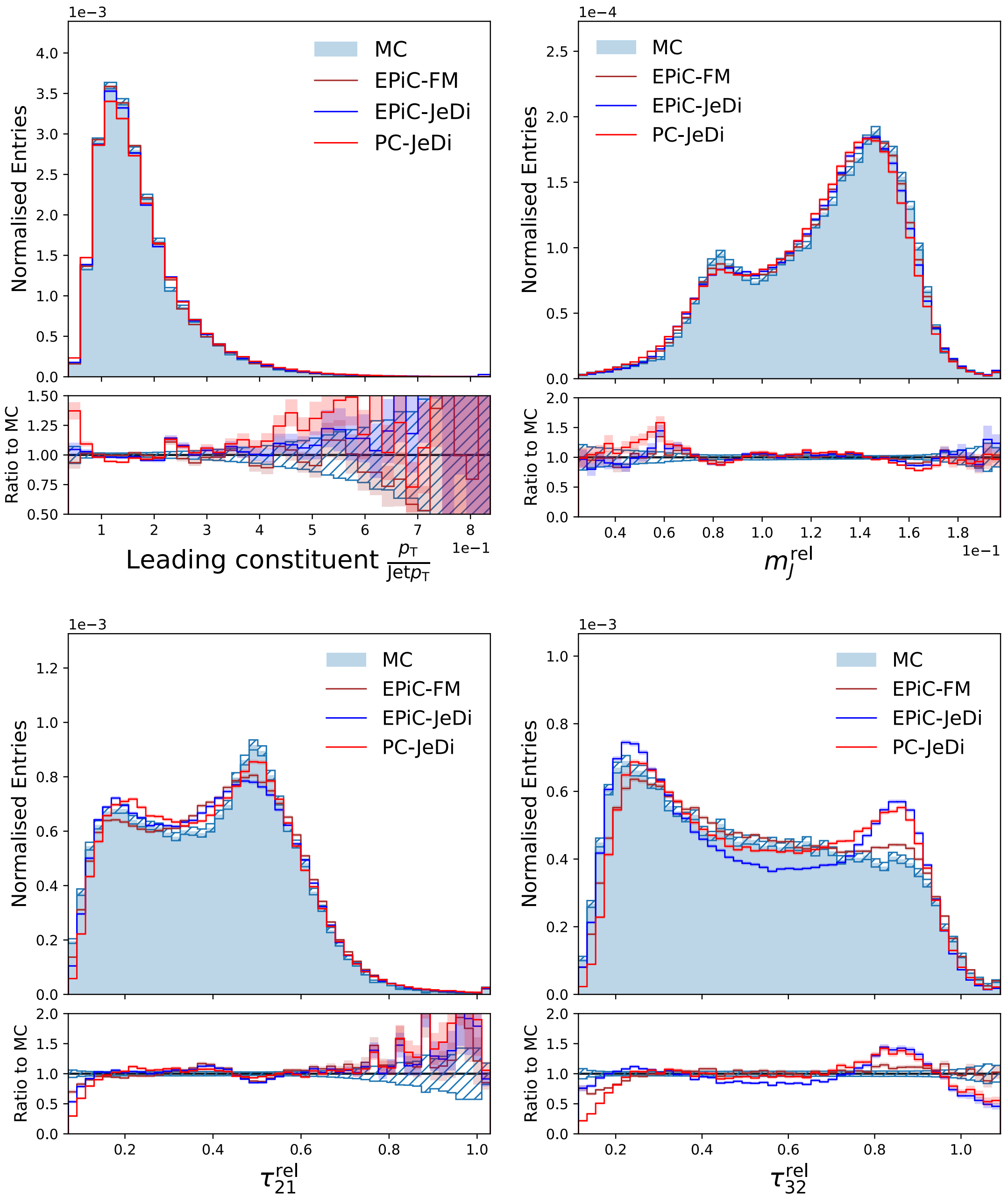
$$\phi^{rel} = \phi - \phi^{jet}$$

$$p_T^{rel} = p_T / p_T^{jet}$$



Results JetNet30 1/2

- Conditioned version (mass, p_T^{jet})
- Unconditioned version
- Generate conditioning with normalizing flow
- Comparison to EPiC GAN and PC-JeDi
- Midpoint ODE solver with 200 model passes
- Substructure most challenging to learn



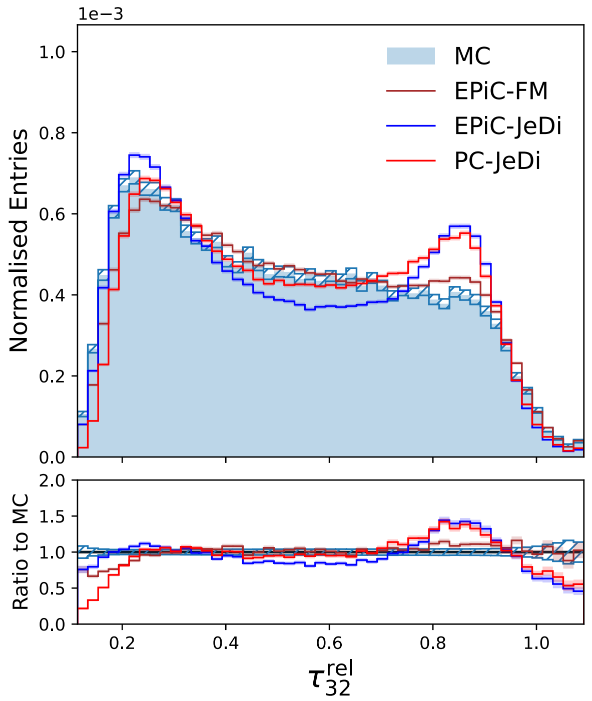
Results JetNet30 2/2

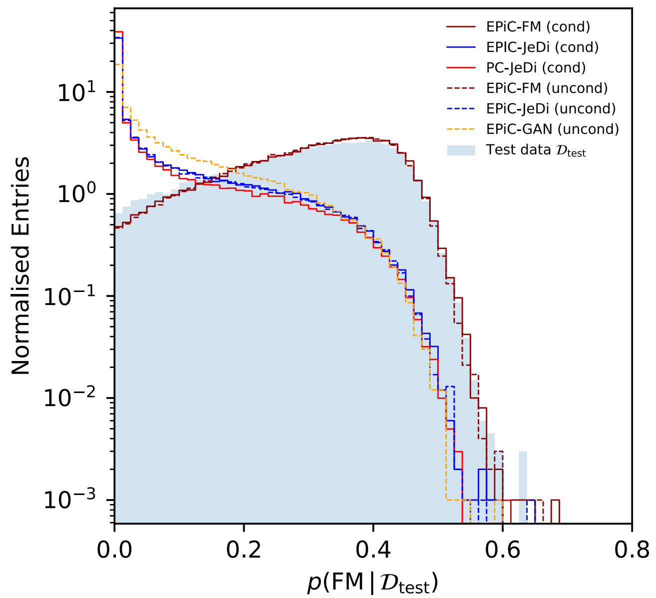
- KLD instead of Wasserstein distance
  - Unintuitive results for W1 on some distributions
- Multi-Classifier Score (ParticleNet)

$$NLP(c) = -E_{x \sim D_{Test}} \log p(c|x)$$

- ➢ EPiC-FM outperforms all models
- ➢ Conditioned models are slightly better
- ➢ Substructure most challenging to learn

Generation	Model	NLP	$\mathrm{KL}^{m}(\times 10^{-3})$	$\mathrm{KL}^{p_T^{\mathrm{const}}}(\times 10^{-3})$	$\mathrm{KL}^{\tau_{21}}(\times 10^{-3})$	$\mathrm{KL}^{\tau_{32}}(\times 10^{-3})$
Conditional	PC-JeDi	3.08	8.56 ± 0.75	3.25 ± 0.09	12.82 ± 1.16	27.08 ± 1.40
	EPiC-JeDi	3.1	5.26 ± 0.51	2.99 ± 0.05	$\mathbf{7.81 \pm 0.61}$	17.34 ± 1.08
	EPiC-FM	**1.35**	$\mathbf{3.77 \pm 0.50}$	$\mathbf{2.03 \pm 0.02}$	$\mathbf{7.40 \pm 0.64}$	$\mathbf{8.09 \pm 0.93}$
Unconditional	EPiC-GAN	3.43	$\mathbf{3.71 \pm 0.42}$	3.33 ± 0.03	$\mathbf{8.28 \pm 0.76}$	17.68 ± 0.91
	EPiC-JeDi	3.11	18.42 ± 1.12	3.73 ± 0.08	$\mathbf{8.00 \pm 0.80}$	15.27 ± 1.35
	EPiC-FM	1.38	5.80 ± 0.54	$\mathbf{2.03 \pm 0.01}$	$\mathbf{7.69 \pm 0.71}$	$\mathbf{9.24 \pm 1.00}$

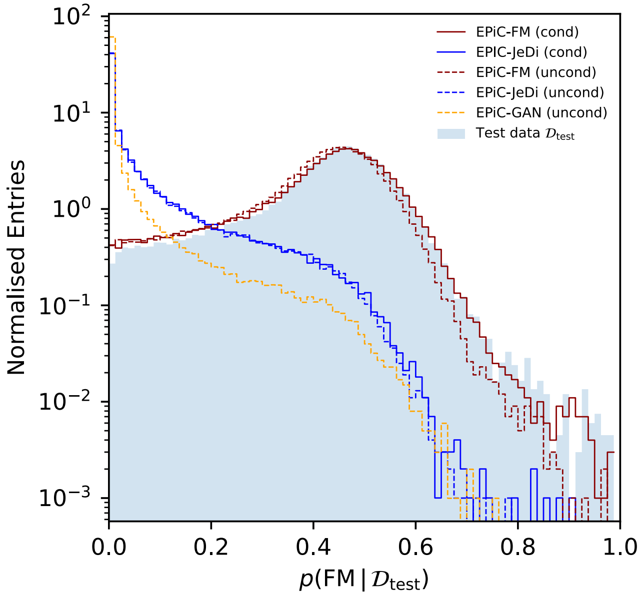


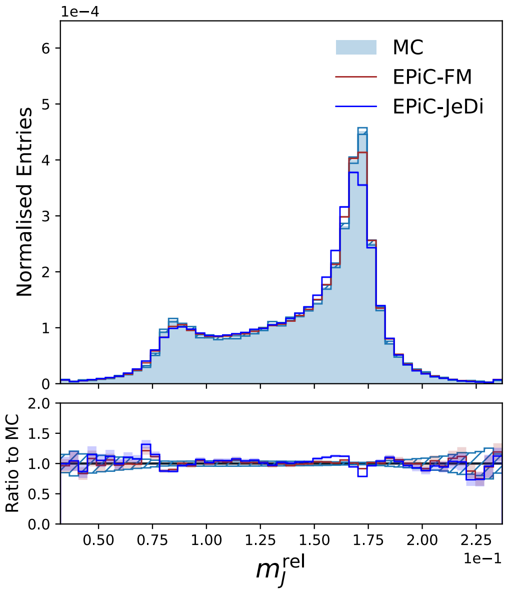


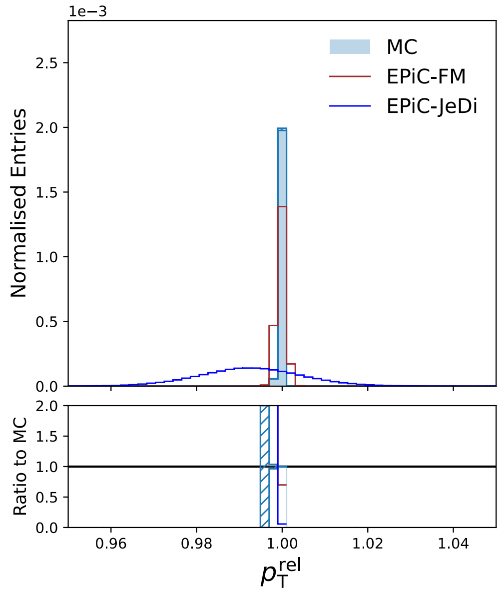
Results JetNet150

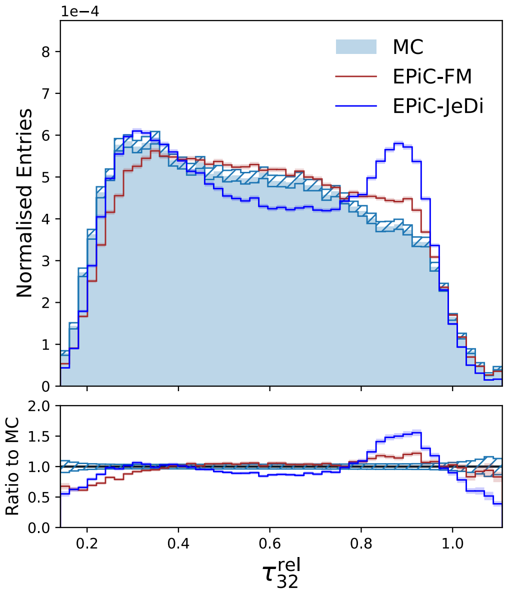
- Conditioned on mass and p_T vs. unconditioned
- Comparison to EPiC GAN
- PC-JeDi too slow for 150 particles

➢ Similar behaviour as for 30 particles
➢ EPiC-FM outperforms all models
➢ Conditioned models are slightly better







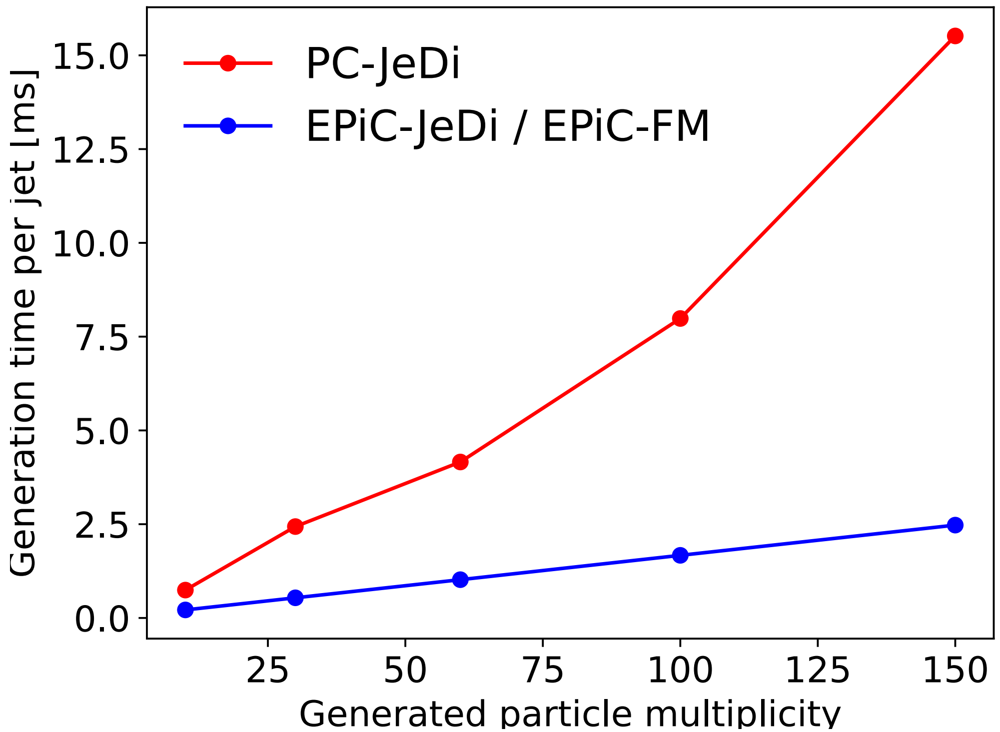


Generation	Model	NLP	$\text{KL}^{m}(\times 10^{-3})$	$\text{KL}^{p_T^{\text{const}}}(\times 10^{-3})$	$\text{KL}^{\tau_{21}}(\times 10^{-3})$	$\text{KL}^{\tau_{32}}(\times 10^{-3})$
Conditional	EPiC-JeDi	5.67	9.10 ± 0.79	6.42 ± 0.76	14.32 ± 1.08	19.92 ± 1.21
	EPiC-FM	**0.12**	$\mathbf{4.30 \pm 0.53}$	$\mathbf{0.84 \pm 0.02}$	$\mathbf{9.43 \pm 0.61}$	$\mathbf{11.22 \pm 1.02}$
Unconditional	EPiC-GAN	11.6	$\mathbf{6.50 \pm 0.63}$	2.22 ± 0.09	20.60 ± 1.55	69.64 ± 3.30
	EPiC-JeDi	5.70	27.46 ± 1.24	6.39 ± 0.60	20.15 ± 1.25	36.50 ± 1.81
	EPiC-FM	0.98	12.95 ± 0.90	$\mathbf{0.87 \pm 0.02}$	$\mathbf{10.59 \pm 0.88}$	$\mathbf{12.14 \pm 0.97}$

Timing Evaluation

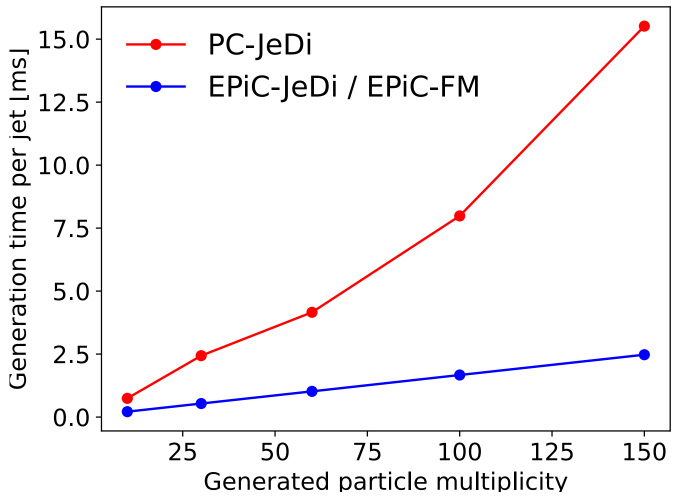
- Better scaling behaviour for EPiC layers
  - 6.2x faster at 150 particles
- Effect increases for larger point clouds like calorimeter showers
- Slower than GANs

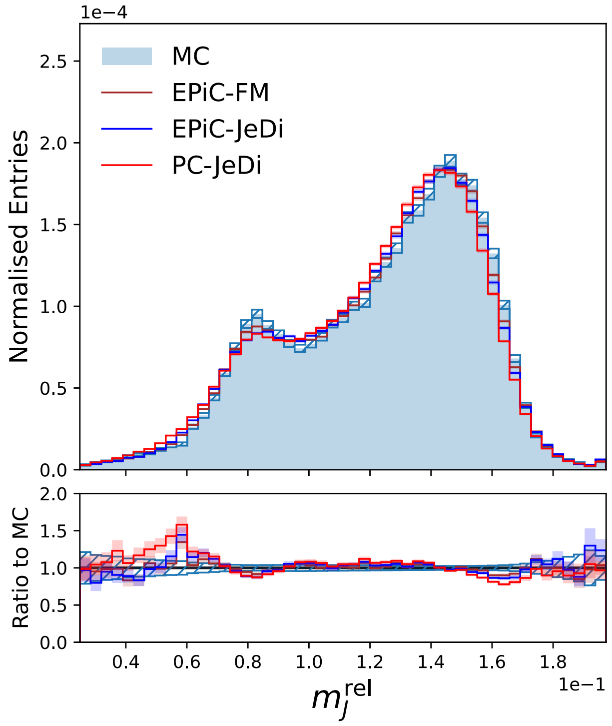
➢ Complementary to distillation approaches
  - See PC-Droid Talk [2307.06836]



Conclusion

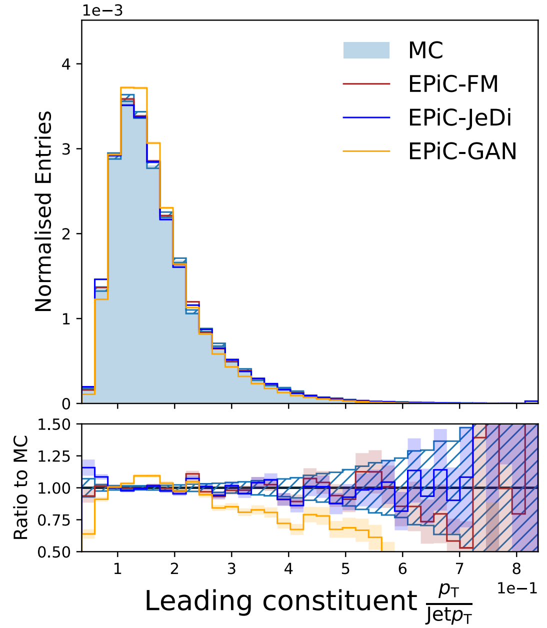
- Generative ML can speed up MC simulations
- Points Clouds are the natural way to represent jets
- We introduce two new models
  - EPiC-FM
  - EPiC-JeDi
- Significantly better scaling behaviour while keeping performance of previous transformer approaches
- EPiC-FM is simpler and performs better than previous diffusion-based models
- Paper on [arxiv: 2310.00049](https://arxiv.org/abs/2310.00049)

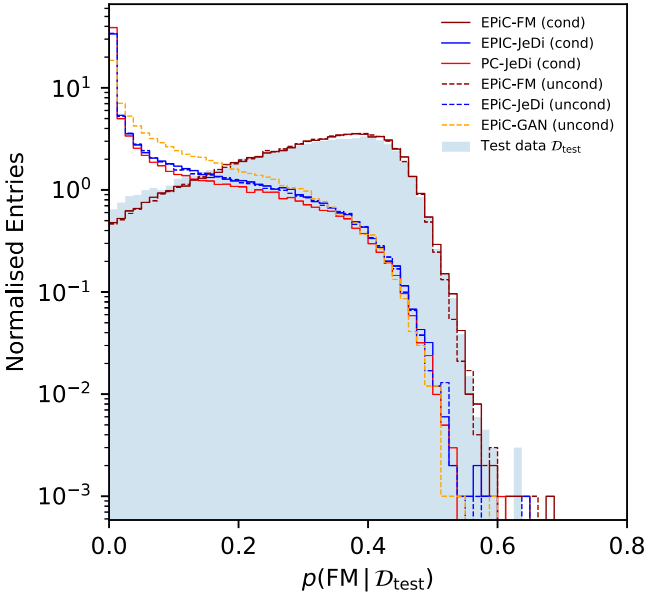


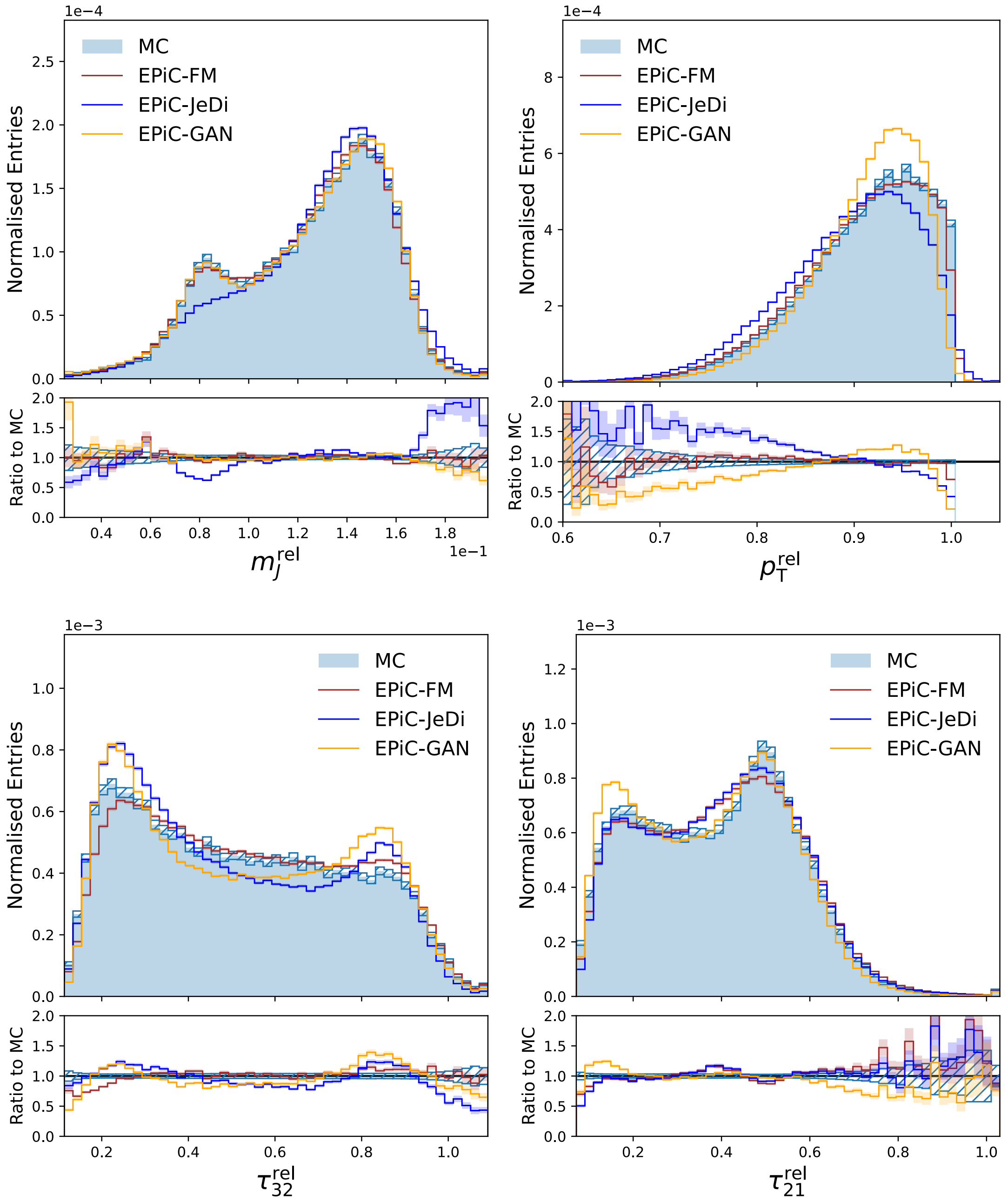


Additional Slides

JetNet30 unconditional Plots

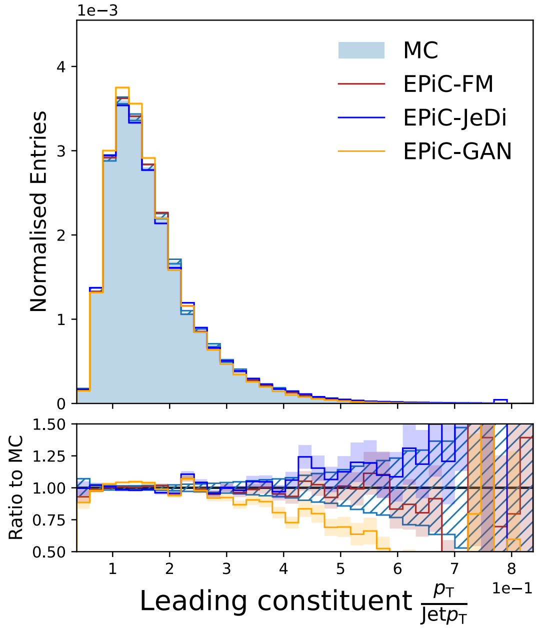


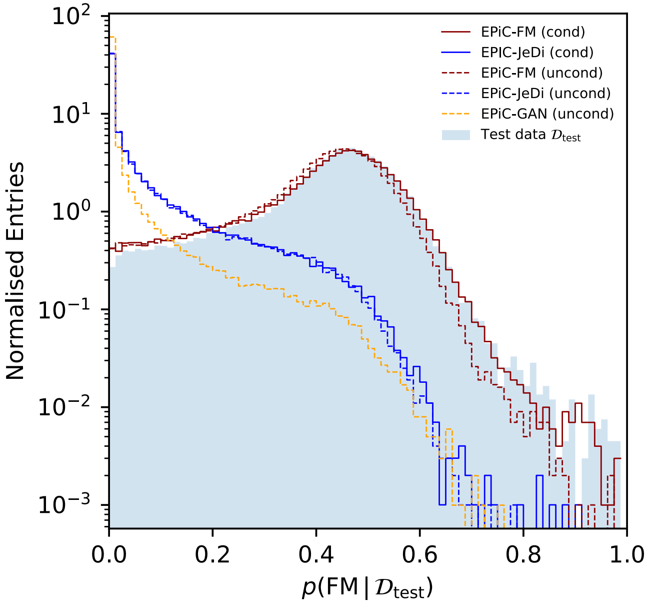


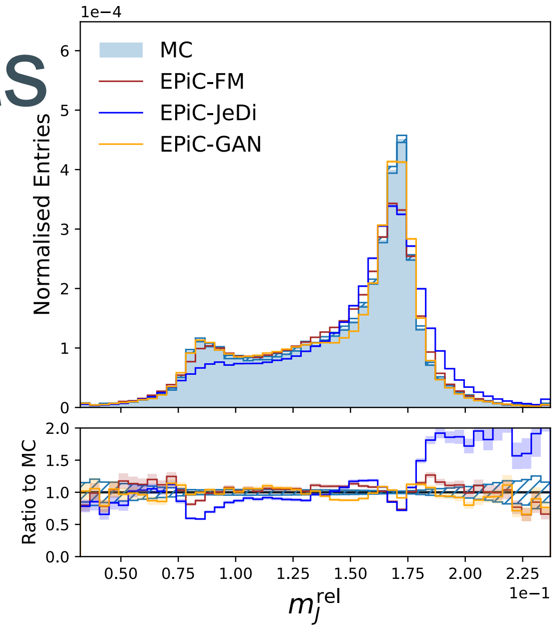


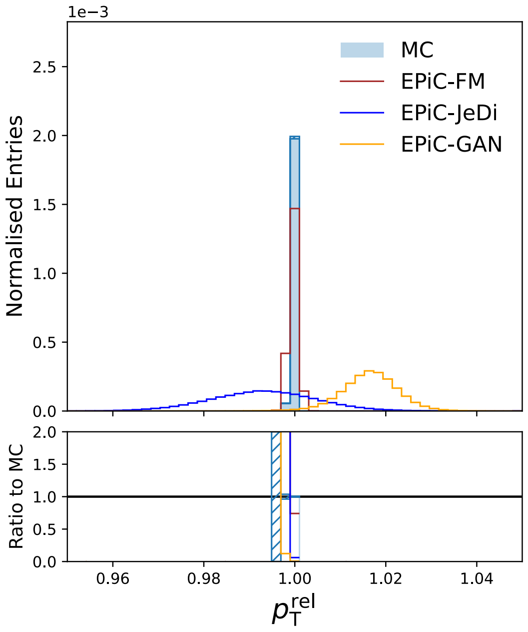
Generation	Model	NLP	$\text{KL}^{m}(\times 10^{-3})$	$\text{KL}^{p_T^{\text{const}}}(\times 10^{-3})$	$\text{KL}^{\tau_{21}}(\times 10^{-3})$	$\text{KL}^{\tau_{32}}(\times 10^{-3})$
Conditional	PC-JeDi	3.08	8.56 ± 0.75	3.25 ± 0.09	12.82 ± 1.16	27.08 ± 1.40
	EPiC-JeDi	3.1	5.26 ± 0.51	2.99 ± 0.05	$\mathbf{7.81 \pm 0.61}$	17.34 ± 1.08
	EPiC-FM	**1.35**	$\mathbf{3.77 \pm 0.50}$	$\mathbf{2.03 \pm 0.02}$	7.40 ± 0.64	$\mathbf{8.09 \pm 0.93}$
Unconditional	EPiC-GAN	3.43	$\mathbf{3.71 \pm 0.42}$	3.33 ± 0.03	$\mathbf{8.28 \pm 0.76}$	17.68 ± 0.91
	EPiC-JeDi	3.11	18.42 ± 1.12	3.73 ± 0.08	$\mathbf{8.00 \pm 0.80}$	15.27 ± 1.35
	EPiC-FM	1.38	5.80 ± 0.54	$\mathbf{2.03 \pm 0.01}$	7.69 ± 0.71	$\mathbf{9.24 \pm 1.00}$

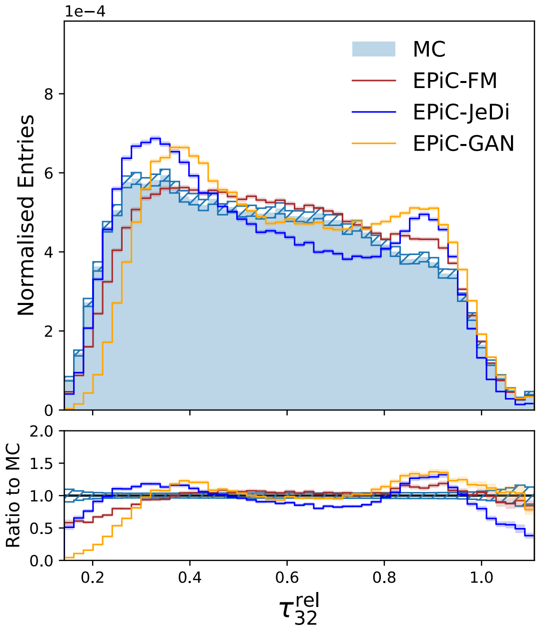
JetNet150 unconditional Plots

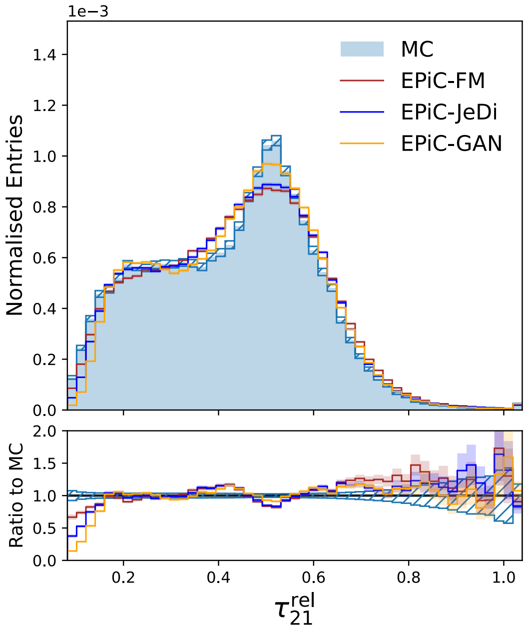












Generation	Model	NLP	KL^{m}($\times10^{-3}$)	KL$^{p_T^{\text{const}}}$($\times10^{-3}$)	KL$^{\tau_{21}}$($\times10^{-3}$)	KL$^{\tau_{32}}$($\times10^{-3}$)
Conditional	EPiC-JeDi	5.67	9.10 ± 0.79	6.42 ± 0.76	14.32 ± 1.08	19.92 ± 1.21
	EPiC-FM	**0.12**	$\mathbf{4.30 \pm 0.53}$	$\mathbf{0.84 \pm 0.02}$	$\mathbf{9.43 \pm 0.61}$	$\mathbf{11.22 \pm 1.02}$
Unconditional	EPiC-GAN	11.6	$\mathbf{6.50 \pm 0.63}$	2.22 ± 0.09	20.60 ± 1.55	69.64 ± 3.30
	EPiC-JeDi	5.70	27.46 ± 1.24	6.39 ± 0.60	20.15 ± 1.25	36.50 ± 1.81
	EPiC-FM	0.98	12.95 ± 0.90	$\mathbf{0.87 \pm 0.02}$	$\mathbf{10.59 \pm 0.88}$	$\mathbf{12.14 \pm 0.97}$

Solver Comparison

Generation	Model	Sampler	**FPND**	$W_1^{m}(\times 10^{-4})$	$W_1^{p_T}(\times 10^{-4})$	$W_1^{EFP}(\times 10^{-5})$	$W_1^{\tau_{21}}(\times 10^{-3})$	$W_1^{\tau_{32}}(\times 10^{-3})$	$W_1^{D_2}(\times 10^{-3})$
Conditional	EPiC-JeDi	EM (SDE)	0.29	16.96 ± 2.00	5.32 ± 1.10	3.47 ± 0.38	7.84 ± 0.77	26.36 ± 1.41	**0.81 ± 0.07**
		Midpoint	0.42	8.29 ± 1.20	14.67 ± 1.38	1.76 ± 0.22	**5.09 ± 0.43**	14.19 ± 0.83	1.35 ± 0.22
		Euler	0.39	8.65 ± 1.14	14.65 ± 1.68	1.79 ± 0.25	5.60 ± 0.46	**13.83 ± 1.11**	1.37 ± 0.17
	EPiC-FM	Midpoint	**0.11**	**5.12 ± 1.18**	**3.36 ± 0.98**	**1.10 ± 0.26**	7.54 ± 0.84	16.33 ± 1.21	0.97 ± 0.17
		Euler	0.19	13.26 ± 1.85	10.95 ± 1.40	3.11 ± 0.35	10.54 ± 1.12	18.72 ± 1.36	1.13 ± 0.11
Unconditional	EPiC-JeDi	EM (SDE)	0.77	16.92 ± 1.36	14.52 ± 1.73	2.88 ± 0.20	12.62 ± 0.82	**12.09 ± 0.75**	2.19 ± 0.18
		Midpoint	1.63	37.54 ± 1.91	33.57 ± 1.48	8.08 ± 0.40	**7.71 ± 0.99**	15.73 ± 1.17	3.69 ± 0.19
		Euler	1.64	37.10 ± 1.72	32.63 ± 1.59	8.33 ± 0.44	8.56 ± 0.87	14.29 ± 0.86	3.86 ± 0.18
	EPiC-FM	Midpoint	**0.14**	**7.69 ± 0.97**	**3.39 ± 0.98**	**1.45 ± 0.30**	7.77 ± 0.80	14.97 ± 1.39	**0.94 ± 0.17**
		Euler	0.39	30.16 ± 1.78	17.55 ± 1.49	6.43 ± 0.42	8.41 ± 0.72	23.53 ± 1.37	1.40 ± 0.10

JetNet30, all solvers with 200 model passes

Generation	Model	Sampler	**FPND**	$W_1^{m}(\times 10^{-4})$	$W_1^{p_T}(\times 10^{-4})$	$W_1^{EFP}(\times 10^{-5})$	$W_1^{\tau_{21}}(\times 10^{-3})$	$W_1^{\tau_{32}}(\times 10^{-3})$	$W_1^{D_2}(\times 10^{-3})$
Conditional	EPiC-JeDi	EM (SDE)	0.26	10.12 ± 2.05	6.46 ± 0.78	5.77 ± 0.81	7.60 ± 0.42	31.34 ± 1.52	1.97 ± 0.23
		Midpoint	0.52	6.61 ± 1.05	18.89 ± 1.25	4.78 ± 0.62	**7.51 ± 0.43**	21.15 ± 1.25	3.13 ± 0.23
		Euler	0.47	6.77 ± 1.55	18.80 ± 1.29	4.97 ± 0.71	8.73 ± 0.58	21.77 ± 1.29	3.39 ± 0.16
	EPiC-FM	Midpoint	**0.12**	**3.74 ± 0.89**	**3.14 ± 1.07**	**2.30 ± 0.42**	8.51 ± 0.98	**20.67 ± 1.33**	1.47 ± 0.19
		Euler	0.15	4.08 ± 0.88	14.24 ± 1.18	2.38 ± 0.49	8.92 ± 0.87	22.54 ± 1.04	**0.65 ± 0.12**
Unconditional	EPiC-JeDi	EM (SDE)	0.52	31.37 ± 2.53	8.46 ± 1.29	13.79 ± 0.91	8.82 ± 0.62	21.56 ± 1.65	3.30 ± 0.19
		Midpoint	1.93	66.07 ± 2.05	35.04 ± 1.51	27.84 ± 0.86	**8.75 ± 0.97**	11.67 ± 0.60	6.24 ± 0.26
		Euler	1.90	66.85 ± 2.14	35.67 ± 1.53	28.03 ± 0.97	9.90 ± 0.89	**11.40 ± 0.82**	6.30 ± 0.19
	EPiC-FM	Midpoint	**0.18**	**10.77 ± 1.12**	**3.25 ± 0.89**	**4.03 ± 0.37**	9.37 ± 0.74	19.85 ± 1.29	**1.11 ± 0.18**
		Euler	0.47	31.86 ± 2.05	21.79 ± 1.45	10.66 ± 0.78	9.65 ± 0.91	28.16 ± 1.43	1.52 ± 0.15

JetNet150, all solvers with 200 model passes

Hyperparameter Choices

Hyperparameter	Value
EPiC layers	6
EPiC global dimensionality	10
Hidden dimensionality	128
Activation function	LeakyReLU(0.01)
Adam-W [91] learning rate	10^{-3}
Learning rate scheduling	Cosine with warm-up
Warm-up epochs	1,000
Batch size	1,024
Training epochs	10,000
Model weights	$\sim 560,000$
Training events	$\sim 110,000$
Test events	$\sim 27,000$