# Full Phase Space Resonant Anomaly Detection

Erik Buhmann, **Cedric Ewen***, Gregor Kasieczka,
Vinicius Mikuni, Benjamin Nachman, and David Shih

08/11/2023 - ML4Jets 2023

arxiv: 2310.06879

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

# Motivation

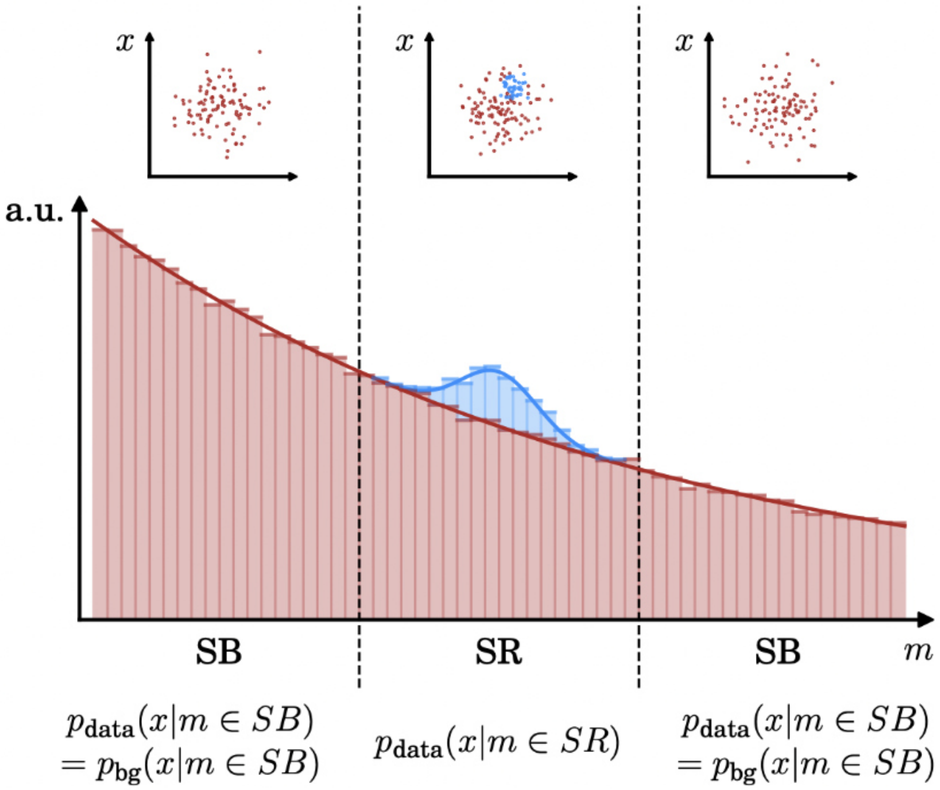
- Existence of physics beyond the standard model is likely
- Too many models for dedicated searches

➢ Need for data-driven model-independent searches

➢ Anomaly detection with ML

Resonant Anomaly detection:

- Feature $m$ with smooth background
- Signal localized in $m$
- Use feature set $x$ to enhance anomaly




[2109.00546]

- Choice of feature set $x$ is difficult
- Previous enhancement to use more features

➢ We want to use all available features

# CATHODE

- Goal: Approximate likelihood-ratio $\rho_{sig+bg}/\rho_{bg}$
- Train conditioned generative model on SB background
- Interpolate into SR and sample background-like events
- Compare generated background and data with a classifier

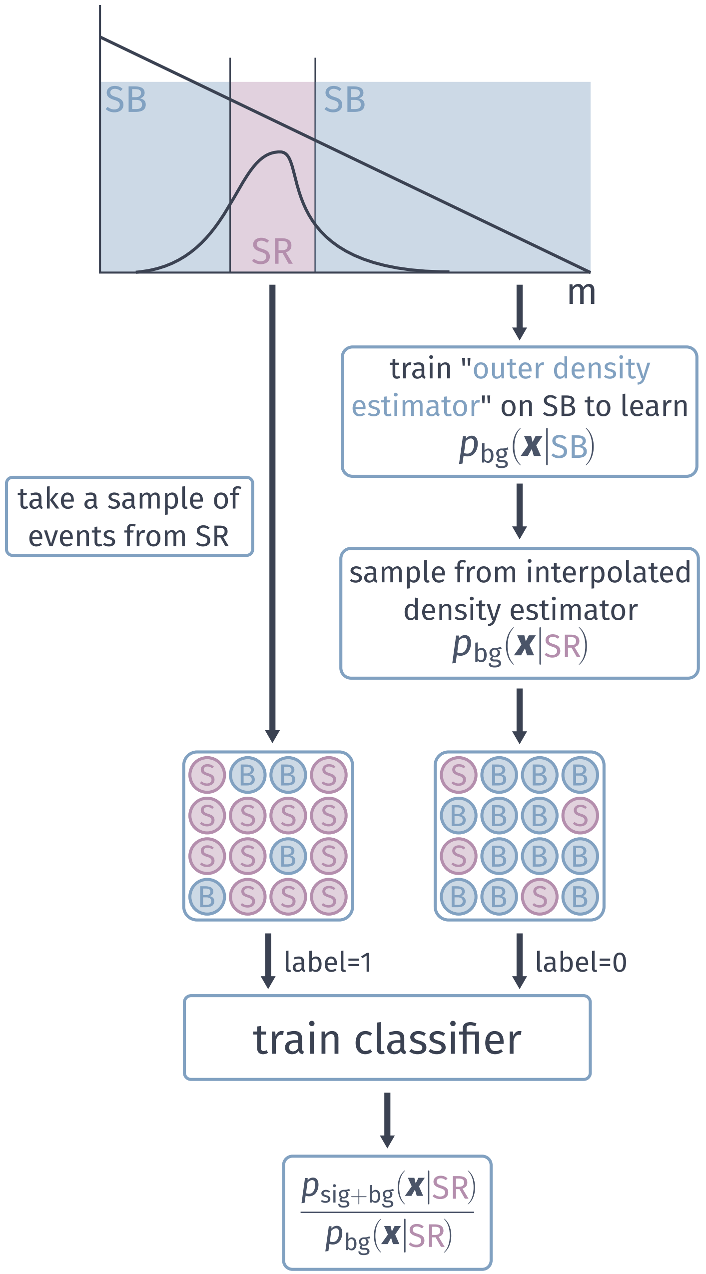


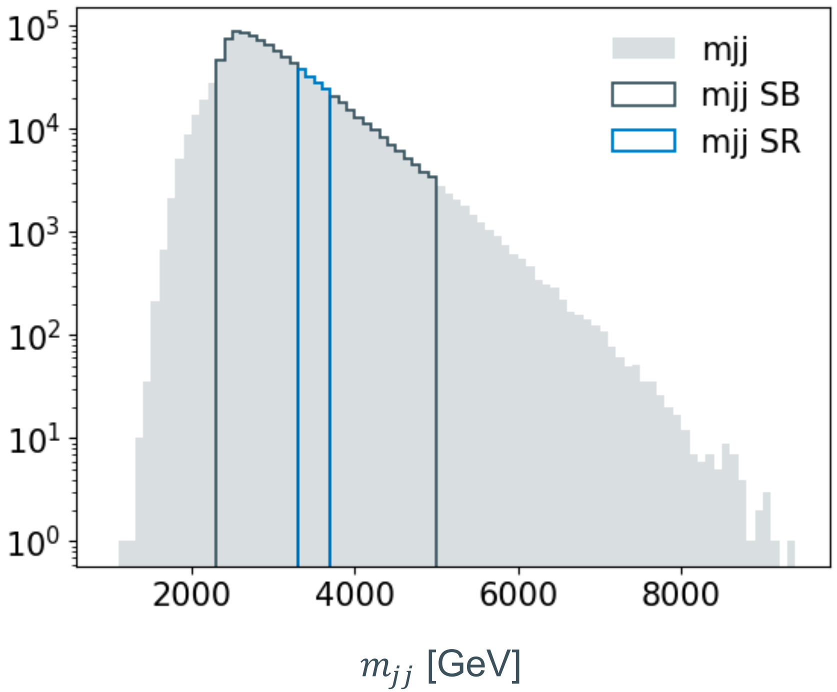
Figure by Tobias Quadfasel

Hallin et al.; Classifying Anomalies THrough Outer Density Estimation (CATHODE); arxiv: 2109.00546

# Dataset

- LHC Olympics 2020 challenge R&D dataset
- Background: QCD
- Signal: $W' \rightarrow XY$ with $X \rightarrow qq, Y \rightarrow qq$
- $m_W = 3.5$ TeV, $m_X = 500$ GeV, $m_Y = 200$ GeV
- Resonant feature: dijet mass $m_{jj}$
- SR: $3300$ GeV - $3700$ GeV
- SB: $2300$ GeV - $3300$ GeV and $3700$ GeV - $5000$ GeV

- Two leading $p_T$ jets selected
- Up to 279 constituents per jet with $p_T, \eta, \phi$




https://lhco2020.github.io/homepage/

# Full Phase Space Resonant Anomaly Detection

## Original Cathode

$m_{j1}, \Delta m, \tau_{21,j1}, \tau_{21,j2}$

4 features

- Discriminative features need to be selected
- Features must contain the anomaly

➢ Simple ML task

## Full Phase Space

2 jets * 279 constituents * 3 features

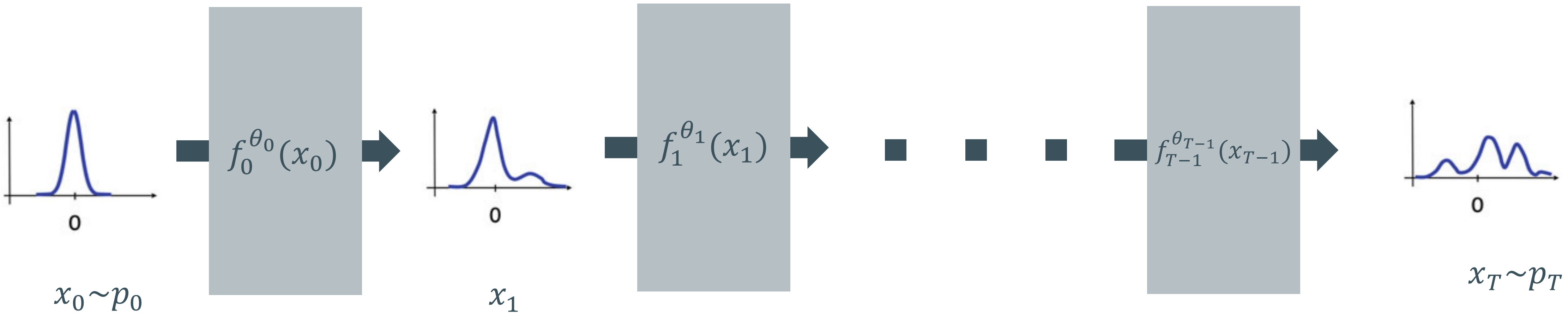
up to 1674 features

- Weak supervision is difficult
- Jets represented as point clouds
  - Permutation invariant
  - Variable jet sizes

➢ Powerful networks needed

| | | |
|---|---|---|
| Normalizing Flow (MAF) | Generative Network | Diffusion / Flow Matching model with DeepSets/Transformer |
| MLP classifier | Classifier | Transformer Point Cloud classifier |

# Normalizing Flows




**Normalizing Flow (NF)**

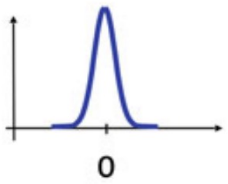
Training: $$\log p_T(x_T) = \log p_0(x_0) - \log \left| \frac{\partial f_t^{\theta}}{\partial x_t} \right|$$

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

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

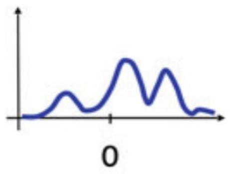
Rezende et al.; Variational Inference with Normalizing Flows; arxiv:1505.05770

# 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)**

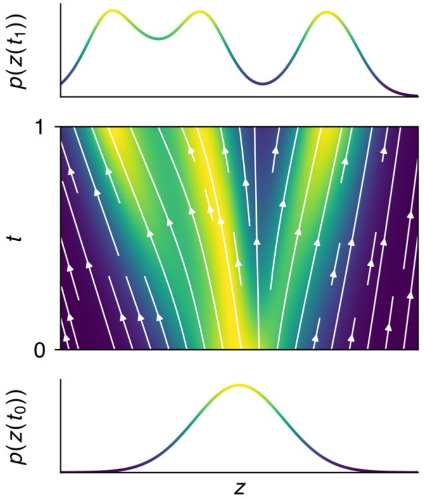
- $f$ must be invertible
- Determinant computationally expensive
  - ➢ Restricted transformations needed

**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

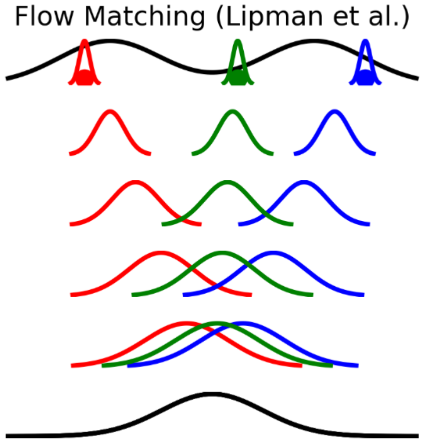
# Flow Matching




[1810.01367]

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

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




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

[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:

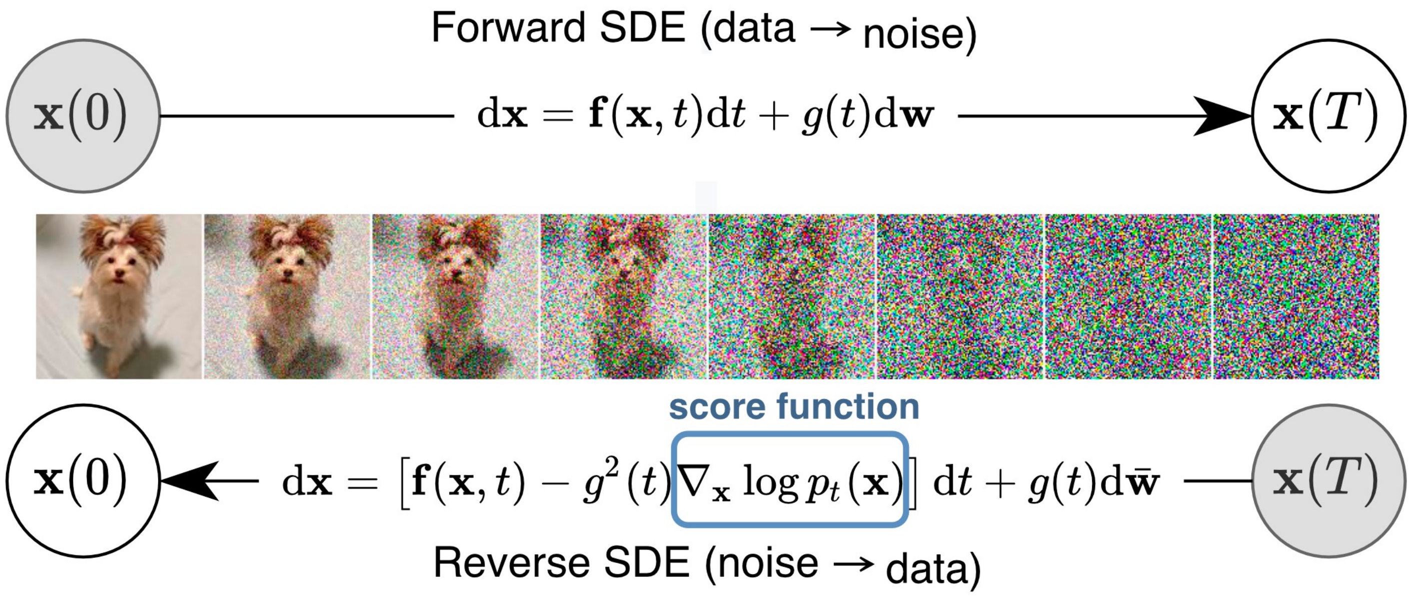
- Simulation-free training objective (no ODE solving during training)
- Regressing against conditional flows

➢ 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

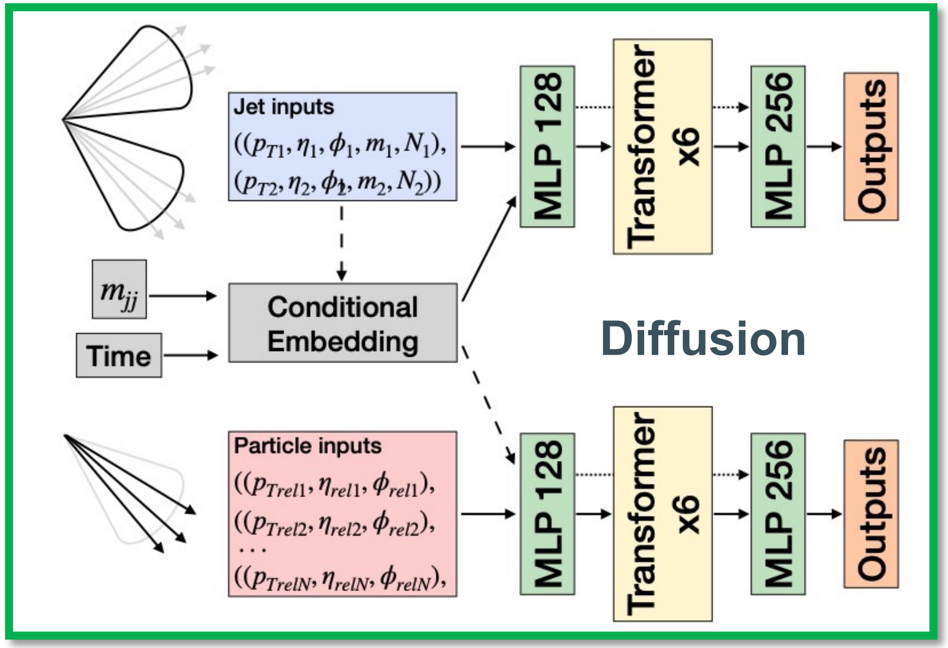
# Architecture

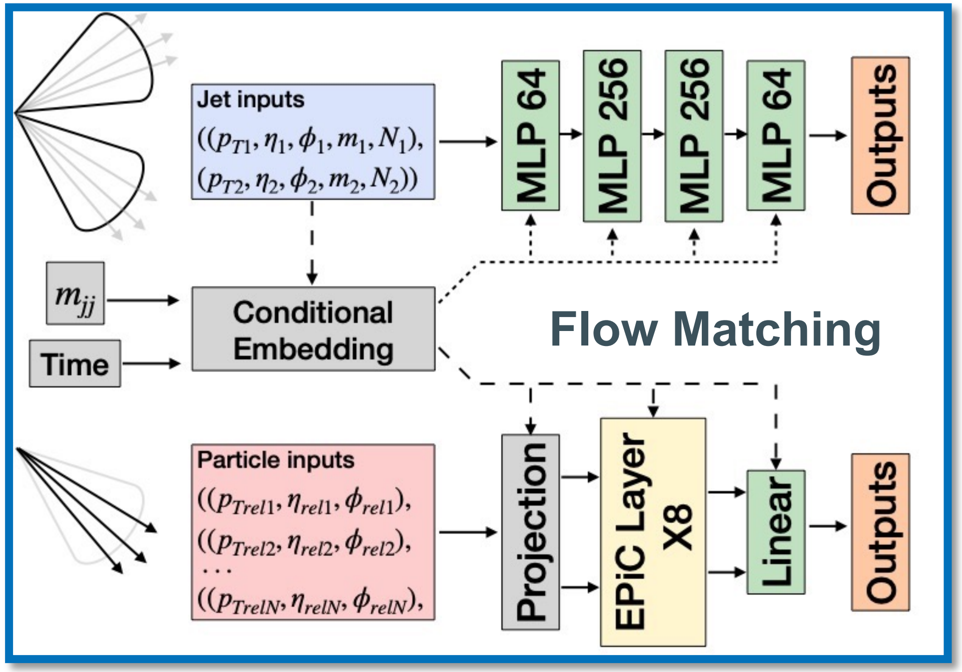
Generation Pipeline:

- $m_{jj}$-model (KDE)
- Jet feature model
  - Conditioned on $m_{jj}$
- Particle feature model
  - Conditioned on jet features

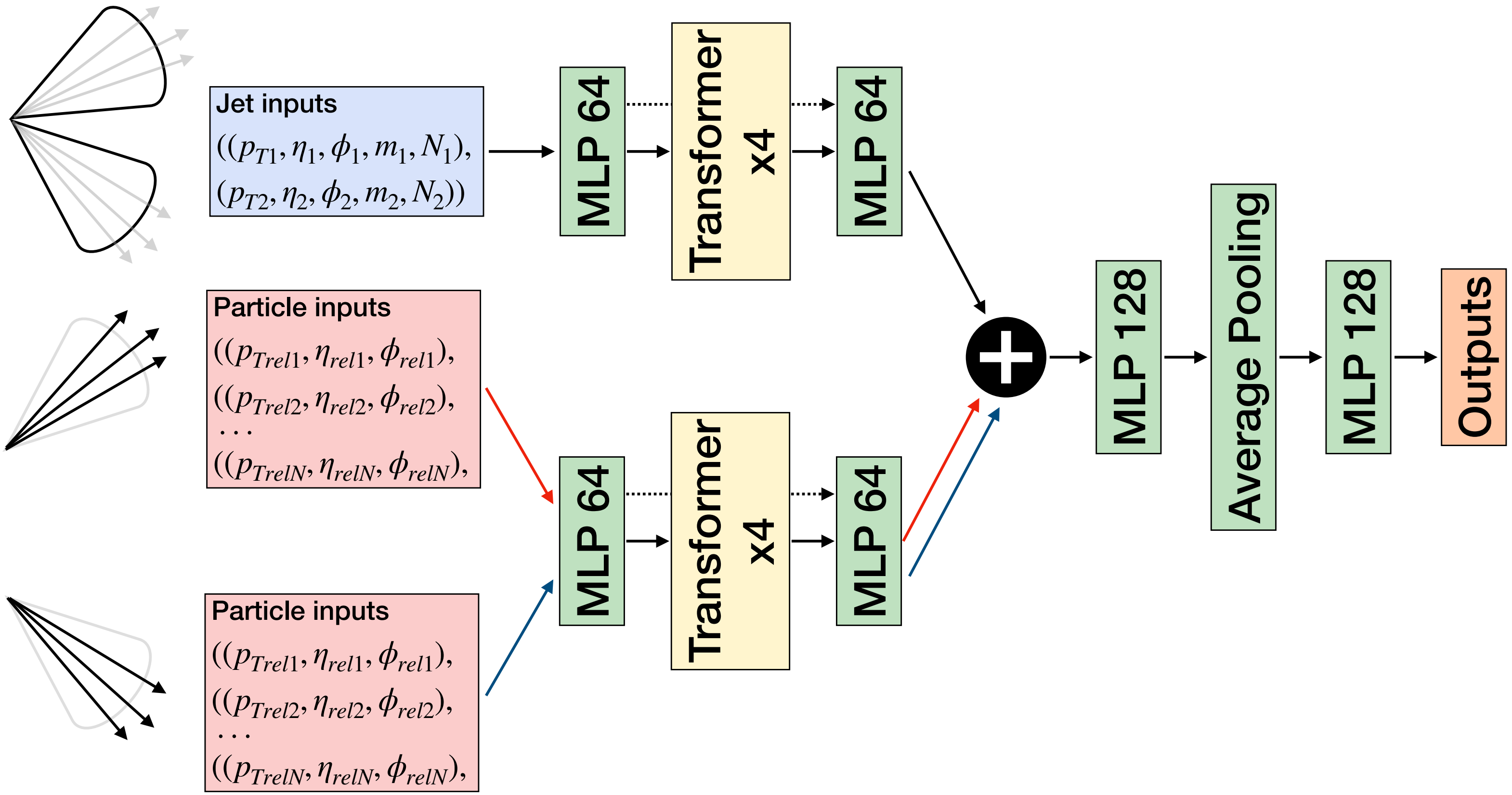
Two approaches:

- Diffusion + Transformer [2304.01266]
- Flow Matching + MLP/ EPiC [2310.00049]
  - EPiC: DeepSets based



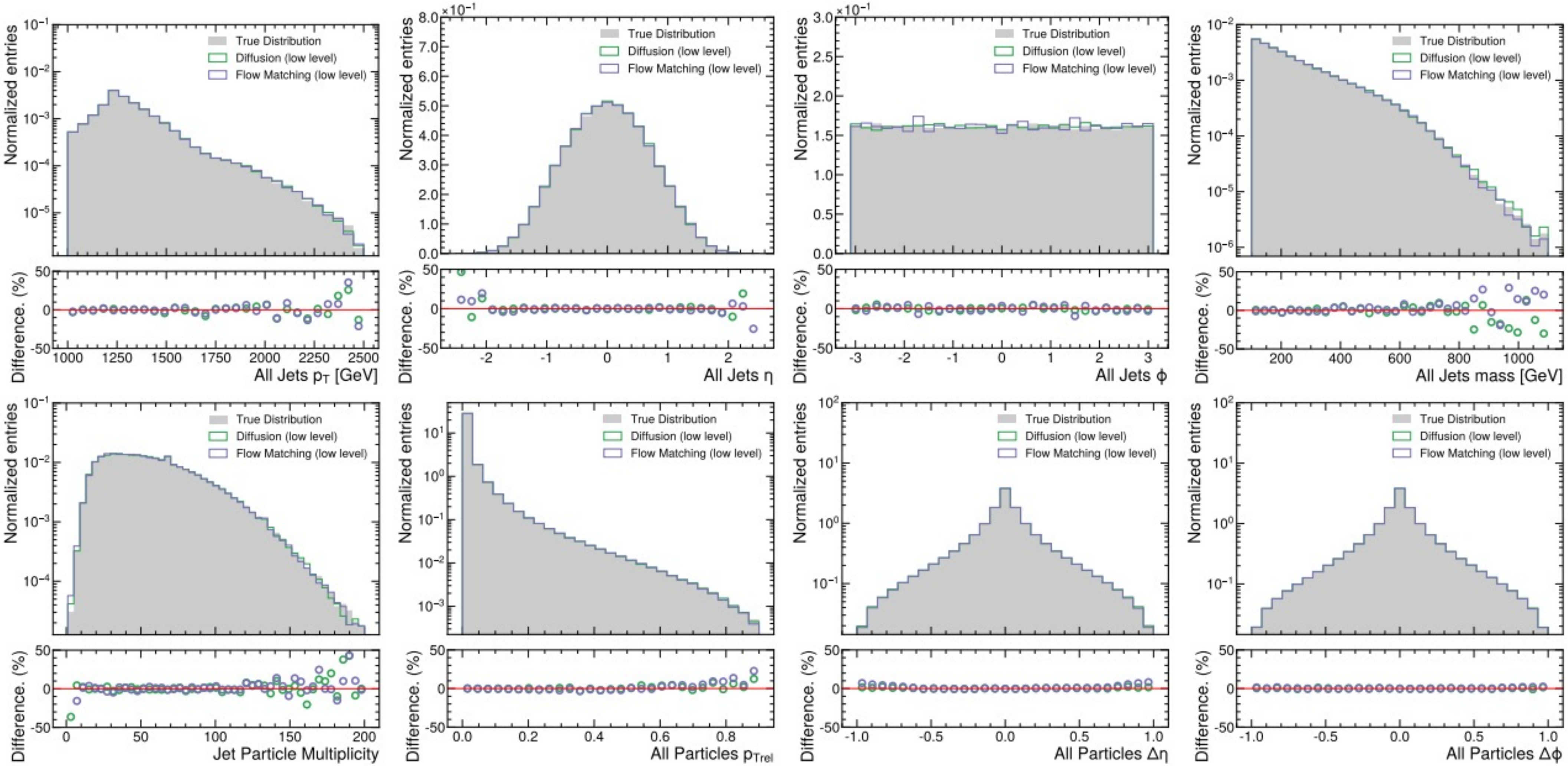
# Classifier




- Point Cloud Classifier
- Input: Particle Features/ Jet Features
- DeepSets/ Transformer architecture
- Equivariant

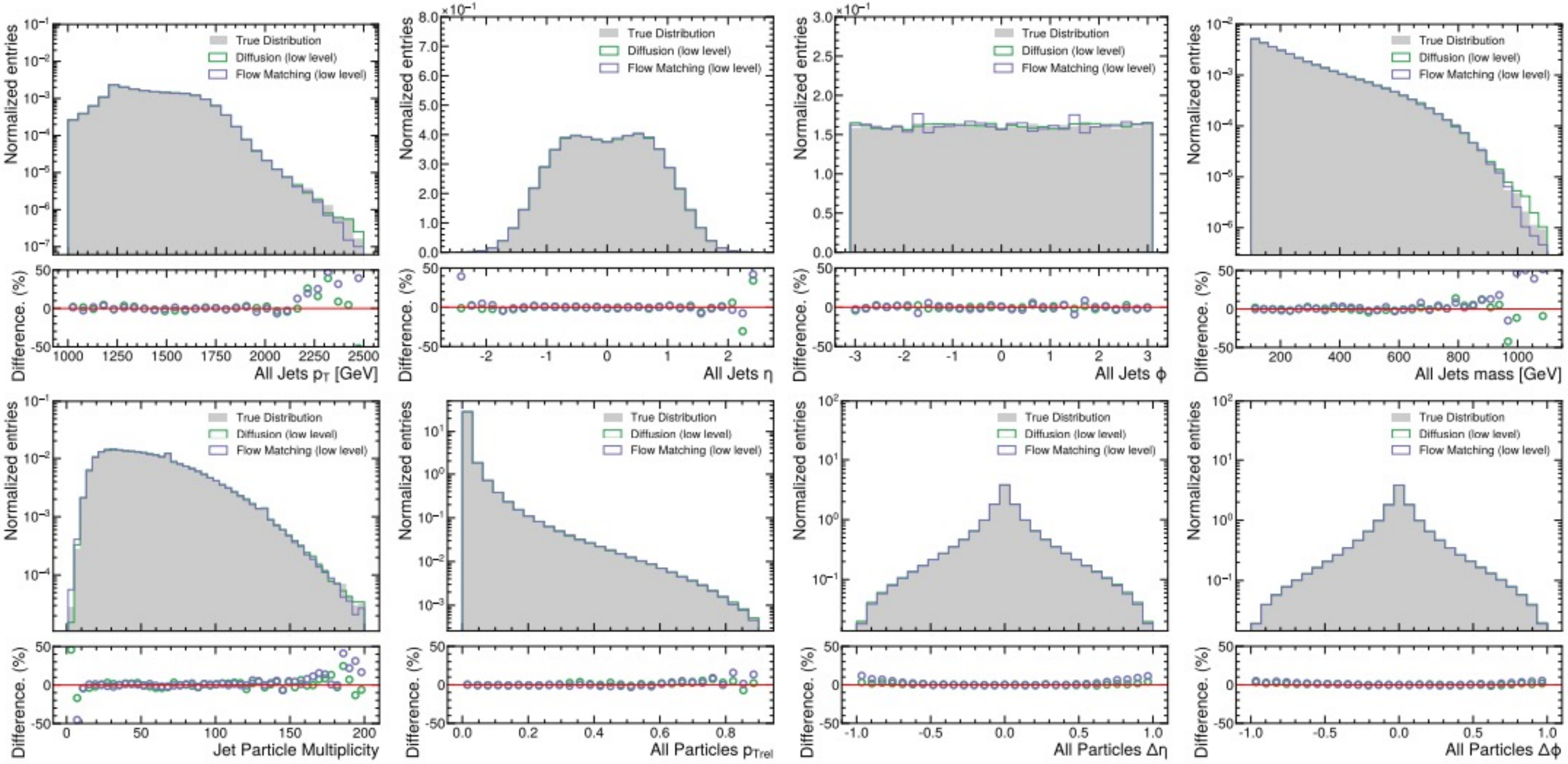
# Results Sideband (SB)

- Classifier AUC: 0.54 (Diffusion) 0.53 (Flow Matching)



# Results Signal Region (SR)

- Classifier AUC: 0.48 (Diffusion) 0.42 (Flow Matching)



# Results

- Classifier evaluated on SIC and ROC curve

$$SIC = \frac{TPR}{\sqrt{FPR}}$$

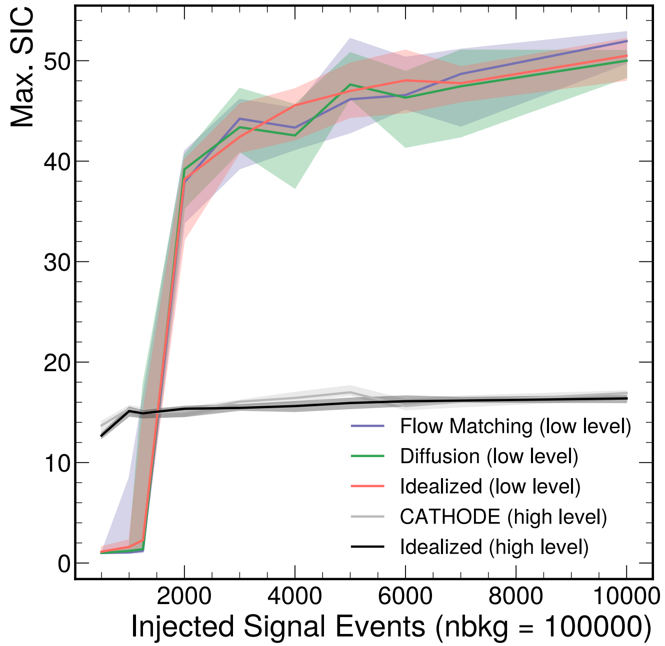
- Both models perform similarly

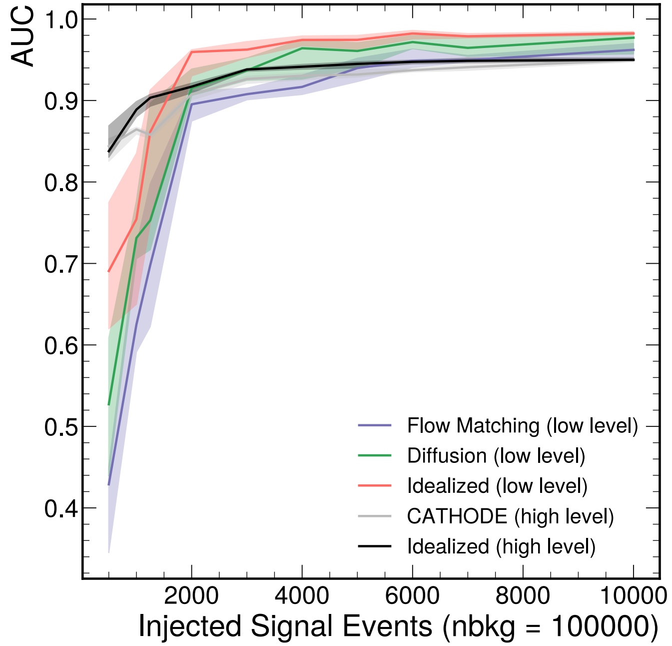
➢ Generative models can fool the classifier

➢ Much higher significance can be achieved for large signal injections

➢ Idealized performance is saturated

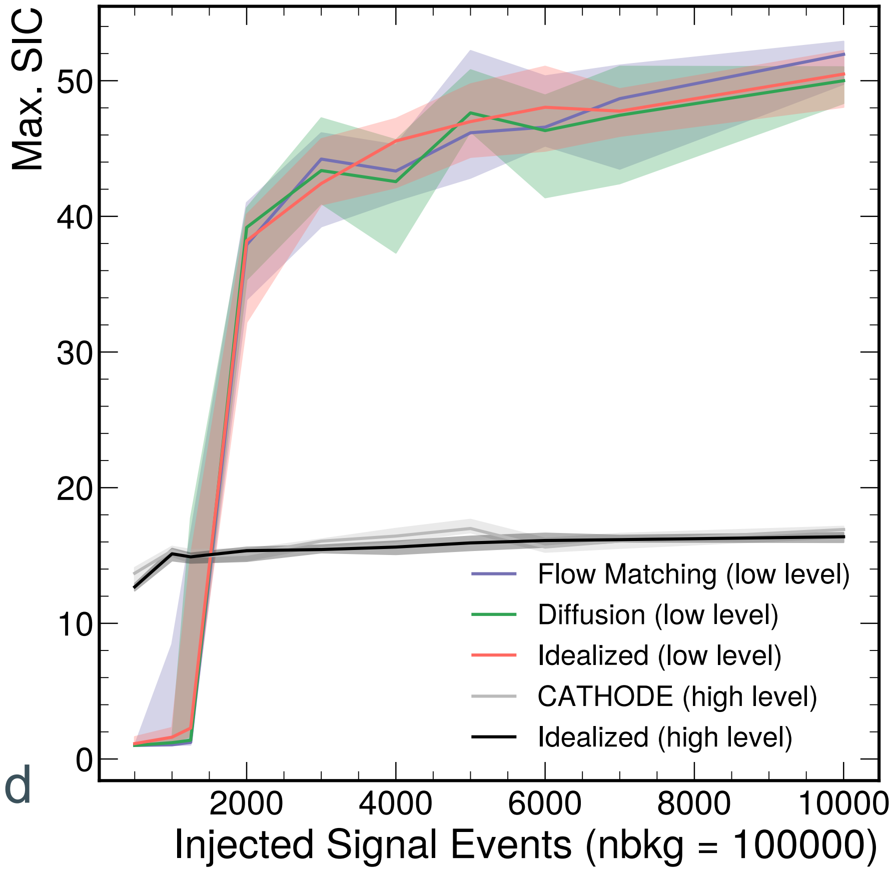
➢ For low signal injections, CATHODE with hand-picked features performs better



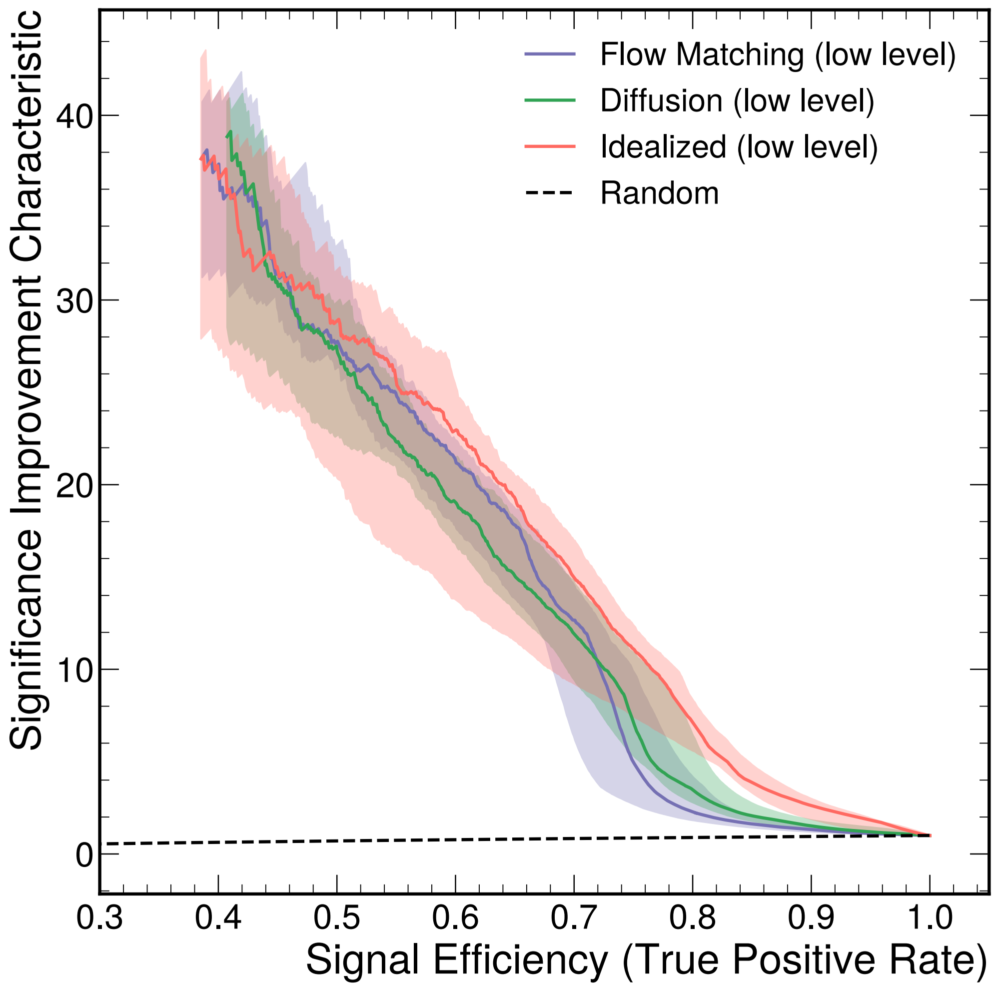
# Conclusion

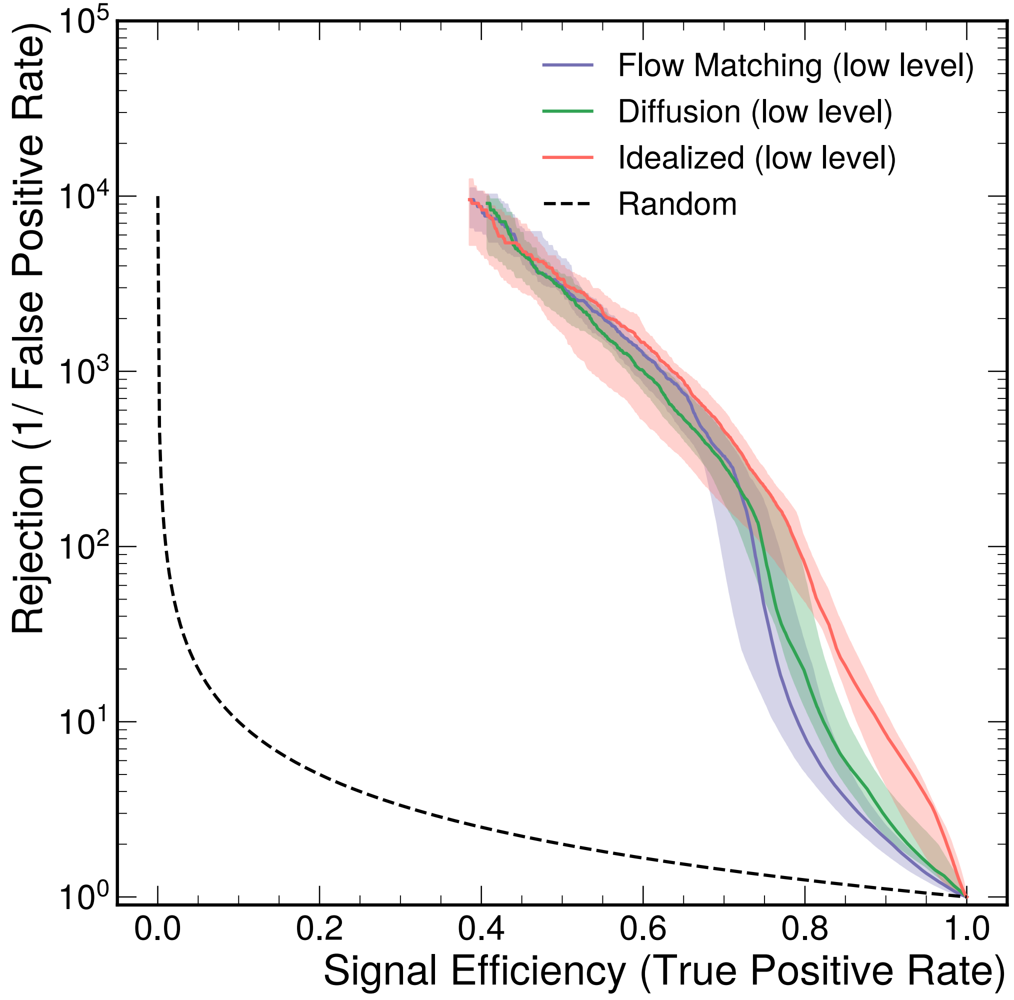
- Model-independent BSM search is important
- Hand-selected features might not contain anomaly
- We applied CATHODE to the full phase space using two state-of-the-art generative models
- Anomalies can successfully be identified
- Larger significance than before
- Currently only sensitive to large signal injections
- Future innovations might lower the signal injection threshold
- Paper on [arxiv:2310.06879](https://arxiv.org/abs/2310.06879)



# Additional Slides

# SIC / ROC Curve 2000 signal injection



# Hyperparameters

| Hyperparameter | Jet-Diffusion | Particle-Diffusion | Jet-FM | Particle-FM | Classifier |
|---|---|---|---|---|---|
| Explicit Conditioning | $t$, $m_{jj}$ | $t$, $p_T$, $\eta$, $\phi$, $m$, $N$, $m_{jj}$ | $t$, $m_{jj}$ | $t$, $p_T$, $\eta$, $\phi$, $m$, | / |
| Time Embedding | Fourier [43] | Fourier [43] | Sin/Cos | Cosine [58] | / |
| Activation function | LeakyReLU(0.01) [59] | LeakyReLU(0.01) [59] | ELU [60] | LeakyReLU(0.01) [59] | LeakyReLU(0.01) [59] |
| Batch size | 128 | 128 | 128 | 1024 | 128 |
| Optimizer | Adam [61] | Adam [61] | AdamW [62] | AdamW [62] | Adam [61] |
| Initial learning rate | $1.6 \times 10{-}3$ | $1.6 \times 10^{-3}$ | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ |
| Weight decay | / | / | $5 \cdot 10^{-5}$ | $5 \cdot 10^{-5}$ | / |
| Learning rate scheduling | Cosine annealing [47] | Cosine annealing [47] | Constant | Cosine with warm-up | / |
| Warm-up epochs | / | / | / | 500 | / |
| Training epochs | 300 | 300 | 10000 | 5000 | 300 |
| Early stopping patience | 50 | 50 | 100 | / | 50 |
| Number of GPUs | 16 | 16 | 1 | 1 | 16 |
| Model weights | $\sim 1.3M$ | $\sim 1.4M$ | $\sim 380k$ | $\sim 2M$ | 438k |