Adaptive Machine Learning on FPGAs: Bridging Simulated and Real-World Data in High-Energy Physics
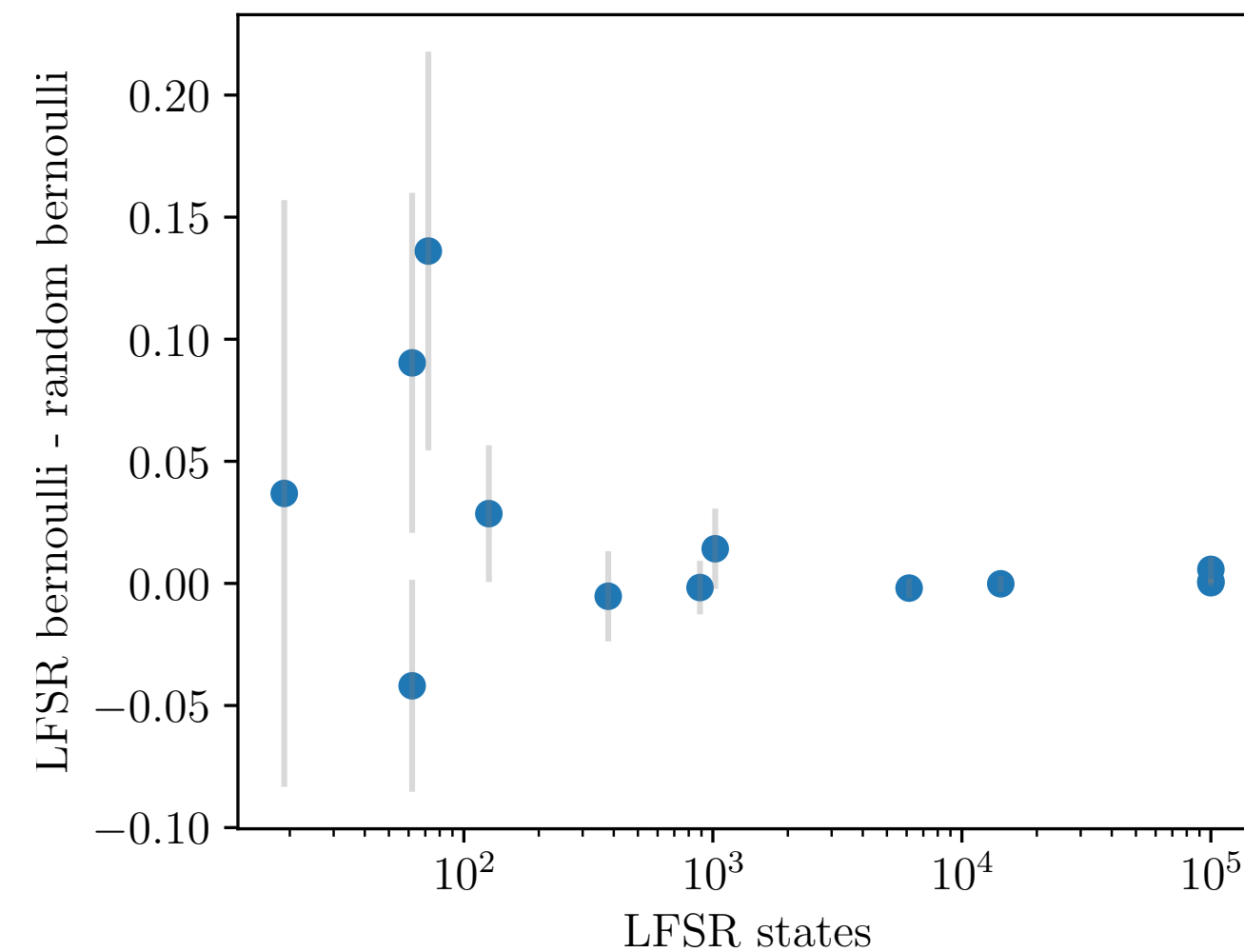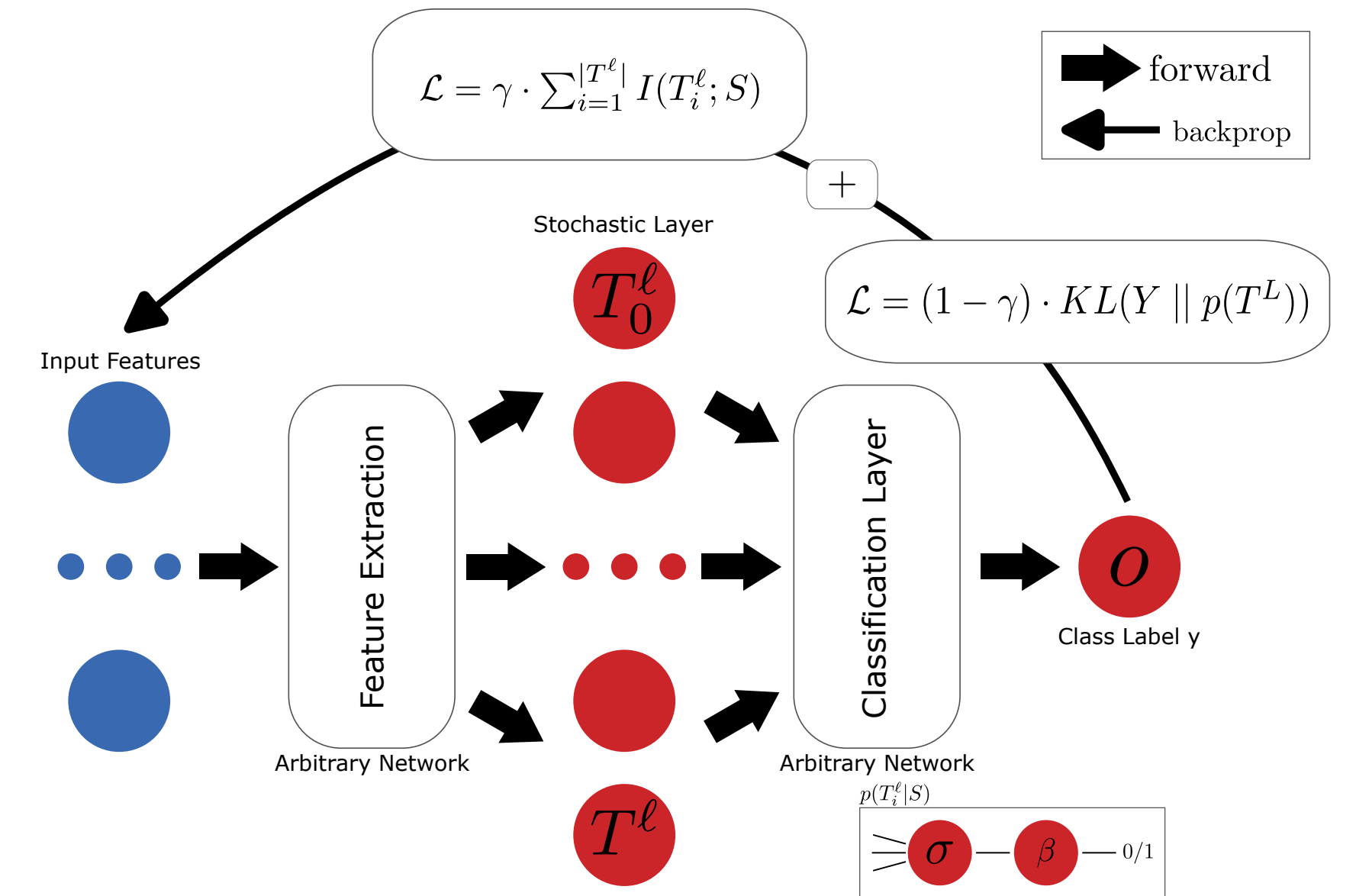
Mattia Cerrato & **Marius Köppel**

Institute of Computer Science Johannes Gutenberg-Universität Mainz

Institute for Particle Physics and Astrophysics ETH Zürich

Marius Köppel 30.04.2024

ETH zürich JG|U

Motivation

- How to train **invariant representations** to bridge simulation and real-data?
- Use **binary stochastically** activations to treat as random variables
- Adversarial classifier need around **2x neurons** domain transformation task
- **Stochastic quantization** neurons by Bernoulli
- On FPGAs use **linear-feedback-shift-register** for Bernoulli distribution

$$\mathcal{L} = \gamma \cdot \sum_{i=1}^{|T^\ell|} I(T_i^\ell; S)$$

forward
backprop

$$\mathcal{L} = (1 - \gamma) \cdot KL(Y \parallel p(T^L))$$

Stochastic Layer

T_0^ℓ

Input Features

Feature Extraction

Classification Layer

O

Class Label y

Arbitrary Network

Arbitrary Network

T^ℓ
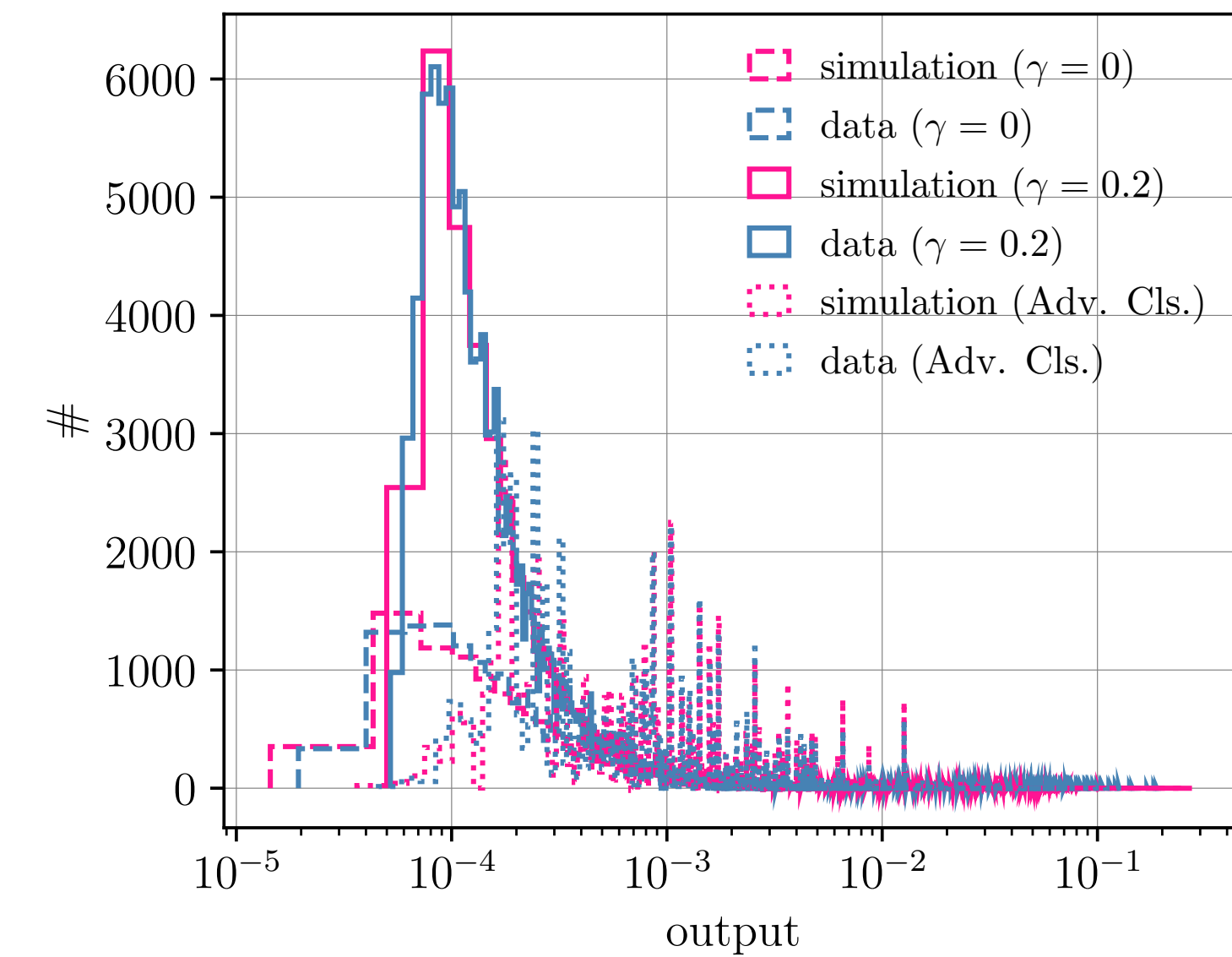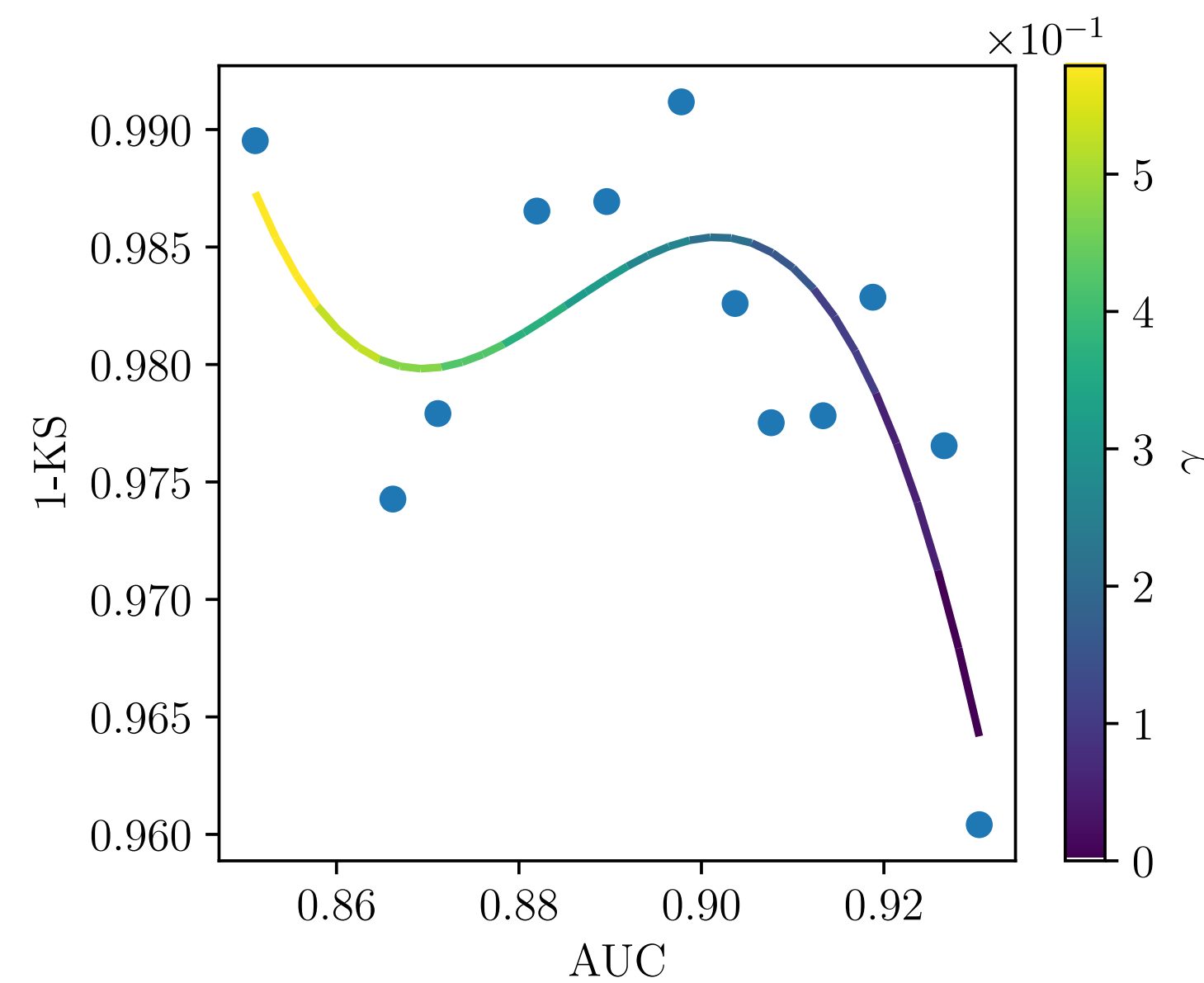
$p(T_i^\ell|S)$

$\sigma - \beta - 0/1$

$$H(T_i^l) = -(1 - \theta_i^l) \cdot \log_2(1 - \theta_i^l) - \theta_i^l \cdot \log_2(\theta_i^l)$$

$$\sum_{i=1}^{|T_l|} I(T_i^l; S) \geq I(T_l; S)$$

[1] https://arxiv.org/abs/2208.02656

2

Experiments

- We analyze the method on LHCb data [2]
- Method can outperform **adversarial classifier** & **normal NN**
- Method obtains stable **invariant/accuracy tradeoffs**
- Use method to be **pile-up** invariant
- Revisit **information bottleneck theory**

Adaptive Machine Learning on FPGAs:
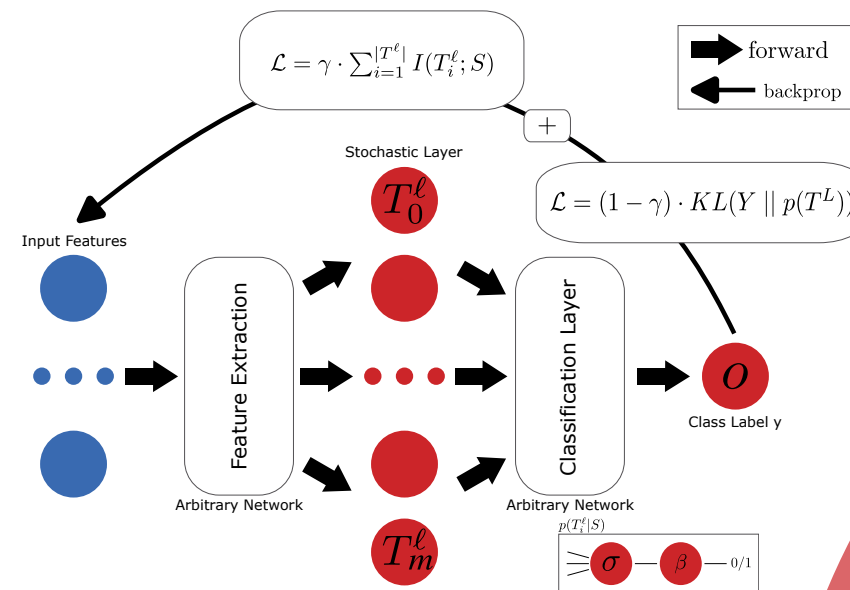# Bridging Simulated and Real-World Data in High-Energy Physics

Mattia Cerrato[1] & Marius Köppel[2]
[1]Institute of Computer Science Johannes Gutenberg-Universität Mainz
[2]Institute for Particle Physics and Astrophysics ETH Zürich

Abstract

- Use **binary stochastically activations** to treat neurons as random variables [1]

$$\mathcal{L} = \gamma \cdot \sum_{i=1}^{|T_i^l|} I(T_i^l; S)$$

forward
backprop

$$\mathcal{L} = (1 - \gamma) \cdot KL(Y \parallel p(T^L))$$

Stochastic Layer

T_0^l

Input Features

Feature Extraction

Classification Layer

O

Class Label y

Arbitrary Network

Arbitrary Network

T_m^l

$p(T_i^l|S)$

σ p → 0/1

Introduction

- Be invariant to **differences** in **simulation** and **data**

- **Adversarial classifier** need **2x** neurons to perform domain transformation task

- Computing of mutual information in **deterministic** neural networks is **hard** or even **impossible**

Method

- **Stochastic quantization** neurons by **Bernoulli**

$$H(T_i^l) = -(1 - \theta_i^l) \cdot \log_2(1 - \theta_i^l) - \theta_i^l \cdot \log_2(\theta_i^l)$$

- On **FPGAs** use linear-feedback-shift-register for Bernoulli distribution
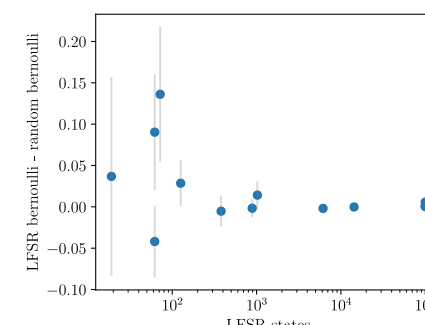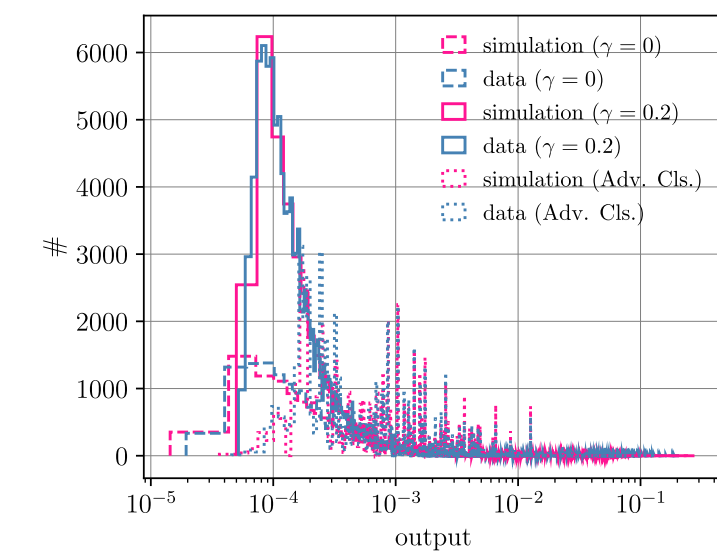
$$H(T_i^l) \qquad H(S)$$

Direct minimization of <u>mutual information</u> in a full precision network using binary <u>stochastically-activated</u> layers for obtaining <u>invariant</u> representations
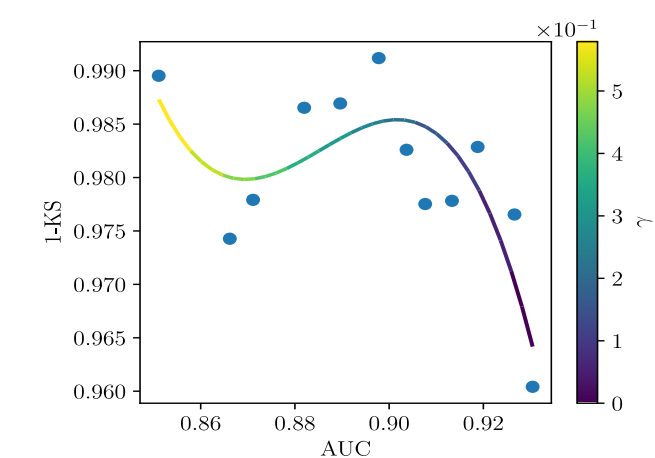
$$\sum_{i=1}^{|T_l|} I(T_i^l; S) \geq I(T_l; S)$$

Experiments

- We analyze the method on **LHCb data** [2]

Results

- Method can **outperform adversarial classifier** & **normal NN**

- Method obtains **stable** invariant/accuracy **tradeoffs**

Outlook

- Use method to be **pile-up** invariant
- Revisit **information bottleneck** theory

References
[1] https://arxiv.org/abs/2208.02656
[2] https://www.kaggle.com/competitions/flavours-of-physics

ETH zürich JG|U