UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Use of deep learning generative models for Monte Carlo event simulation in the context of LHC experiments

Raúl Balanzá García

Cotutors: Jon Ander Gómez Adrián, José Francisco Salt Cairols, Roberto Ruiz de Austri Bazan

3rd Workshop on Compact Objects, Gravitational Waves and Deep Learning

University of Minho

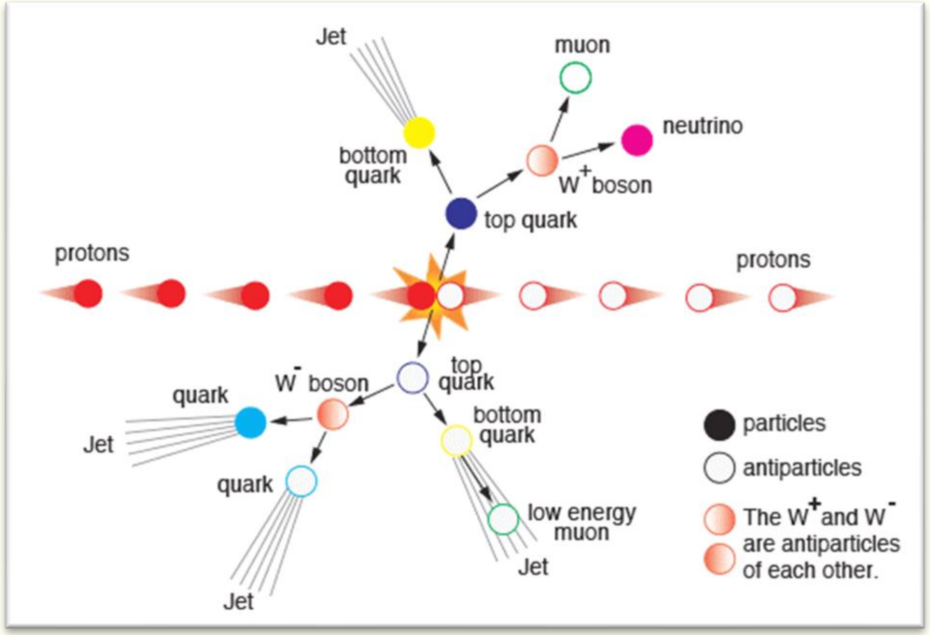
Index

INTRODUCTION

1. Introduction

Problem definition

- Standard Model: current theoretical physics framework
  - Does not explain some key aspects of the behaviour of matter
- LHC (CERN) → experimentation looking for new models
  1. *p-p* collisions are produced to obtain **real data**
  2. **Simulated data** is generated based on new models: *New Physics*
  3. Both data are compared → verify new model



1. Introduction
Objetives

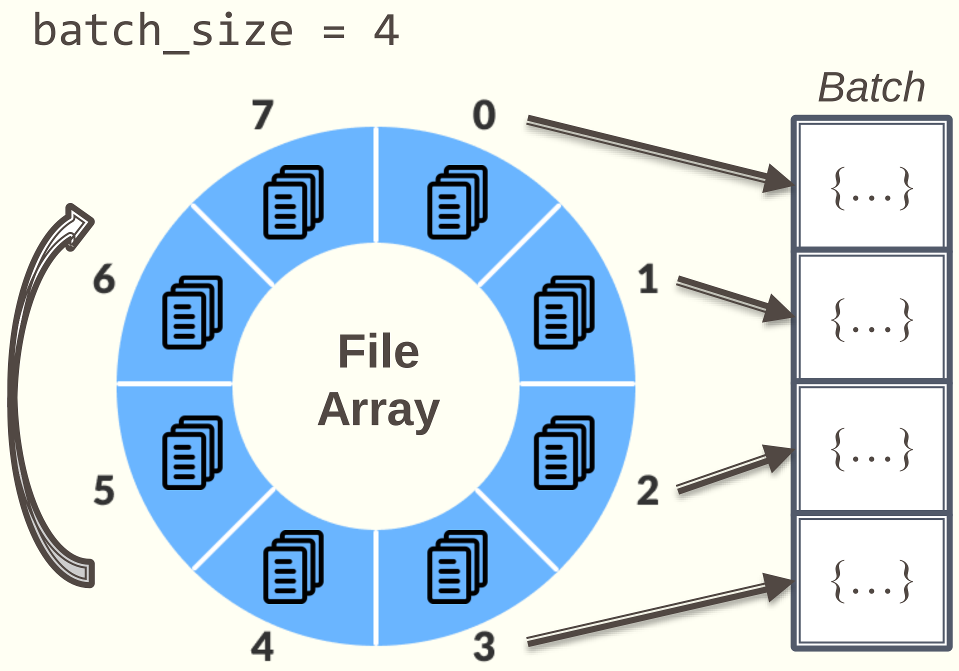
- Generating simulated events is very costly
  - Traditional methods (Monte Carlo) consume lots of **time** and **energy**
  - Future experiments → billions of events → not feasible with current models
  - *Solution:* generative models → generate events in an **efficient** way

- Objetives
  - Create generative models to generate events based on data that was generated through simulation by the Monte Carlo method
  - Accelerate the generation process to reduce the energy and time costs
  - Keep the accuracy of generated events
  - Determine which is the best model

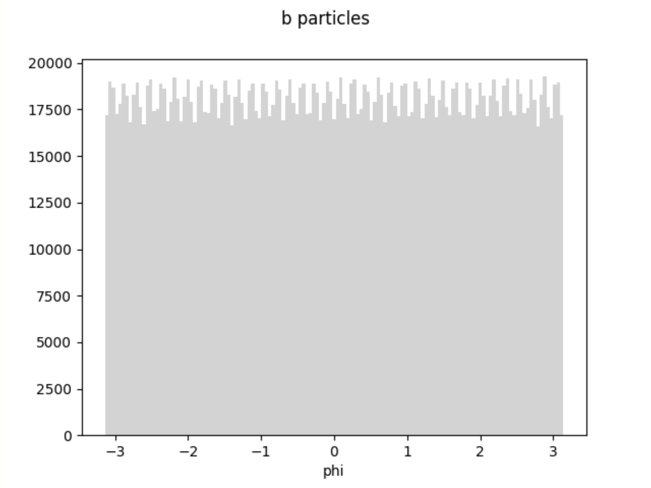
DATA

Physics processes

- *Dataset* with processes from the **Standard Model** and **New Physics**
  - Contains generated events → traditional methods
- Information per **event**:
  - MET
  - METϕ
  - Per each particle (max. 19):
    - E, η, ϕ, p_T
- Used processes:
  - Standard Model: *ttbar*
  - New Physics: *stop_02*
- Huge amount of information
  - Impossible to load in main memory
  - *Solution:* circular buffer

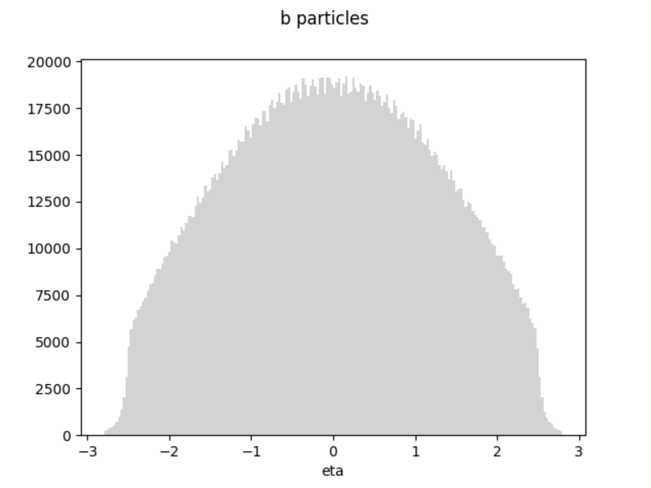


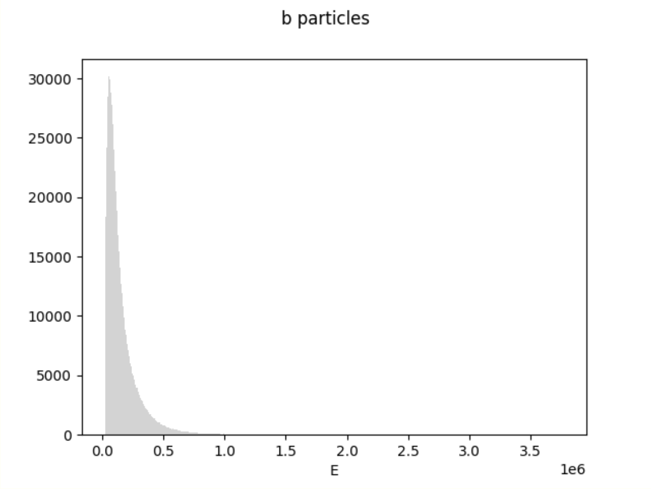
3. Data
Probability distributions



← ϕ y $MET\phi$: uniform

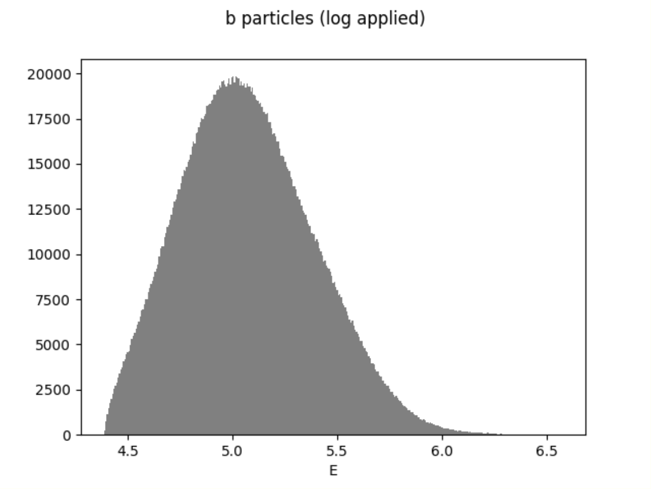
η: similar to gaussian with trimmed extreme values →





- MET, E y p_T: log-normal

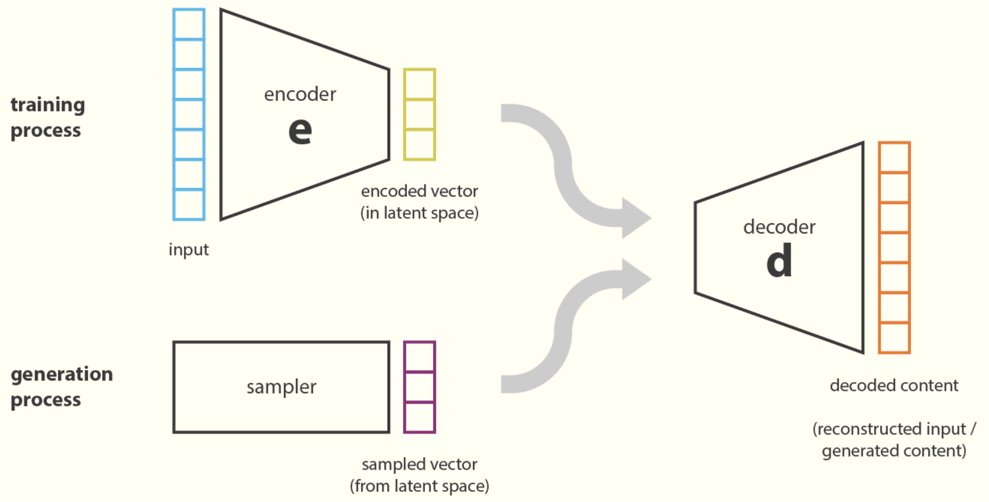
- Applied logarithm to resemble a **normal** distribution → makes learning easier



MODELS

2. Models
Variational Autoencoders

- Generative model
- Training/generation:
  1. Encode the input to the *latent space*
  2. Generate data following its probability distribution*
  3. Decode data to input dimensions
- Variants: β-VAE and α-VAE

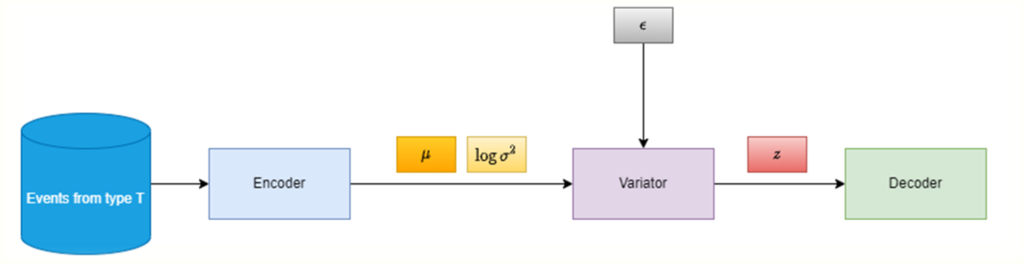


2. Models

VAE variants

β-VAE

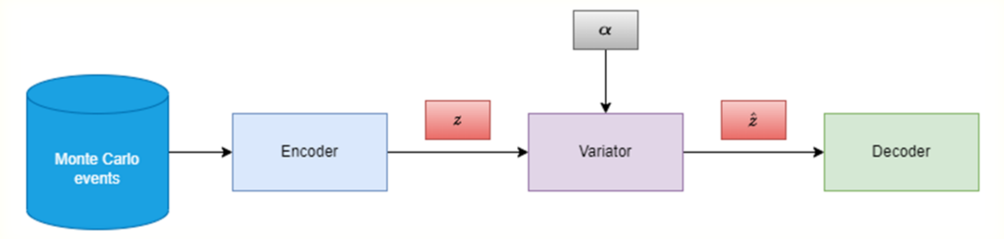
- *Loss function* weights are proportional to β:
$$L_{VAE} = (1 - \beta)MSE + \beta KL$$
- Encoding: $z = \epsilon * e^{\frac{1}{2}\log \sigma^2} + \mu$
  - $\epsilon \sim \mathcal{N}(\mu = 0, \sigma = 1)$



α-VAE

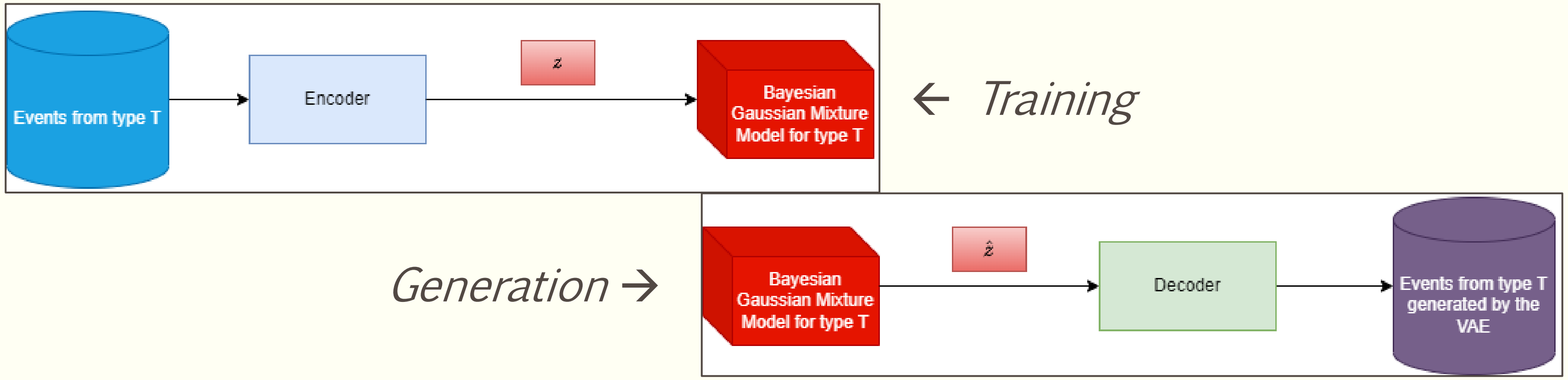
- Gaussian noise is added to the encoding with the following distribution:

$$\mathcal{N}(\mu = 0, \sigma = \alpha)$$



Bayesian Gaussian Mixture Models

- Unsupervised learning model
- Assumes: data can be described by a normal distribution
- Has γ components
  - Each one is a multivariate Gaussian distribution
  - Has its own mean vector and covariance matrix
- Generates data based on the learned distribution.



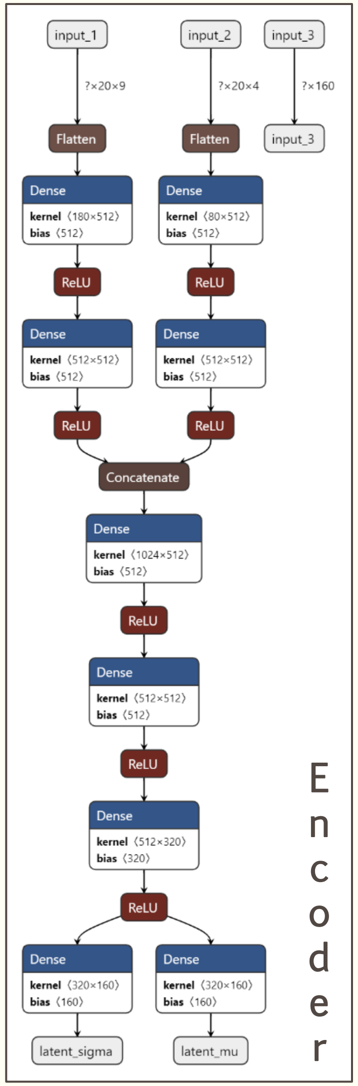
← *Training*

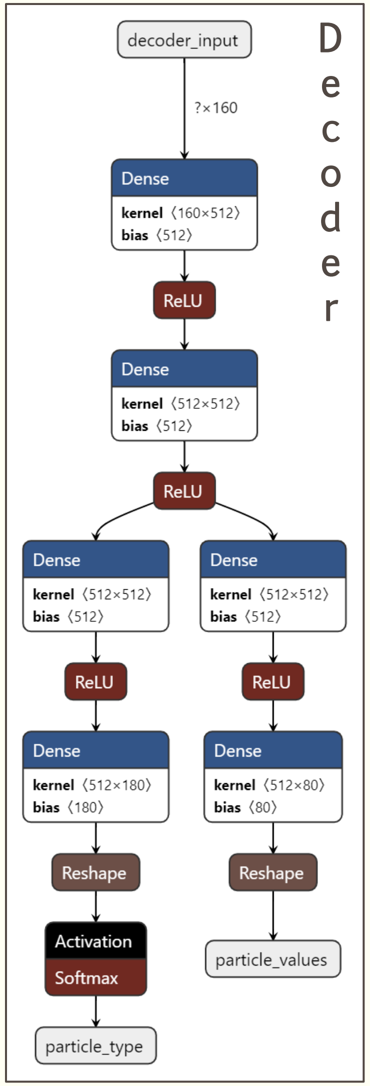
Generation →

EXPERIMENTATION

Experiment I: Initial model

- *MET* + *MET*ϕ considered one more particle
- Input/output data → 2 capas
  - Particle identifier (1-Hot vector)
  - Particle properties (E, η, ϕ, p_T)
- Physics process: *ttbar* (Standard Model)
- Model: β-VAE + BGMM
  - $\beta \in \{0.0, 0.001, 0.01, 0.1, 0.2, 0.5\}$
  - $\gamma \in \{10, 50, 100\}$

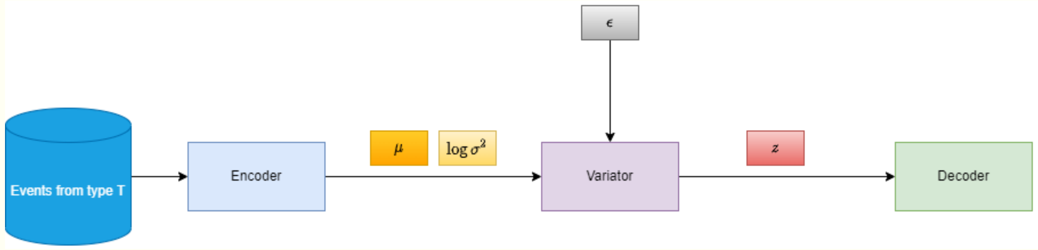




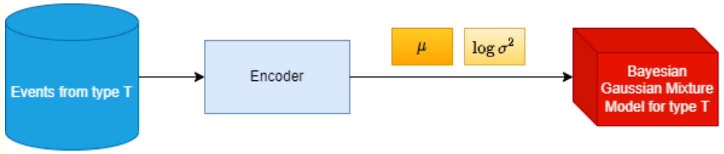
4.1. Experiment I

Training and generation

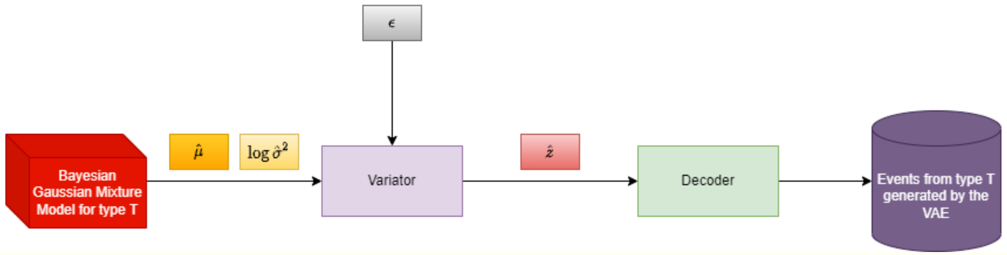
1. Training the β-VAE with original data



2. Training the BGMM with encodings of the *already trained encoder*

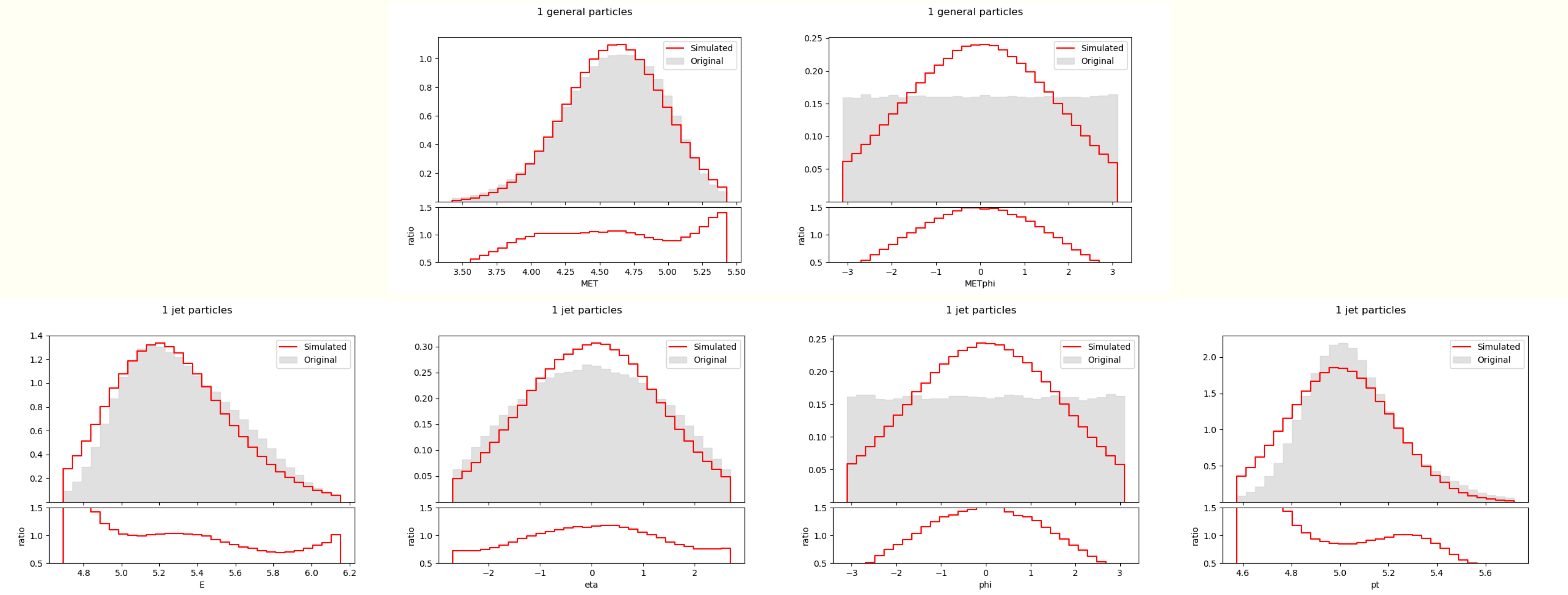


3. Generation of events taking the BGMM as the starting point



4.1. Experiment I

Results ($\beta = 0.01, \gamma = 50$)

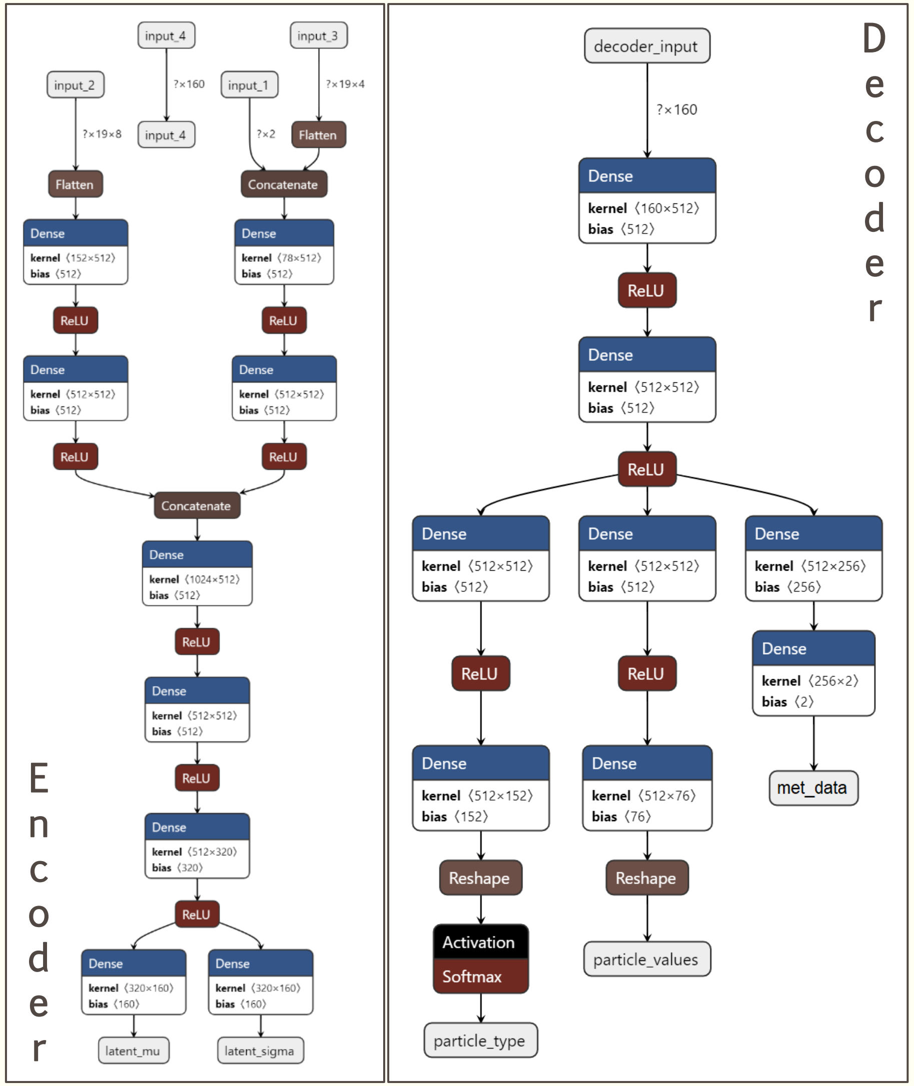


- *Problem:* uniform distributions.

4. Experimentation

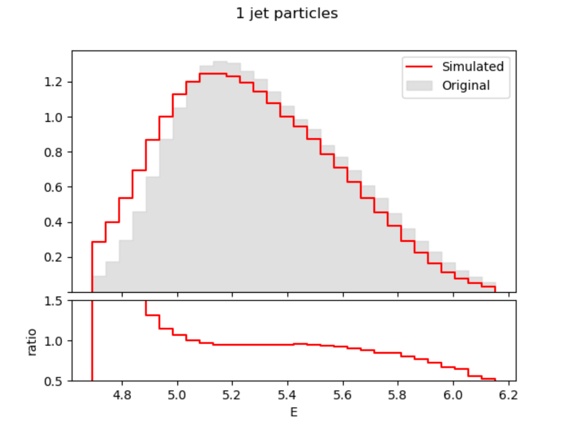
Experiment II: Input splitting

- Split *MET* + *METϕ* in a separate layer
- Same training process
- Physics process: *ttbar* (Standard Model)
- Input/output data → 3 layers
  - Particle identifier (1-Hot vector)
  - *MET* + *METϕ*
  - Particle properties (E, η, ϕ, p_T)
- Model: β-VAE (+ BGMM)
  - $\beta \in \{0.0, 0.001, 0.01, 0.1, 0.2, 0.5, 0.7, 1\}$
  - $\gamma = 100$

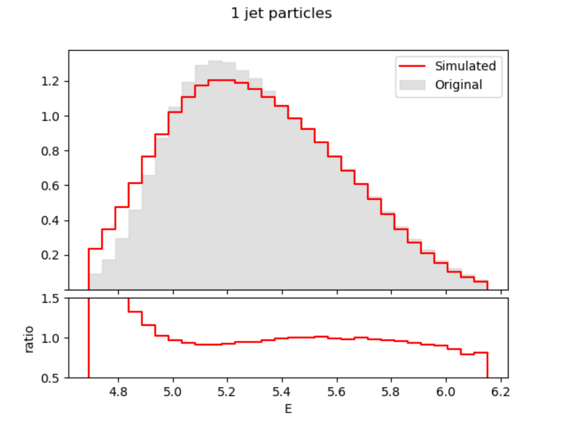


4.2. Experiment II

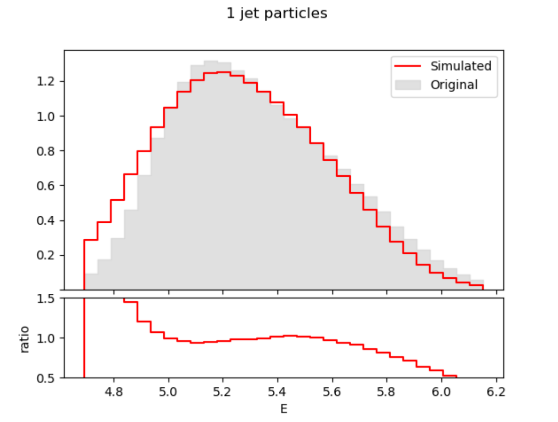
Results – comparing β in attribute E of the 1st jet



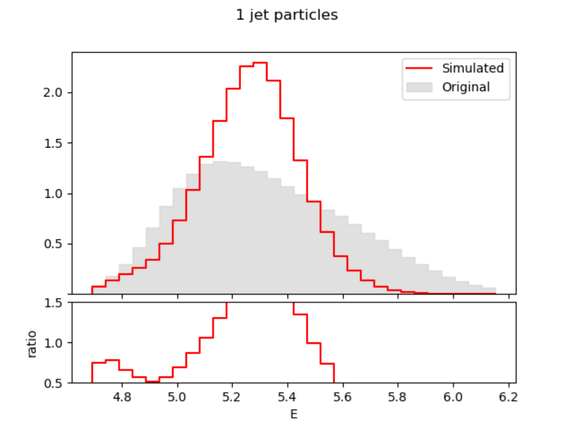
$\beta = 0.0$



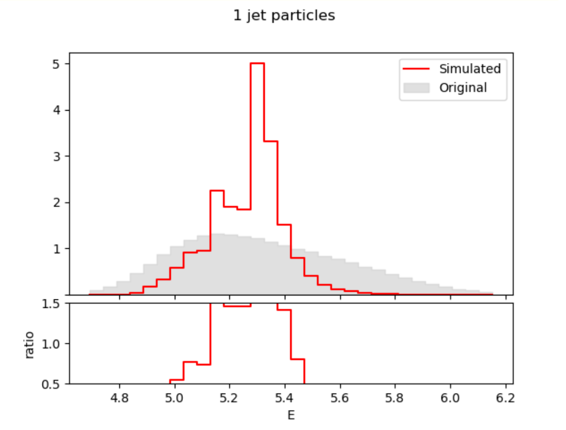
$\beta = 0.001$



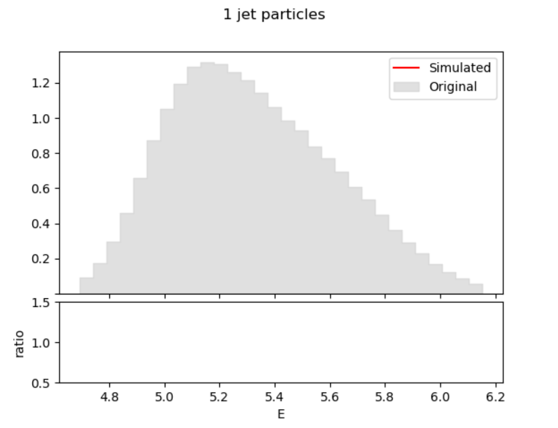
$\beta = 0.01$



$\beta = 0.1$



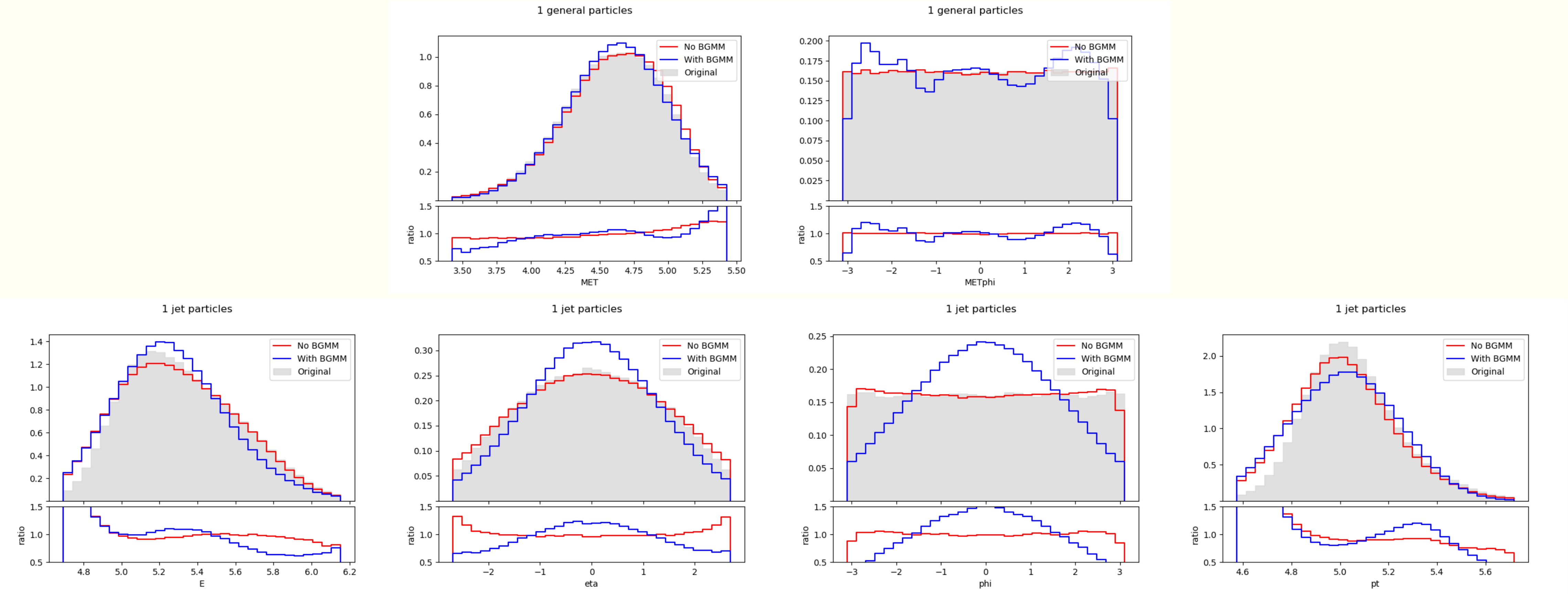
$\beta = 0.5$



$\beta = 1.0$

4.2. Experiment II

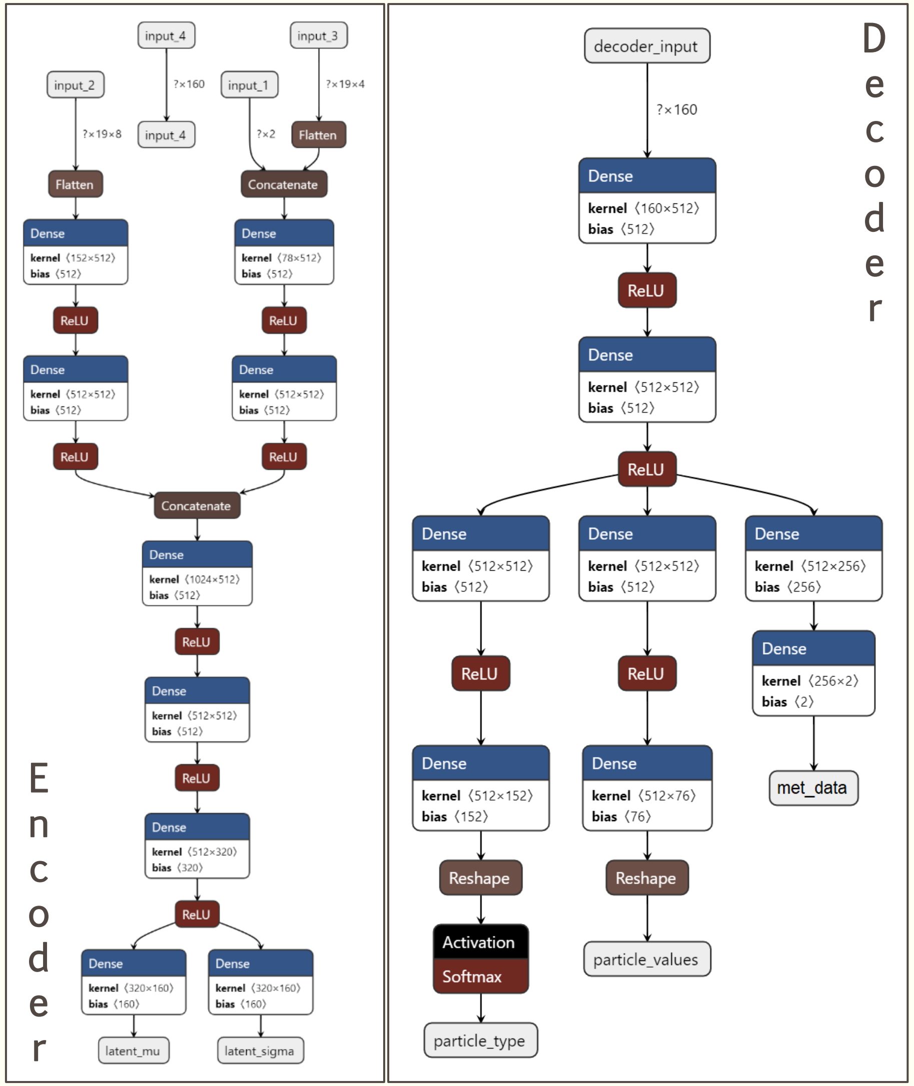
Results ($\beta = 0.001, \gamma = 100$)



- *Observation:* uniform distributions are correct in the model without BGMM.

Experiment III: Introducing the α-VAE

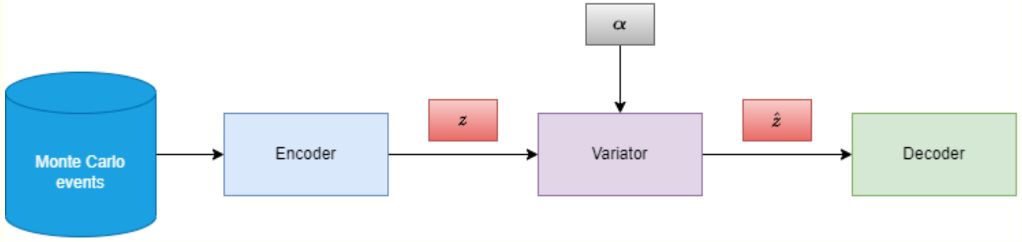
- Same VAE architecture from experiment II
  - The *variator* changes
- Physics process: *ttbar* (Standard Model)
- Model: α-VAE (+ BGMM)
  - $\alpha \in \{0.1, 0.2, 0.3\}$
  - $\gamma = \{10, 20, 50, 100\}$



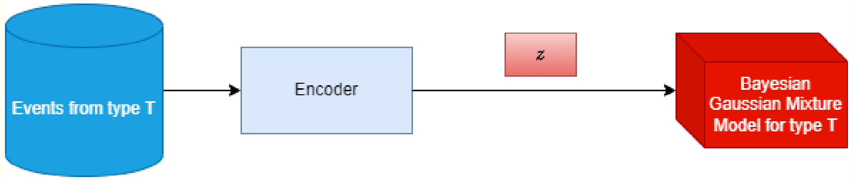
4.3. Experiment III

Training and generation

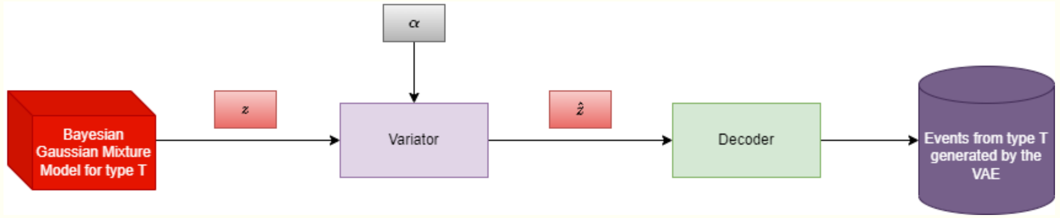
1. Training the α-VAE with original data



2. Training the BGMM with encodings of the *already trained encoder*

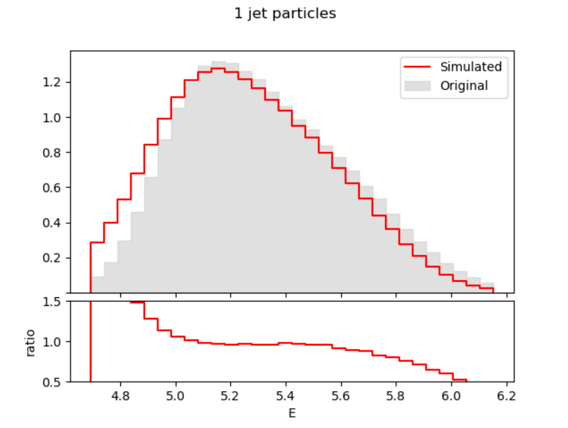


3. Generation of events taking the BGMM as the starting point

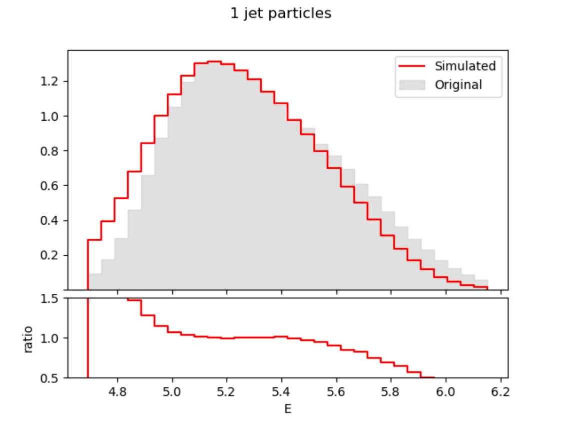


Results – comparing α

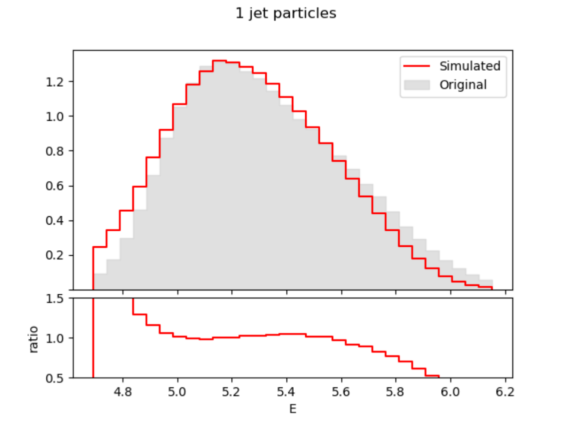
1st jet
Attribute E



$\alpha = 0.1$

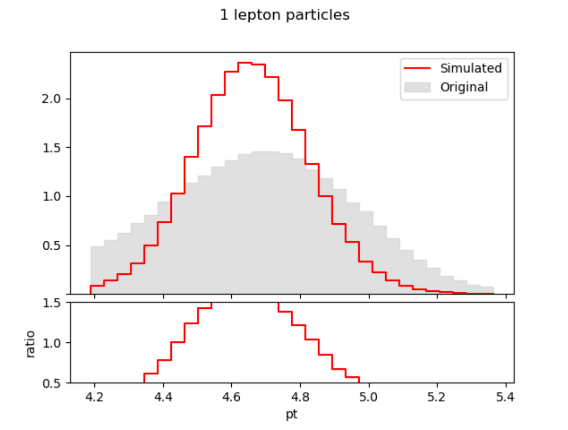


$\alpha = 0.2$

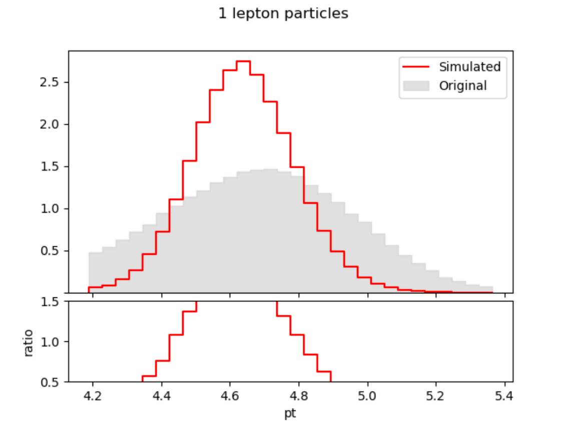


$\alpha = 0.3$

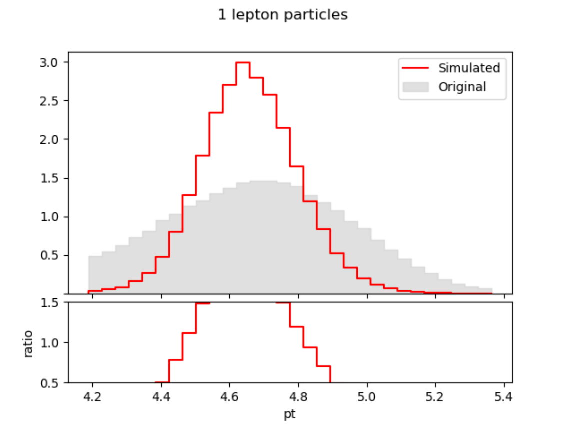
1st lepton
Attribute p_T



$\alpha = 0.1$

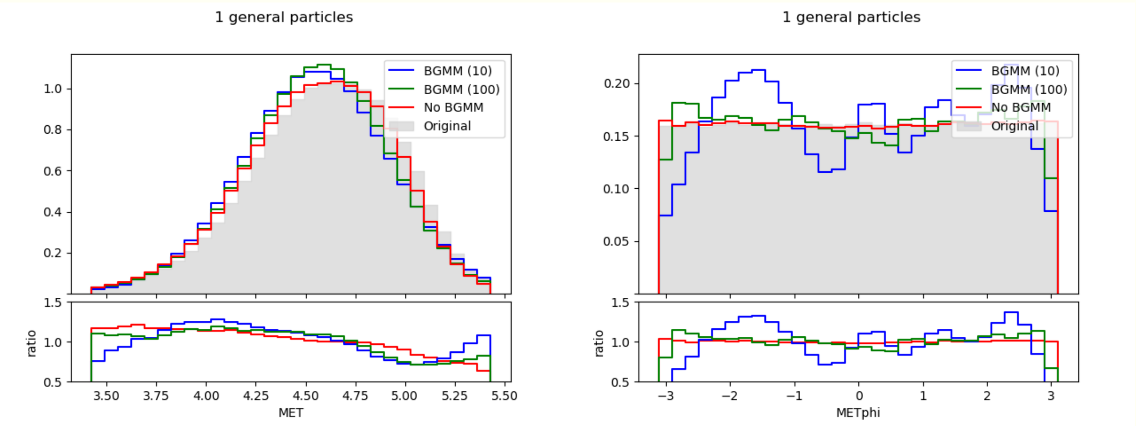


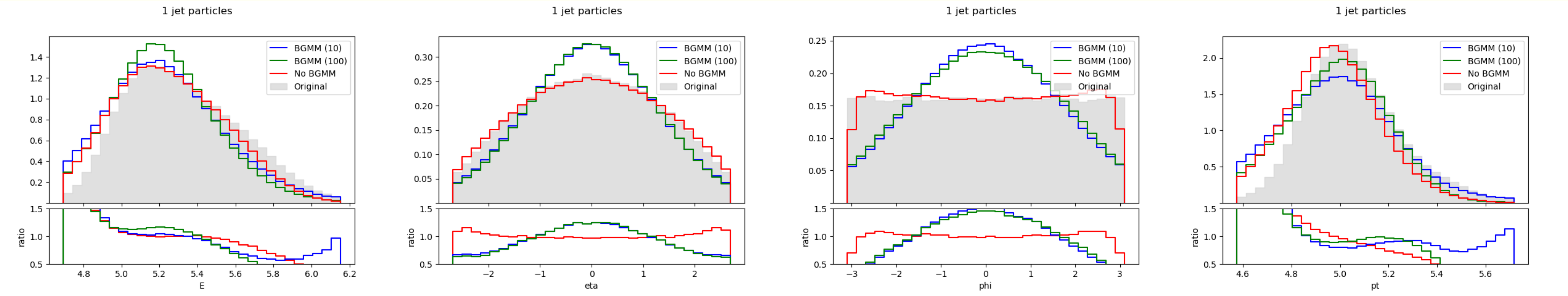
$\alpha = 0.2$



$\alpha = 0.3$

Results ($\alpha = 0.2, \gamma \in \{10, 100\}$)





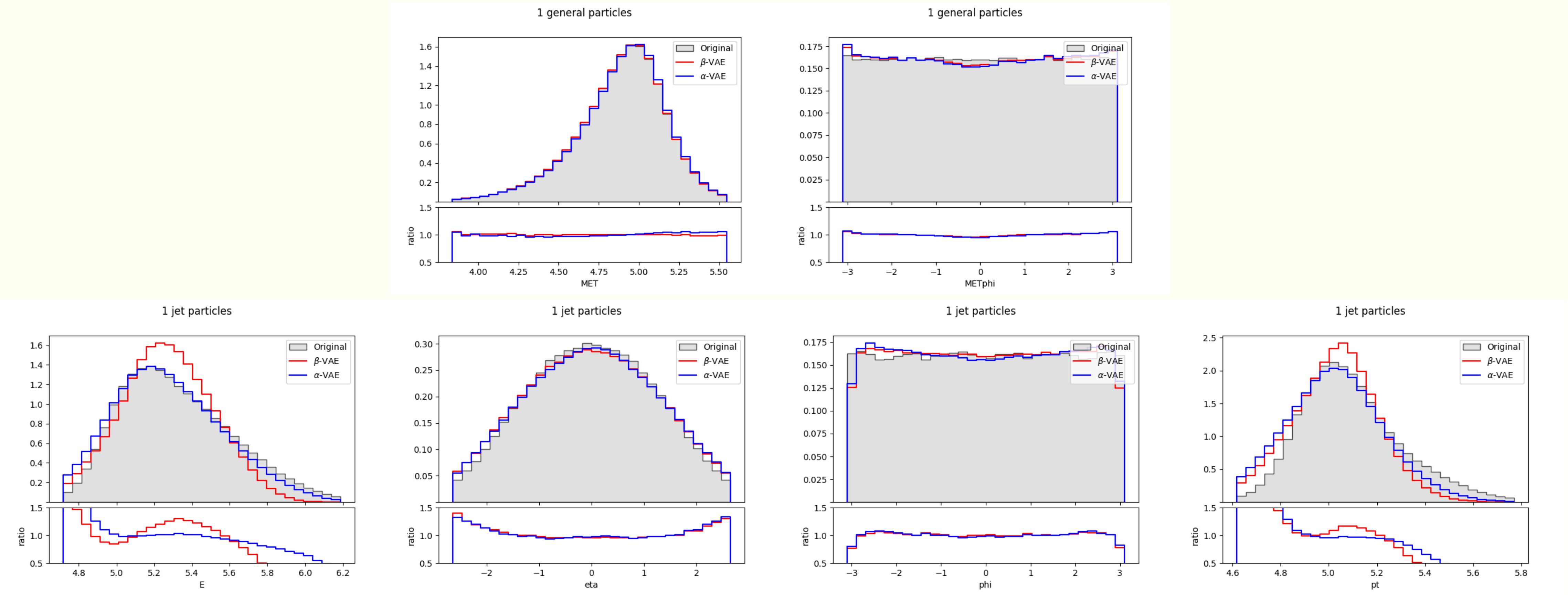
- *Observation:* BGMM keeps obtaining a worse result.

New Physics processes (BSM)

- After selecting the best model → training with a BSM process
- Physics process: *stop_02* (New Physics)
- Selected models:
  - β-VAE without BGMM ($\beta = 0.001$)
  - α-VAE without BGMM ($\alpha = 0.2$)
- We did not test models with BGMM → lack of time

4.4. New Physics

Results ($\beta = 0.001, \alpha = 0.2$)



- *Observation:* Similar results, α-VAE adjusts better in some cases

CONCLUSIONS

4. Conclusions

- Achieved objectives: more efficient event generation
- 2 models with promising results: β-VAE and α-VAE without BGMM
  - We should keep adjusting model parameters to improve results
- BGMM does not obtain the expected results
  - ¿Requires more components?
  - It will be required to perform additional experimentation

4.1. Future work

- Generating events in a fast way is a need of critical importance
  - Further research is required
- Proposals
  - Experiment the use of more components in BGMMs
  - Creation of advanced metrics
  - Usage of other types of models: GAN, *Flow models*…

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Use of deep learning generative models for Monte Carlo event simulation in the context of LHC experiments

Raúl Balanzá García
Cotutors: Jon Ander Gómez Adrián, José Francisco Salt Cairols, Roberto Ruiz de Austri Bazan

3rd Workshop on Compact Objects, Gravitational Waves and Deep Learning
University of Minho

Full text: https://files.raulbalanza.me/pdf/memoria_tfg.pdf