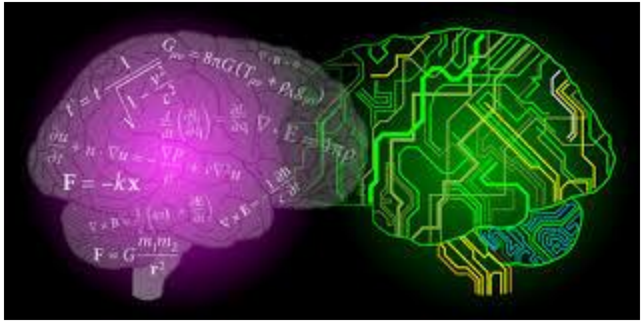
Leveraging transformers, RL and GAs to identify key b-hadron backgrounds

G. Hijano, D. Lancierini, A. Marshall, A. Mauri, P. Owen, M. Patel,
K. Petridis, S. R. Qasim, N. Serra, W. Sutcliffe, H. Tilquin

12/09/2024

Overview

- Motivation
- Previous works
- Current goals
  - Branching Ratio prediction model
  - Generative model
  - Background finder
    - Reinforcement Learning approach
    - Genetic Algorithm approach



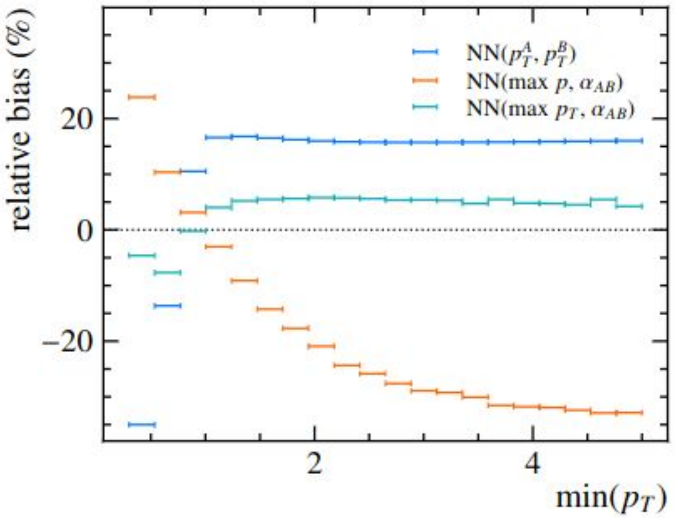
Motivation

- Performing a particle physics analysis involves:
    - simulation
    - thinking about all the backgrounds
    - checking possible mismodellings
    - …
- This can lead to **human errors**.
- Our goal is to use **Machine Learning** (ML) to:
    - automatize the workflow
    - improve the accuracy

Previous works: efficiency mismodelling

Previous works on the Devil's Advocate project: https://link.springer.com/article/10.1140/epjc/s10052-023-11925-w

- Proposal of a new method based on **machine learning** to play the devil's advocate and investigate the **impact of** detector or physics **mismodellings** in a quantitative way
    - Focused on the signal efficiency

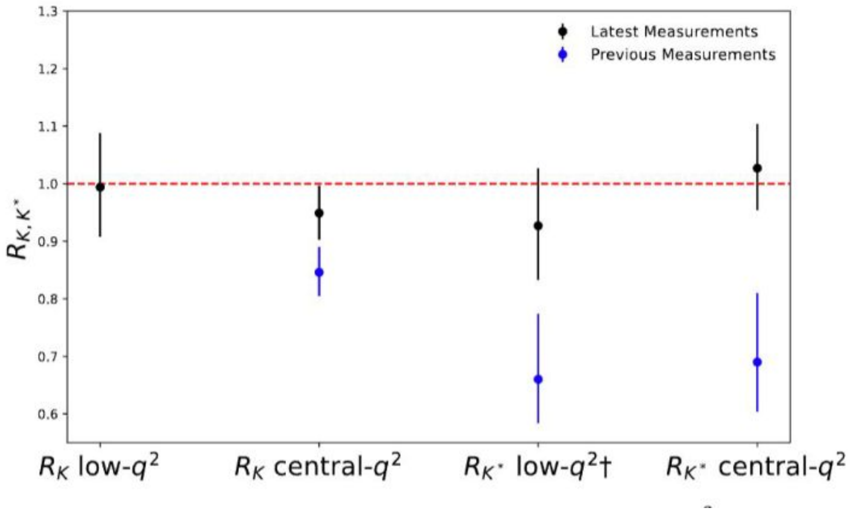


Efficiency mismodelling introduced up to a 30% relative bias (on differential Branching Ratio (BR))

Current goals: backgrounds

- Design an algorithm to automatize the procedure of finding the **most problematic backgrounds** (events that mimic the response of the signal in the detector) for a signal specified by the user
- Missing backgrounds have led to confusion in analyses performed in the past
    - Example:

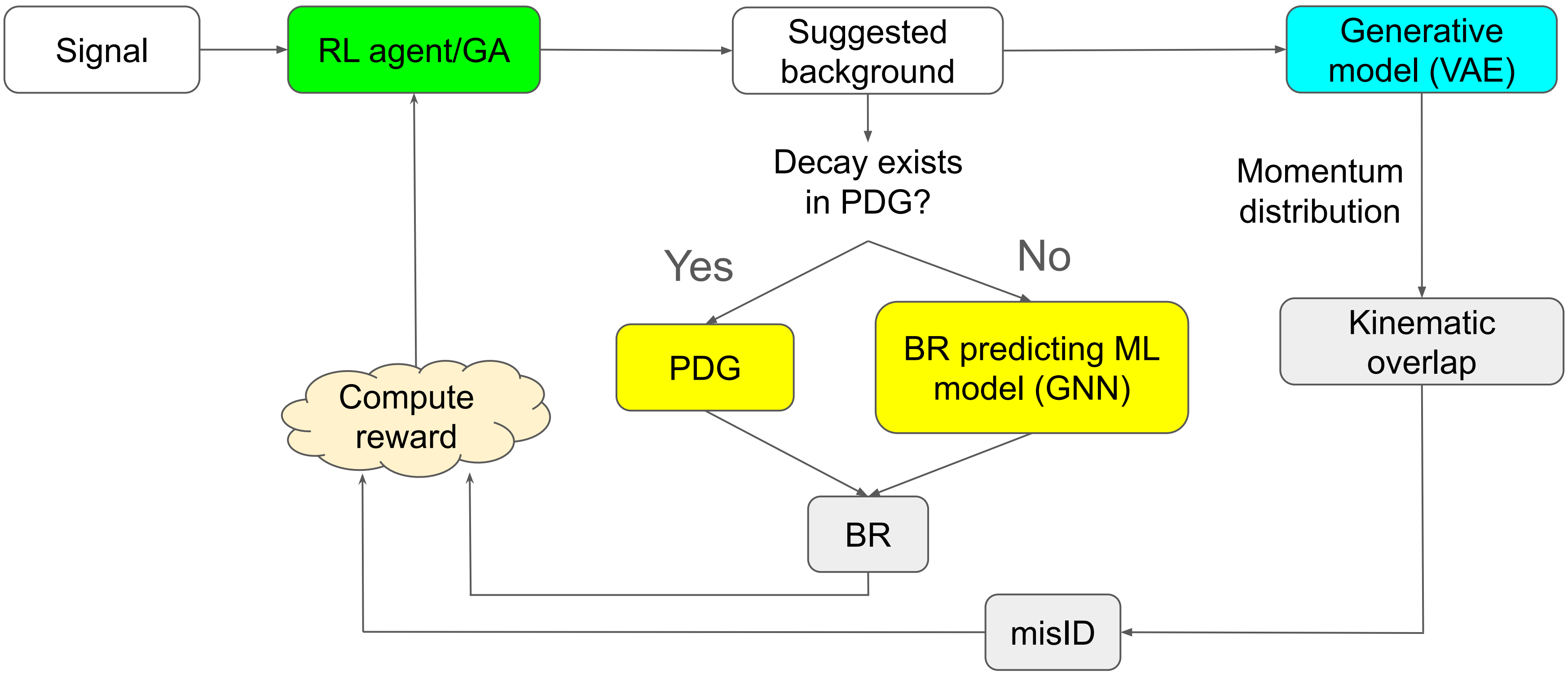
https://journals.aps.org/prd/abstract/10.1103/PhysRevD.108.032002



- R_{K,K^*} measurements initially showed **discrepancies** w.r.t. SM predictions (the expected value was 1)
- Statistical fluctuations? Detector or physics mismodellings? New physics?
    - Reason for discrepancy: **missing background**

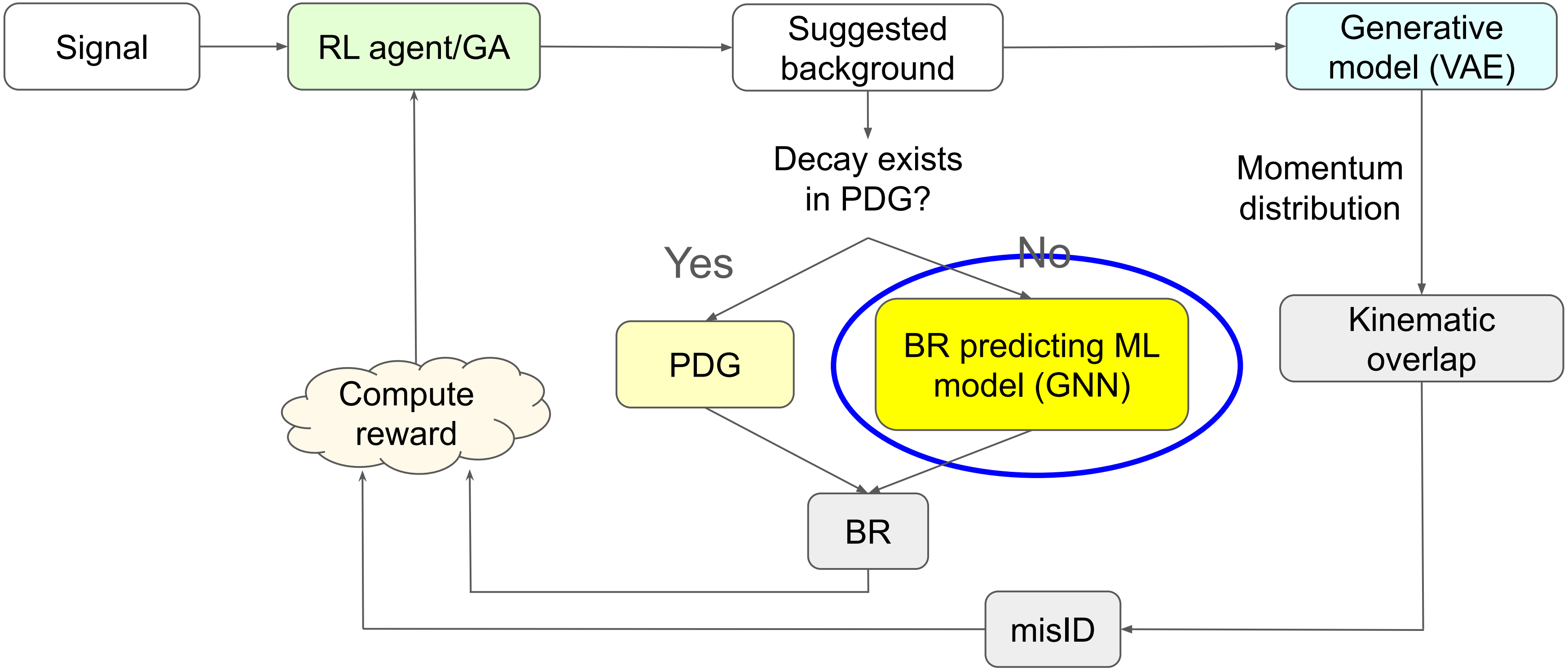
Workflow

Particle Data Group (PDG): https://pdg.lbl.gov/



Reward = BR x misID x Kinematic_overlap

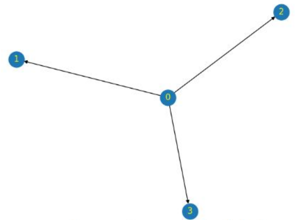
Workflow

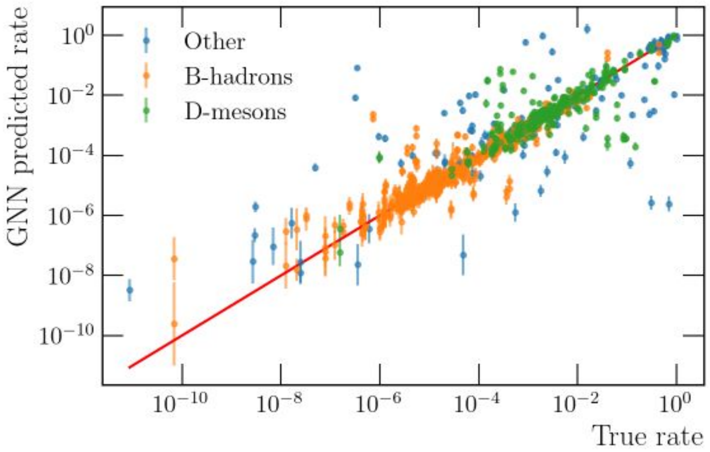


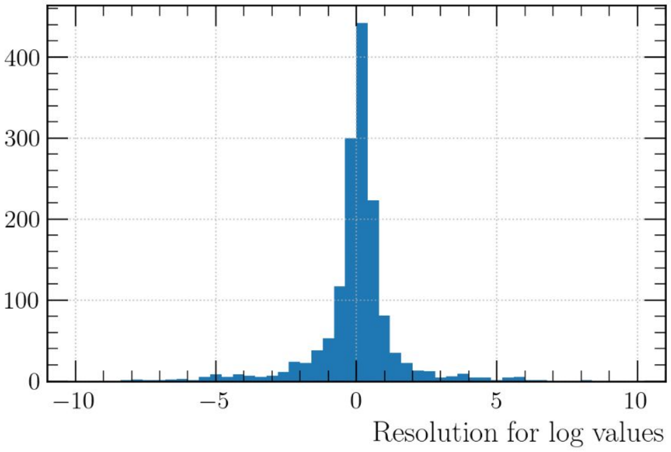
Reward = BR x misID x Kinematic_overlap

BR predictions

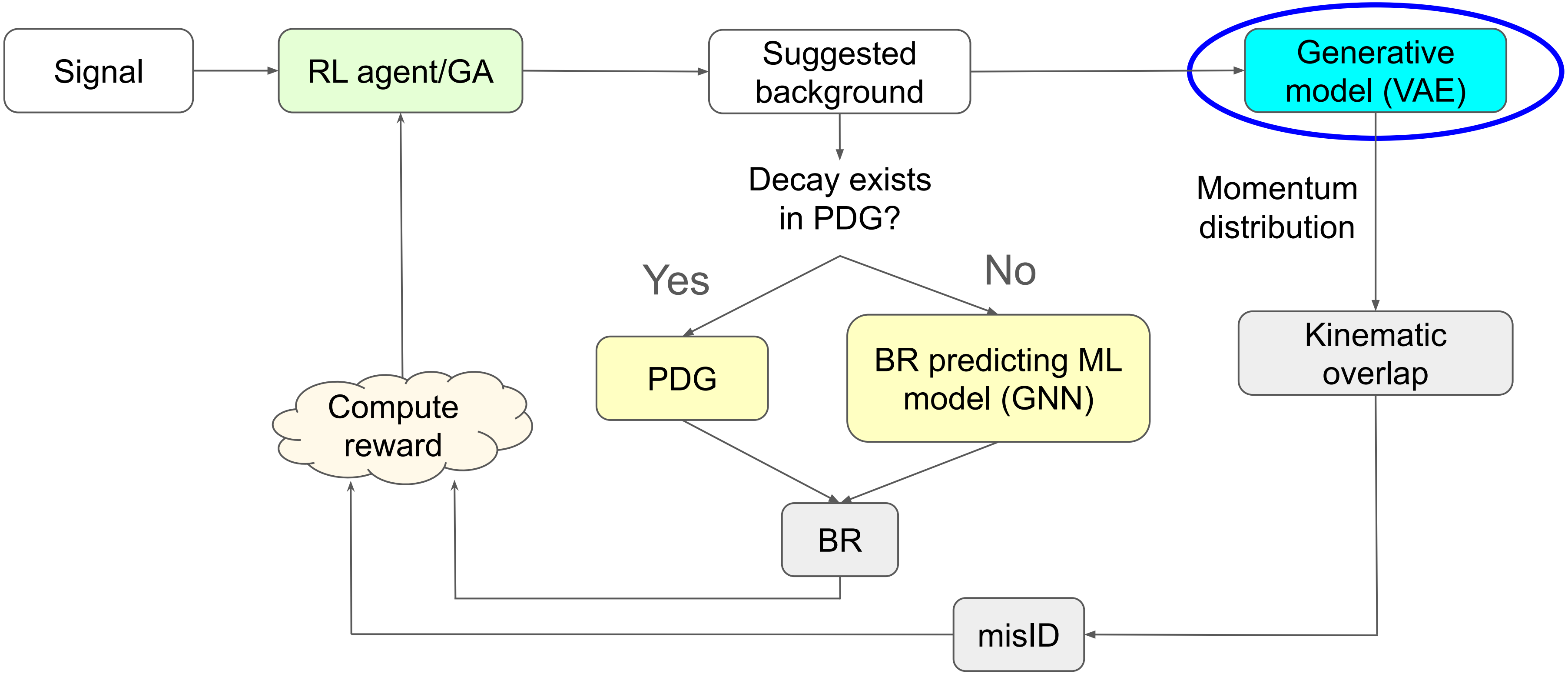
- **Branching Ratio** (BR) predictions for decays not present in the Particle Data Group (PDG)
    - Graph Neural Network (**GNN**) model trained on the PDG data
    - **Graphs** to represent decays
        - Graph is invariant under particle ordering
    - Using a Bayesian architecture to obtain uncertainty on predictions
    - First results seem promising
        - We just need a rough estimate in most cases







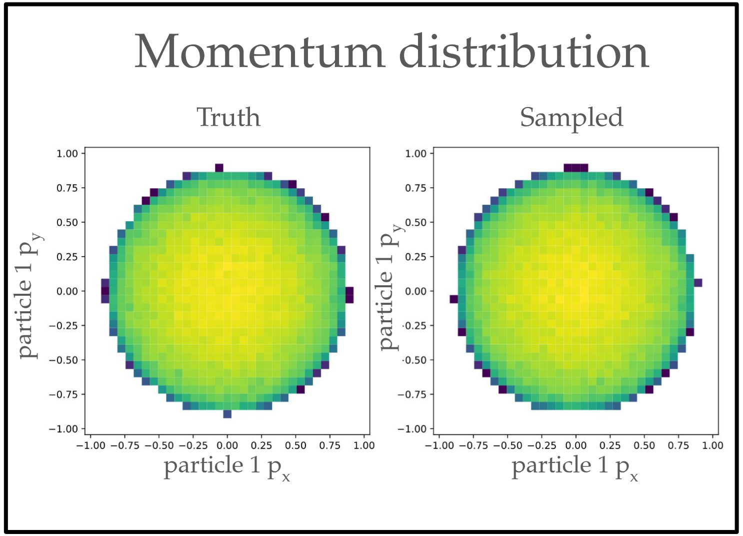
Workflow

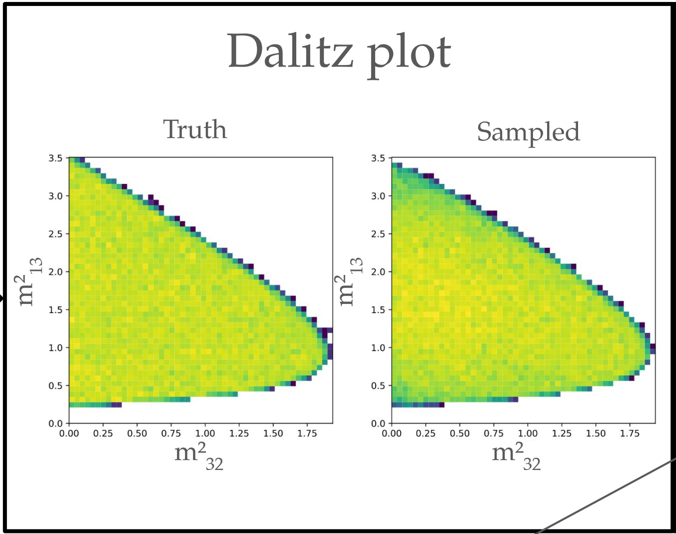


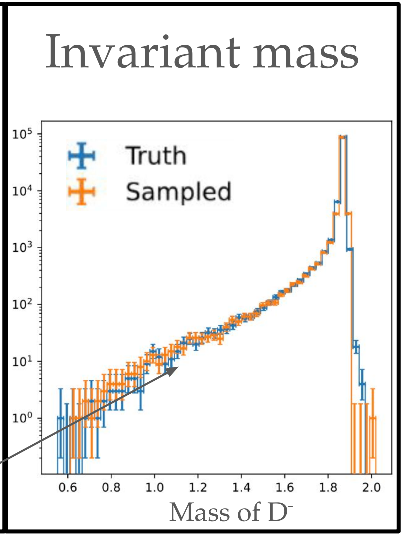
Reward = BR x misID x Kinematic_overlap

Generative model: generator level variables

- Given masses of daughter particles, their **momentum distribution** can be estimated to obtain the **kinematic overlap** between signal and background.
- ML is already being used for fast simulation in particle physics, for example, in: https://link.springer.com/article/10.1140/epjc/s10052-022-10258-4.
- Using Monte Carlo as truth.
- Example: $D^- \rightarrow K^0 e^-$ anti-ν_e Variational Autoencoder (**VAE**) model



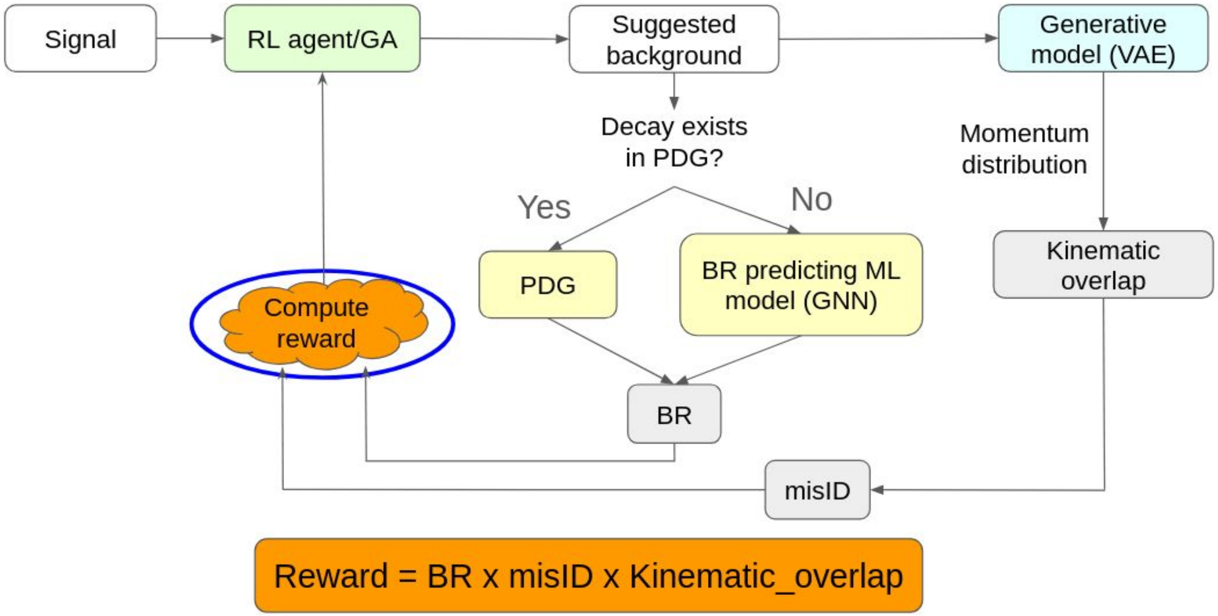




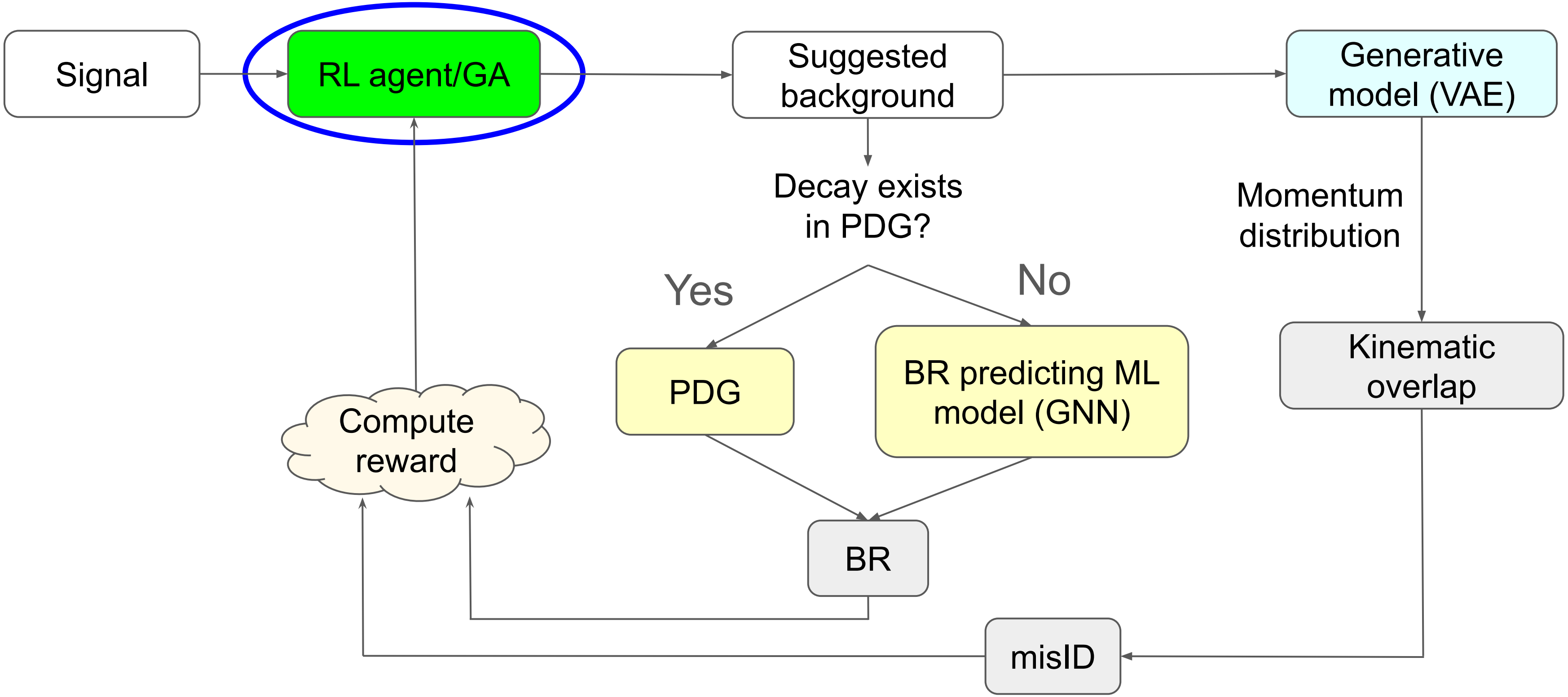
Radiative tail correctly modelled!

Reward

- **Reward** describes how problematic a background is w.r.t. a signal
- Currently using a toy model for the reward. In the future, ML models previously mentioned will provide input to the agent.



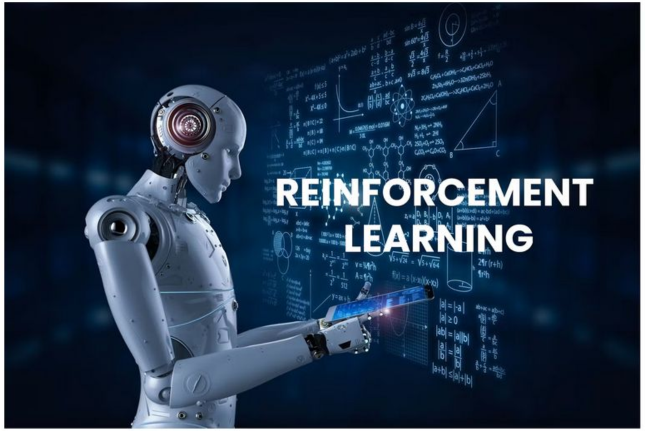
Workflow



Reward = BR x misID x Kinematic_overlap

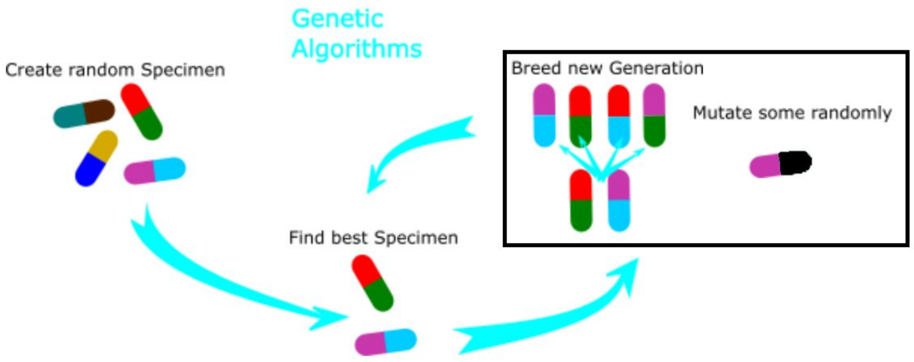
Uncovering of hidden backgrounds

- A nested **for loop** to iterate over the PDG recursively to build all possible decay chains?
  - Impractical. Need a smarter approach.
- **Genetic Algorithm (GA)** approach
- **Reinforcement Learning (RL)** approach
- GAs perform an efficient exploration while RL allows generalization



Background finder: GAs

- Genetic Algorithm (GA) approach:
  - **AI** technique to solve **optimization** problems. Inspired by evolution process and natural selection theory.
    - Population of individual solutions represented by its **genes**. Selection of best individuals. **Combination** and **mutation** to obtain offspring. Iterative process towards an optimal solution.
    - Advantages: Robustness with respect to local maxima or minima. Better results in problems that do not adapt properly to traditional optimization techniques
  - Goal: optimize a fitness function that describes how problematic a background is w.r.t. a signal
  - Individuals are backgrounds

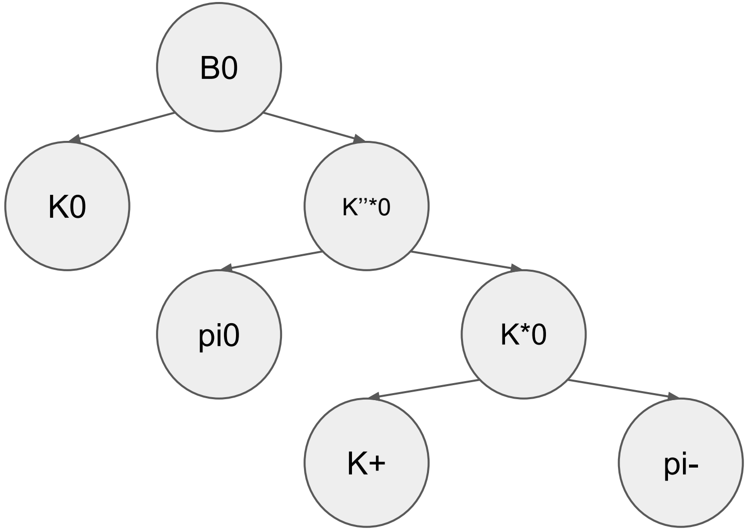


GAs for background finder

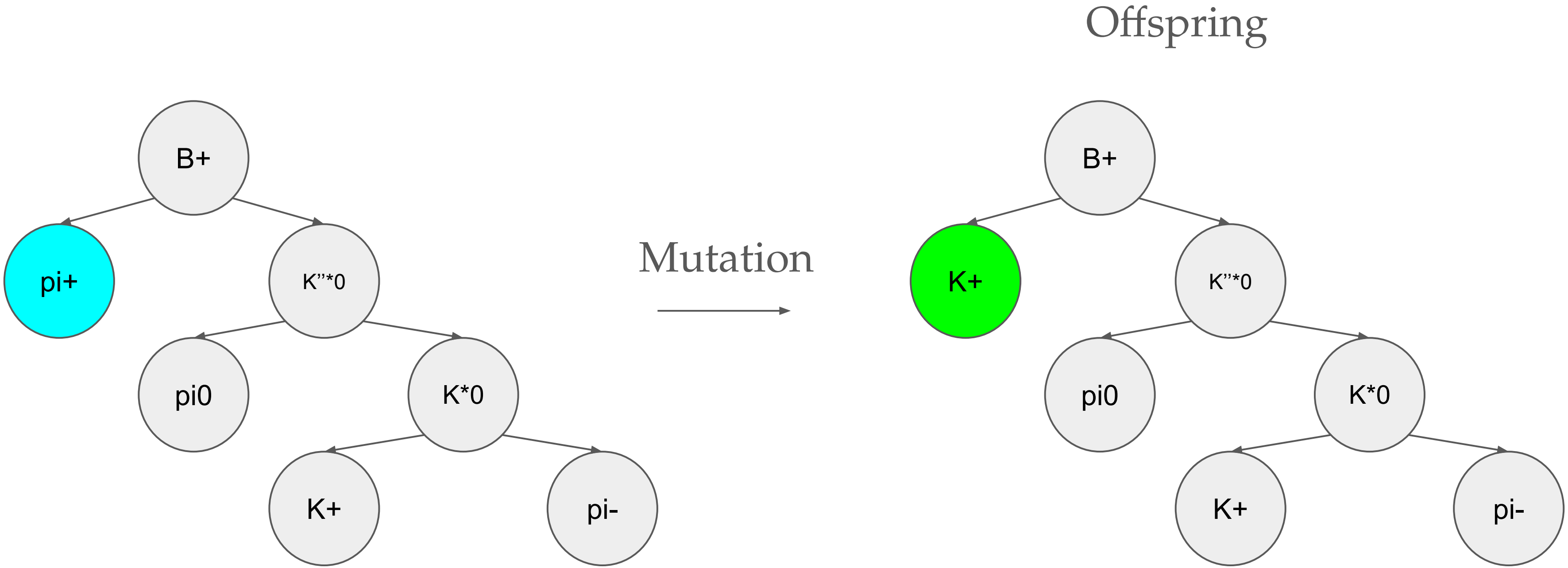
- **Genes** of individuals are represented by a **tree structure** that describes the decay chain
- Apart from the traditional **variation processes** (combination and mutation) we introduced custom variation processes:
  - To build **intermediate resonances**
  - To make the search more efficiently based on the **physical properties**
  - To be able to use **learnt information** in future optimization problems

Signal: $B^0 \rightarrow K^{*+}(K^+ \pi^0) \pi^-$

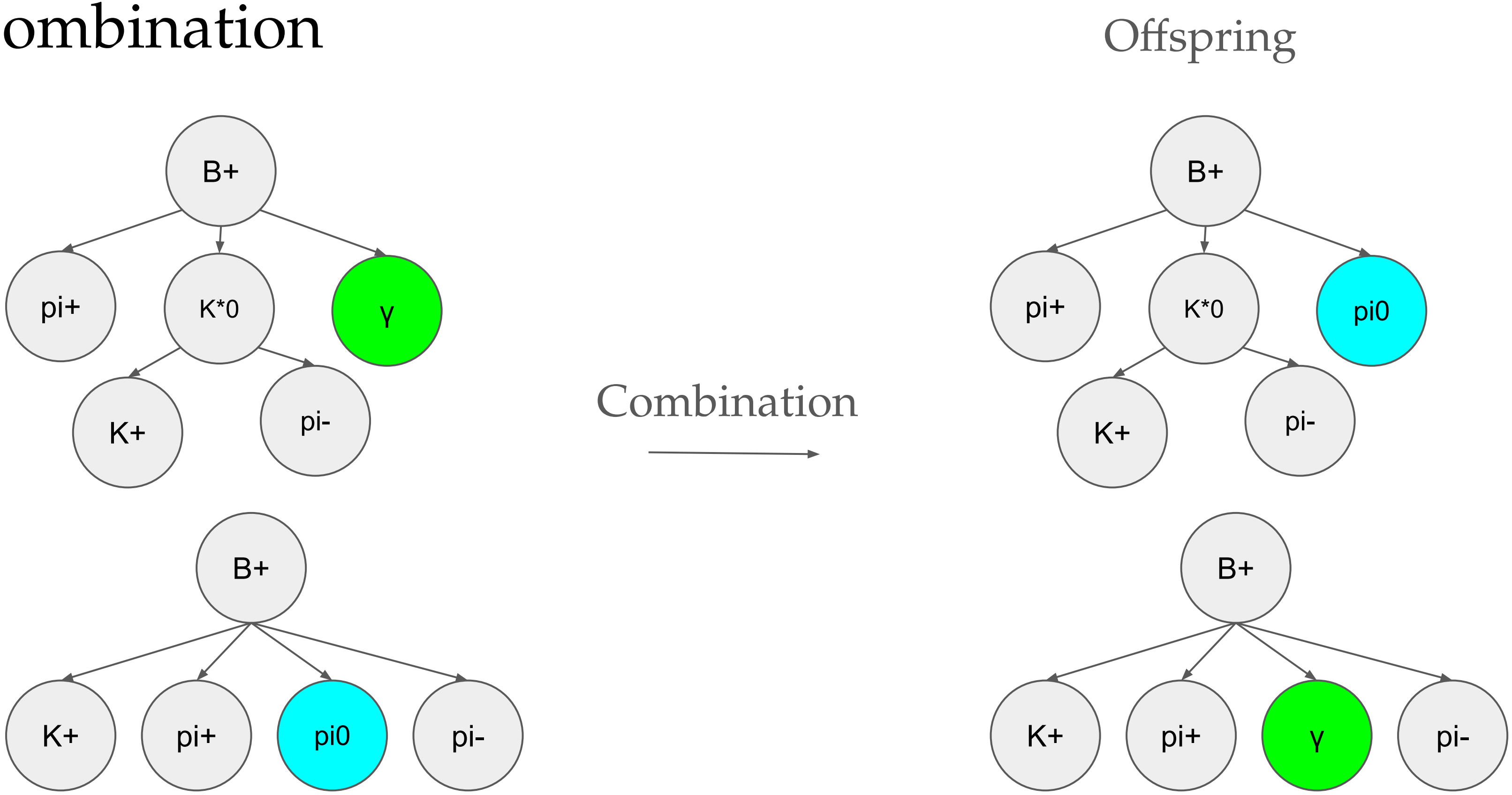
Background candidate:



Mutation

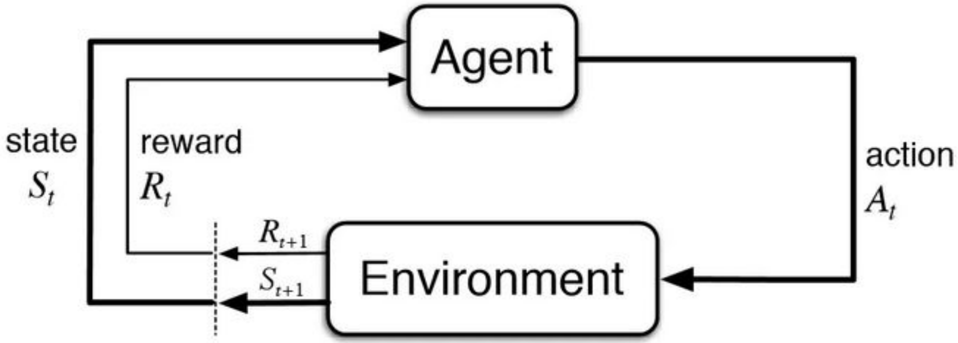


Combination



Background finder: RL

- Reinforcement Learning (**RL**) approach
  - RL is a type of ML where an **agent learns to make decisions** by performing actions in an environment to maximize cumulative rewards.
  - RL is suitable for exponentially growing spaces. Outperforms humans in games like chess and Go.
  - Goal: **train a ML model** that can successfully **predict** the most problematic backgrounds for new signals. Agent will learn the decay modes.



RL approach

Token: element of a sequence that needs to be converted into numerical data to provide input to a language model

- **State** will be described by a sequence of tokens
  - State is initialized with the information of the signal
- **Agent** will predict one token in each step, filling up the information of the background
- We can deal with **intermediate resonances**, **misidentifications** and **partially reconstructed backgrounds** with a few tokens.
- Example:

  - Signal: $B^0 \text{->} K^+ \pi^0 \pi^-$

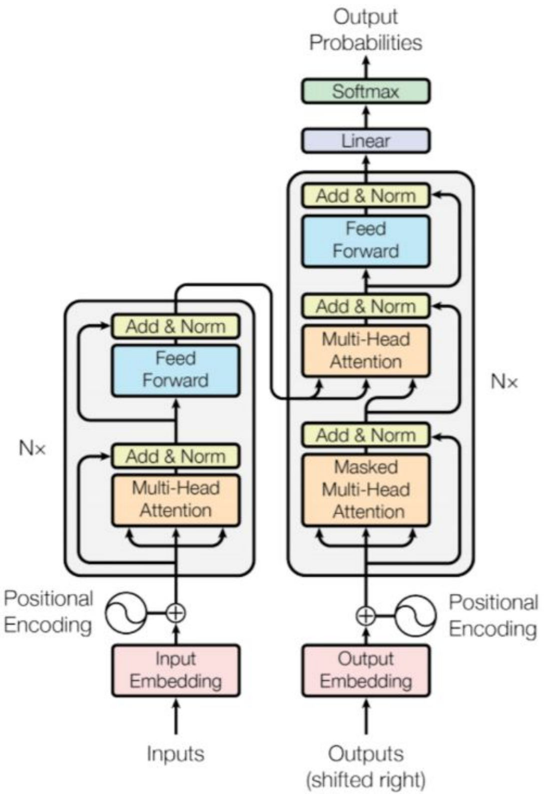
  K^- and π^- are misidentified

  - Background: $B^+ \text{->} K^{*+}(K^+ \pi^0) \pi^+ K^-$

  π^+ is not detected

RL strategy

- **Agent**: **transformers** are an ideal architecture for dealing with tokenized sequences
- **Action masking** is applied to mask the tokens that do not make sense for the current state
- **Pretraing of** the agent to improve learning
- Playing a complete game on a **GA suggestion** with the desired frequency during training of agent
    - **Guided search** to reach high rewards



Pretraining: example

- The agent can be pretrained in a supervised way to **output possible backgrounds** given a signal.
- This also makes the agent **learn how to build decay sequences** that make sense
- Example:
  - Signal:

$$B^{+} \rightarrow \pi^{+}\ \text{anti-}D^{0}(\ K^{+}\ \pi^{-}\ )$$

  - Background:

π^{0} not detected

$$B^{+} \rightarrow \pi^{+}\ \text{anti-}D^{0}(\ K^{+}\ \pi^{-}\ \pi^{0}\ )$$

Most problematic background correctly predicted

Experiments

- 50 signals analyzed
- Target: 4 most problematic backgrounds (hall of fame) of each signal
- Technique: Genetic Algorithms
  - Population of 1.000 individuals and evolution of 40 generations
- Results:

	Experiment 1	Experiment 2
Space size	≃ 1.000.000	≃ 300.000.000
Explored space	4.7%	0.16%
Found backgrounds	99.5%	90.5%

- Example:
  - Signal: $B^- \to e^-$ anti-ν_e $D^0(\pi^+ K^-$ pi$^0)$
  - Hall of fame (according to toy model reward):
    - $B^- \to e^-$ anti-ν_e $D^{*0}(\pi^0$ D0$(\pi^+ K^-))$
    - $B^- \to e^- \pi^0$ anti-ν_e $D^0(\pi^+ K^-)$
    - $B^- \to \mu^-$ anti-ν_μ $D^0(\pi^+ K^- \pi^0)$
    - $B^- \to \mu^-$ anti-ν_μ $D^{*0}(\pi^0 D^0(\pi^+ K^-))$

GAs are performing a very efficient search

Conclusions

- Goal:
  - Design an algorithm that can successfully find the most problematic backgrounds to:
    - **accelerate the workflow** in particle physics
    - **avoid human errors**.
- Results:
  - The **performance** of each of the models involved seems to be very **promising**
- **What is next?**
  - **Improve the performance** of each of the models as much as possible
  - RL/GAs:
    - Implement/improve variation processes to make the search more efficient
    - Increase gradually the complexity of the problem, and explore each of the approaches to identify their weak spots
    - **GAs and RL** can be **combined** to overcome these weak spots
      - GAs perform an efficient exploration while RL allows generalization
  - Wrap all 3 subprojects together in order to apply this tool to **real case scenarios**

Thank you very much for your attention