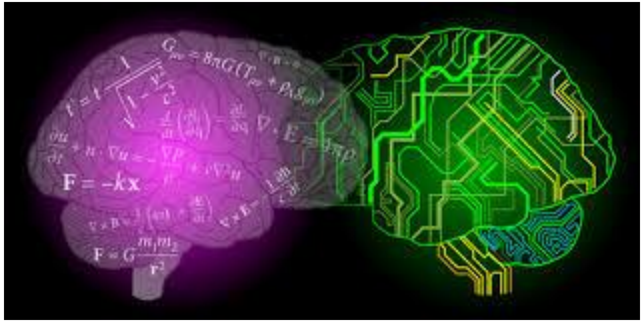
Playing the devil's advocate to bound hidden systematic uncertainties

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

20/07/2024

Overview

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



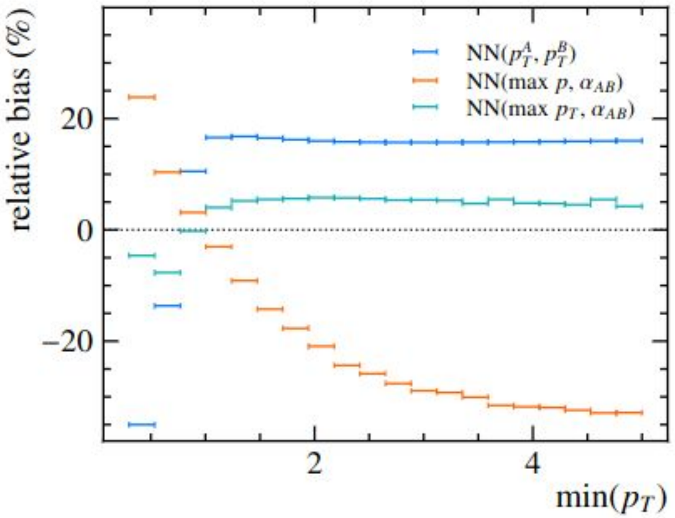
Motivation

- 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

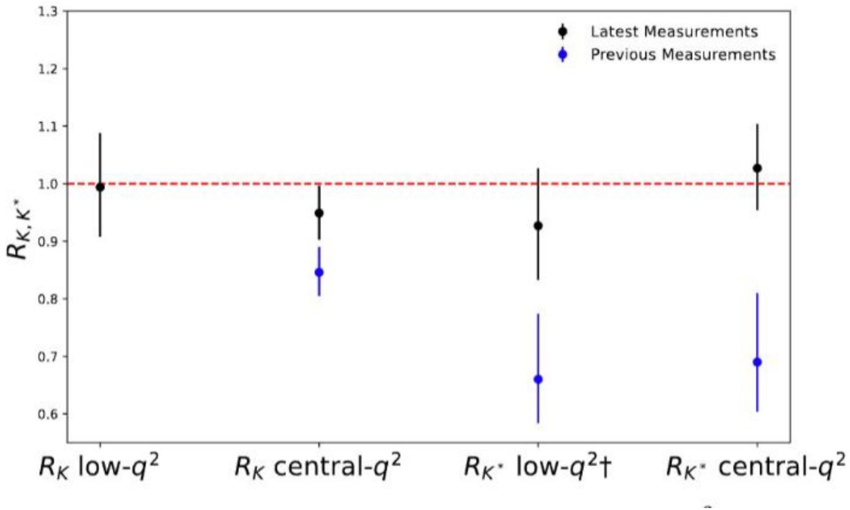


Efficiency mismodelling introduced up to a 30% relative bias (on differential Branching Ratio (BR))

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:

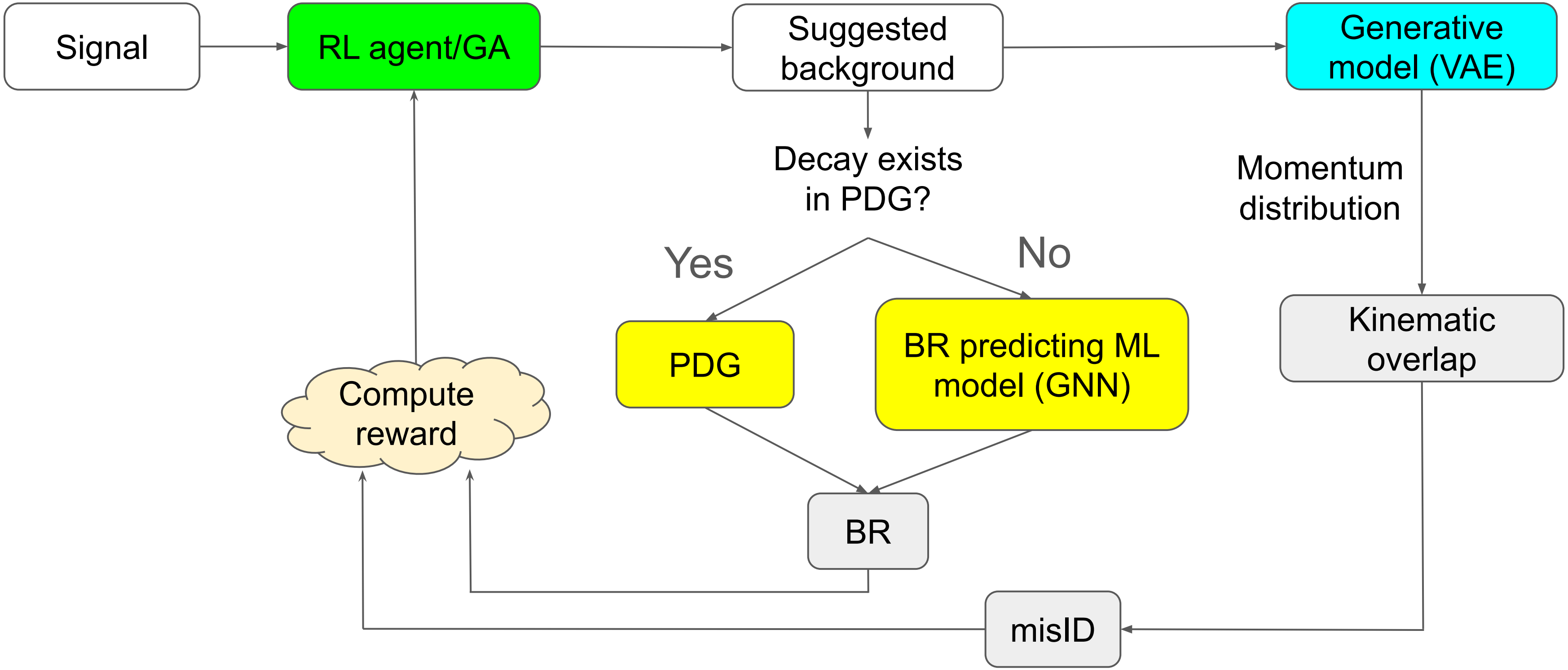
https://journals.aps.org/prd/abstract/10.1103/PhysRevD.108.032002



- 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**

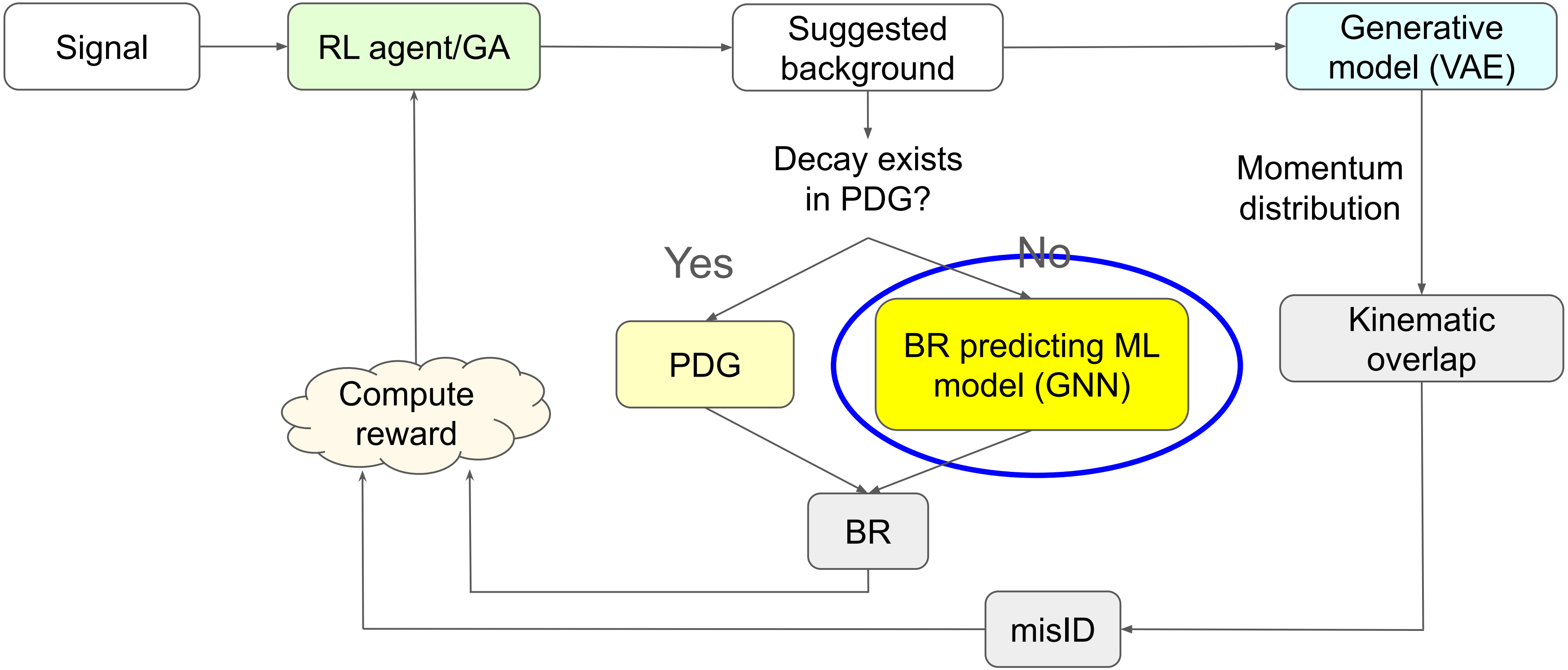
Workflow

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



Reward = BR x misID x Kinematic_overlap

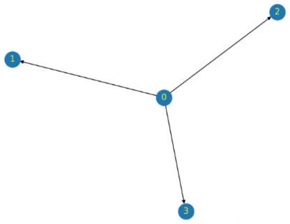
Workflow

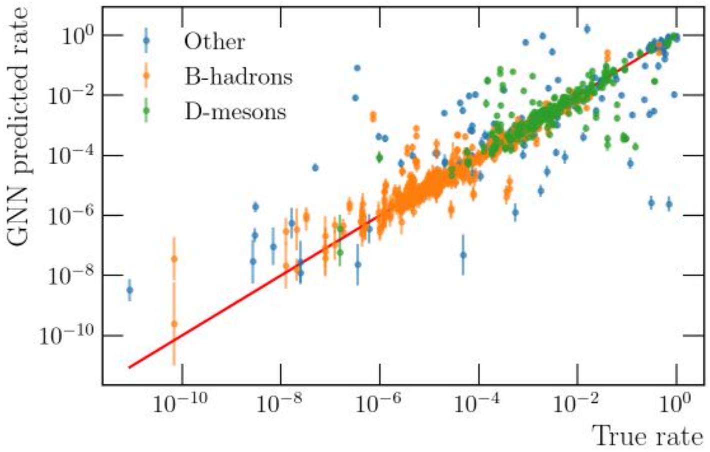


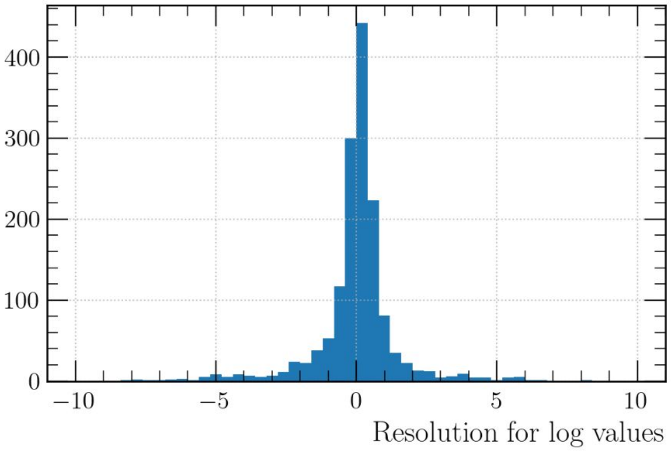
Reward = BR x misID x Kinematic_overlap

BR predictions

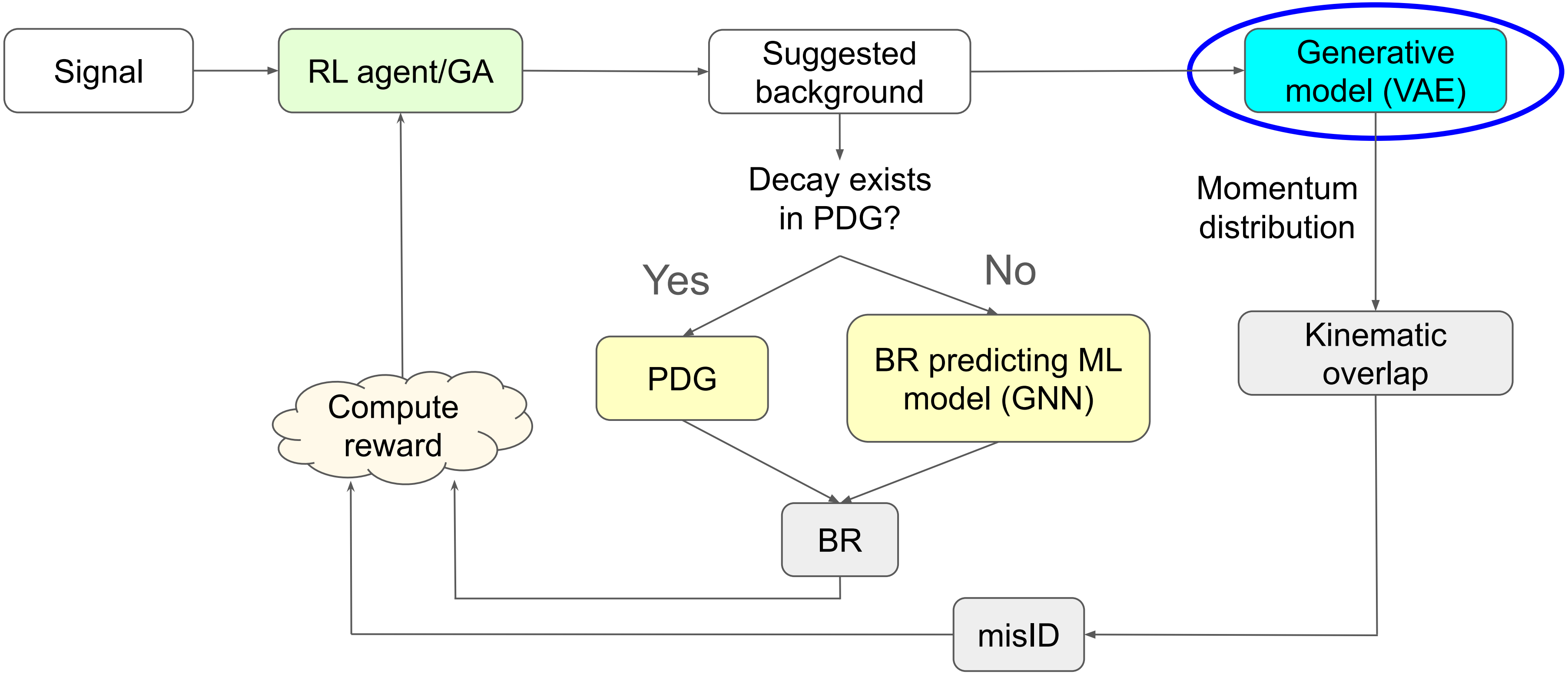
- **Branching Ratio** (BR) predictions for decays not present in the Particle Data Group (PDG)
  - Graph Neural Network (**GNN**) model trained on the PDG data
  - **Graphs** to represent decays
    - Graph is invariant under particle ordering
  - Using a Bayesian architecture to obtain uncertainty on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







Workflow



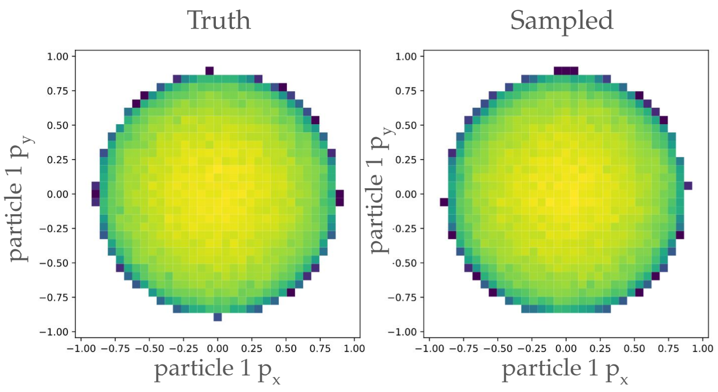
Reward = BR x misID x Kinematic_overlap

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

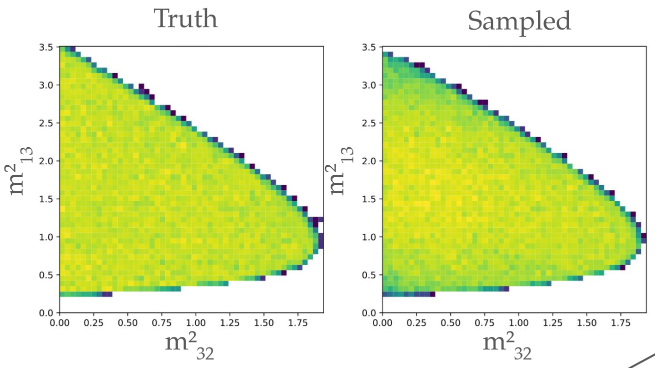
Variational Autoencoder (**VAE**) model

Momentum distribution

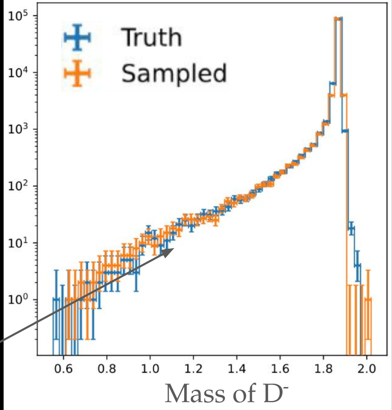


⇒

Dalitz plot



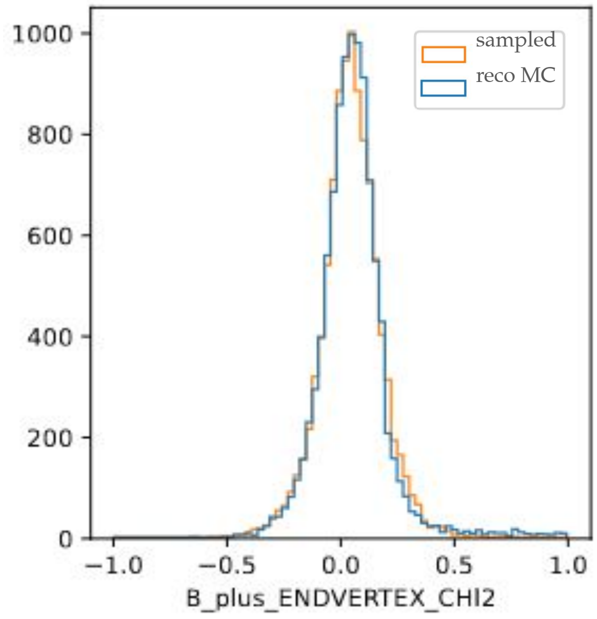
Invariant mass



Radiative tail correctly modelled!

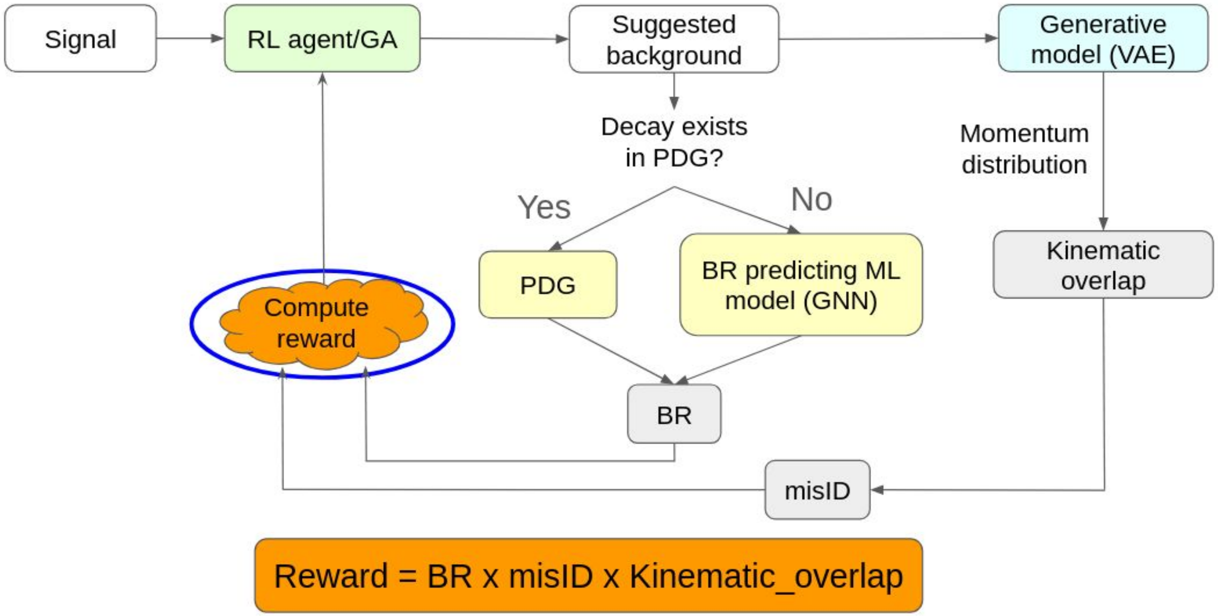
Generative model: reconstructed variables

- **Detector performance on reconstructed variables** can also be modelled with generative models
- Using reconstructed Monte Carlo as target
- Model: Wasserstein Generative Adversarial Network with Gradient Penalty (**WGAN-GP**)
- Example:
  - Distribution of B_plus_ENDVERTEX_CHI2 variable
  - Very good agreement between the sampled and the reco MC

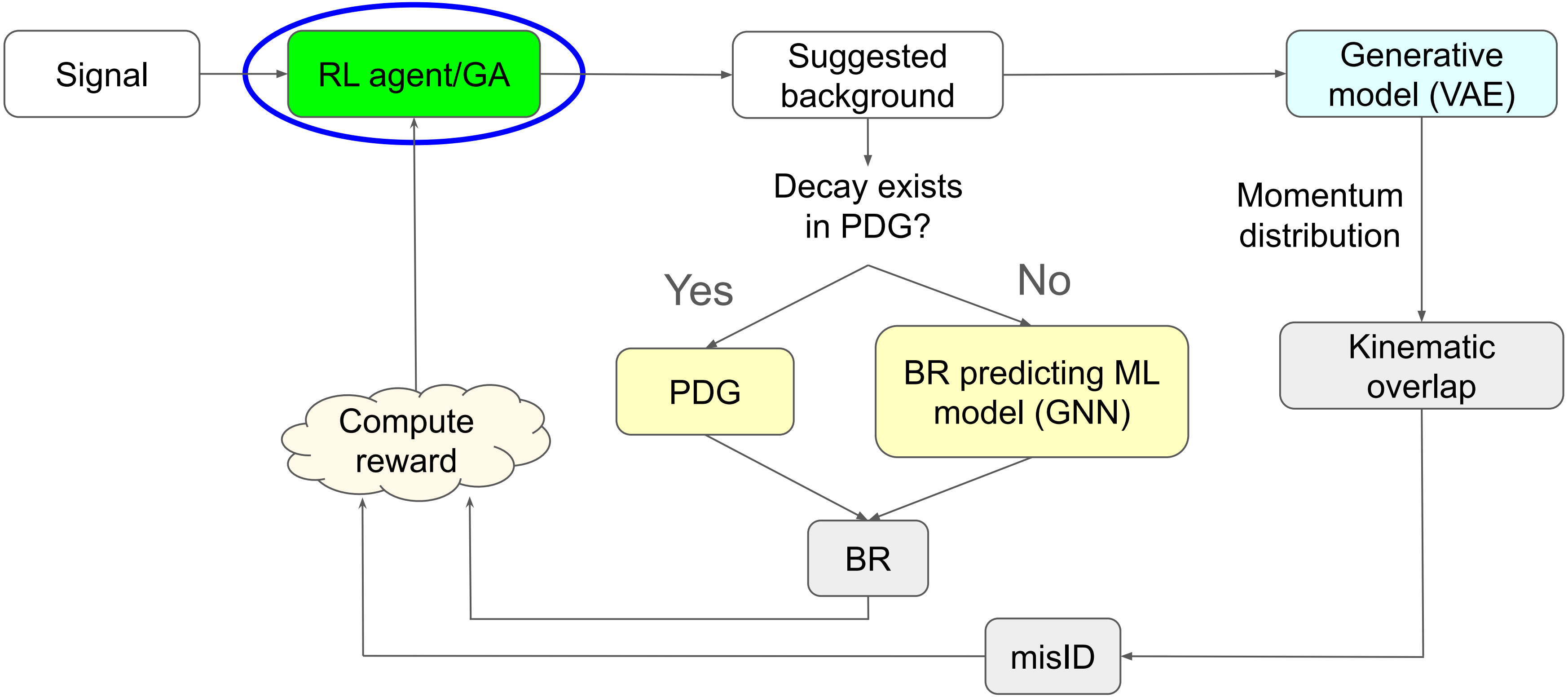


Reward

- **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.



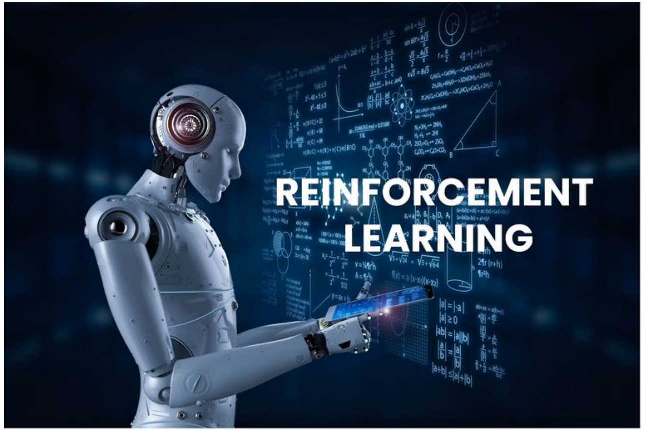
Workflow



Reward = BR x misID x Kinematic_overlap

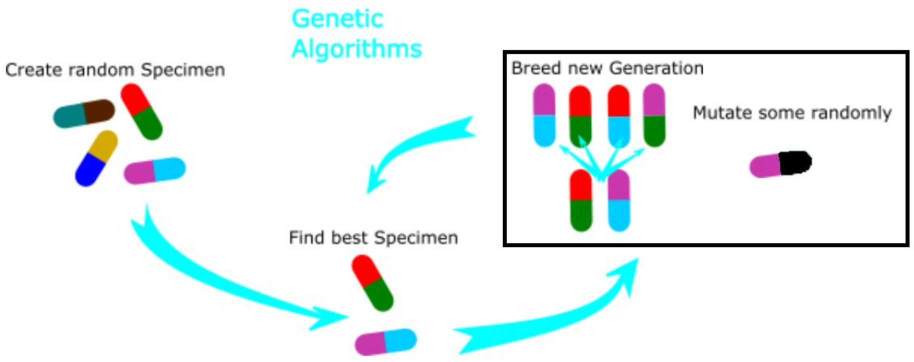
Uncovering of hidden backgrounds

- 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

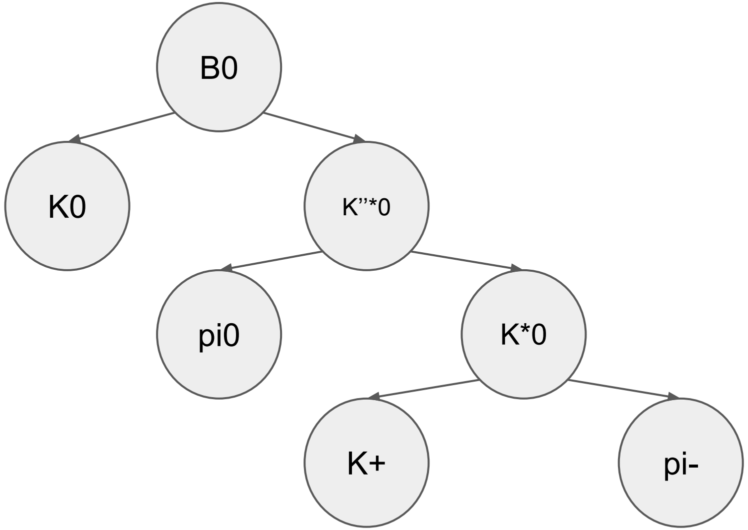


GAs for background finder

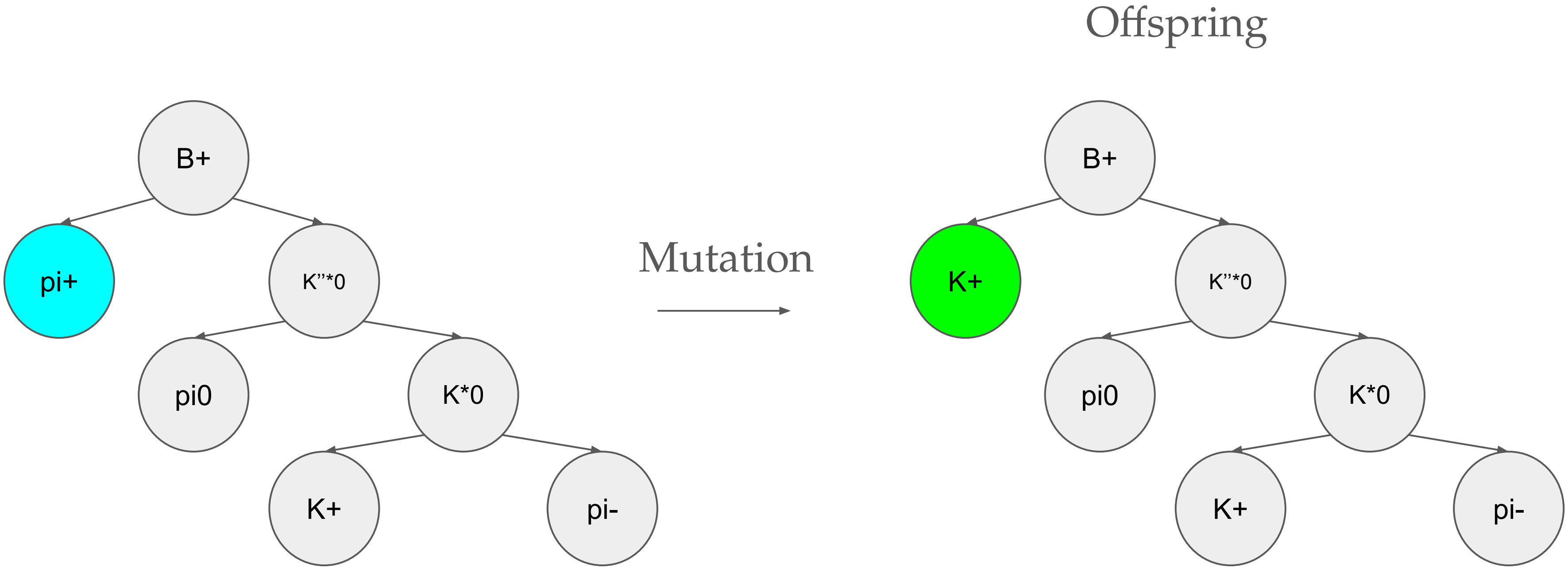
- **Genes** of individuals are represented by a **tree structure** that describes the decay chain
- Apart from the traditional **variation processes** (combination and mutation) we introduced custom variation processes:
  - To build **intermediate resonances**
  - To make the search more efficiently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

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

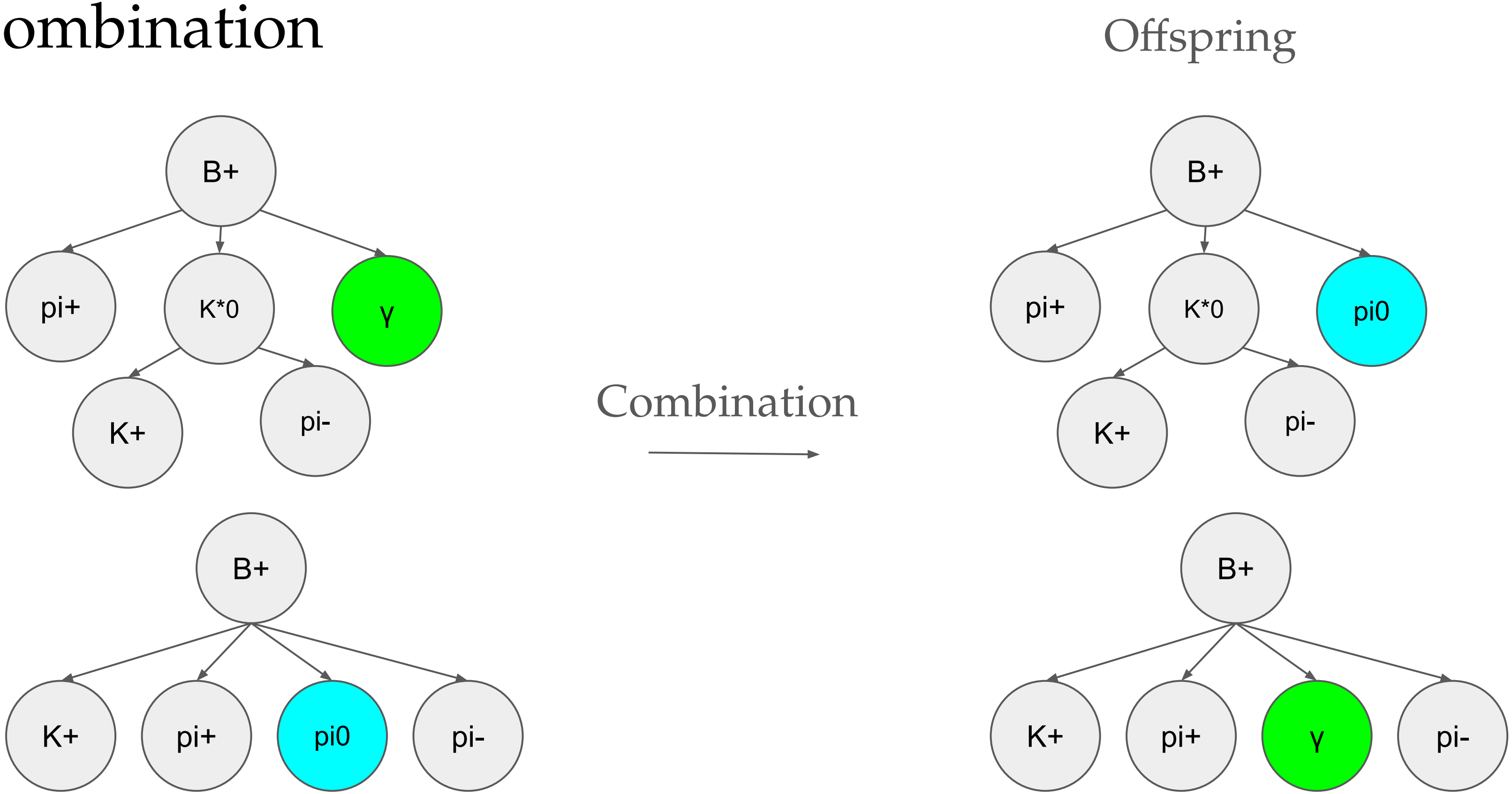
Background candidate:



Mutation

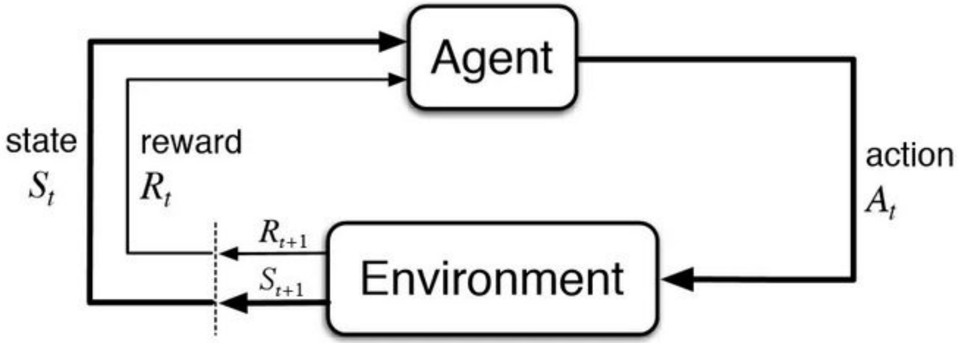


Combination



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: $B^0 \to K^+ \pi^0 \pi^-$

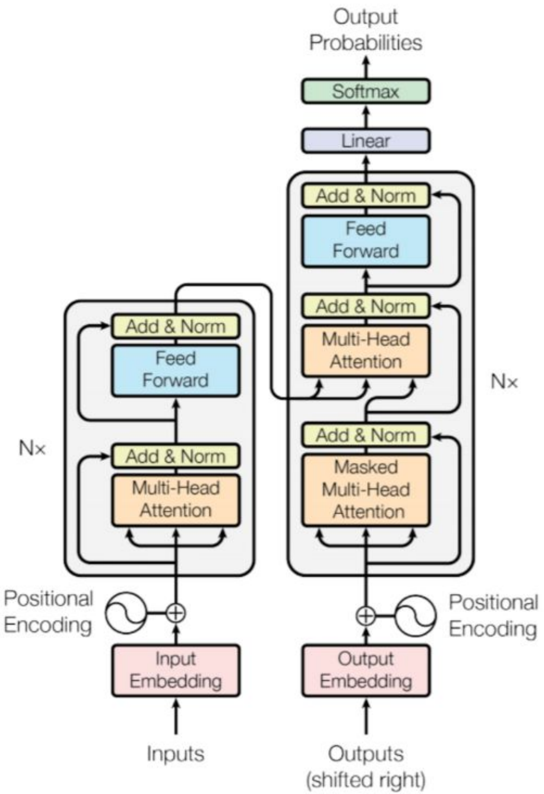
  K^- and π^- are misidentified

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

  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:

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

  - Background:

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

π^0 not detected

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%	0.04%
Found backgrounds	99.5%	71%

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

GAs are performing a very efficient search

Conclusions

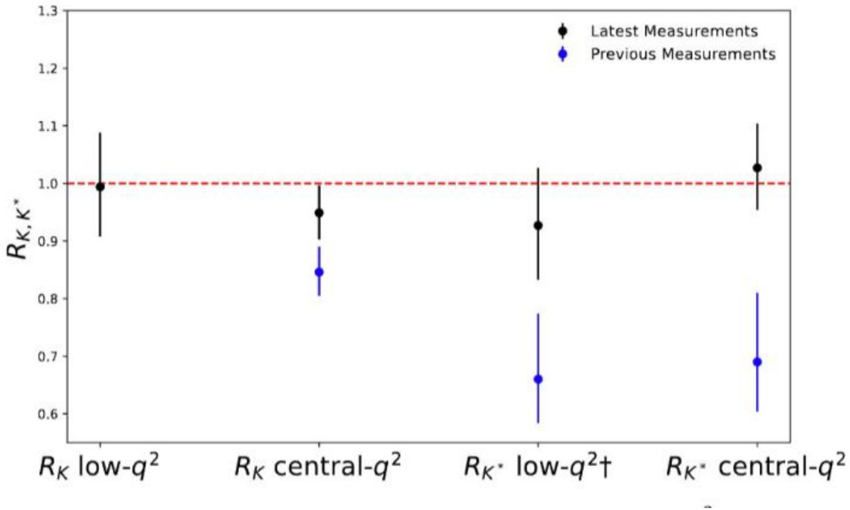
- Goal:
  - Design an algorithm that can successfully find the most problematic backgrounds to:
    - **accelerate the workflow** in particle physics
    - **avoid human errors**.
- Results:
  - The **performance** of each of the models involved seems to be very **promising**
- **What is next?**
  - **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

Auxiliar slides

Motivation

https://arxiv.org/pdf/2212.09153



- Lepton Flavour Universality (LFU) test initially showed discrepancies w.r.t. SM predictions
- Statistical fluctuations? Detector or physics mismodellings? New physics?

$$R_{K^{(*)}} = \frac{\mathcal{B}(B \to K^{(*)}\mu\mu)}{\mathcal{B}(B \to K^{(*)}ee)}$$

Reward/fitness function

- **Backgrounds** give a similar response to the signal in the detector
- The **reward/fitness function** describes how problematic a background is w.r.t. a signal
- The reward/fitness function takes into account several factors:
  - BR of the background: **Probability** to see this certain decay.
  - misID factor: Possible **misidentifications** of particles
  - PartReco factor: To deal with **partially reconstructed backgrounds** (some particles might not be detected)
- Currently a toy model for the reward/fitness function is being used:
  - BR is read from **decay.dec** file
  - Apply a penalty for every **misID**. Consider misIDs just between particles of same charge
  - Apply a penalty for every **not detected particle**. Number of detected particles in signal and background needs to be the same
- In the future, the ML models previously mentioned will provide input to the agent

Tokenization

- Tokens for representing particles
- Index tokens: {1, 2, …, N_S} → N_S tokens (N_S = number of final state particles in signal)
- Token for)
- Token for "END"
- Token for "LOST"
- Token for separating signal and background (not an action)
- Token to represent elements with a not assigned token (not an action)

"LOST" and "index tokens" are related to the next token in the sequence and allow to describe misID and Part. Reco.

Example:

K^- and π^- are misidentified

- Signal: $B^0 \text{->} K^+ \pi^- \pi^0$
- Background: $B^+ \text{->} K^{*+}($ 1 K^+ 3 π^0) LOST π^+ 2 K^- END

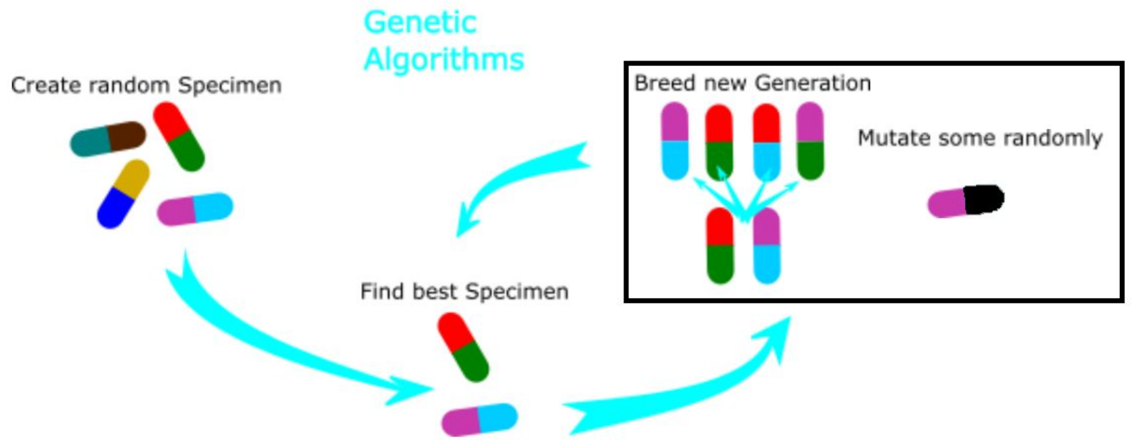
π^+ is not detected

Interesting points:

- We can deal with **intermediate resonances**, **misidentifications** and **partially reconstructed backgrounds** with a few tokens.
- Tokens (and thus actions in RL) are O(N). For example, actions like $pi^+ \text{<->} K^+$ would be $O(N^2)$.
- We can exploit a lot the **masking** of actions/tokens. For example, if last predicted token was "LOST", then we can mask all other actions that are not final state particles

Genetic Algorithms (GAs)

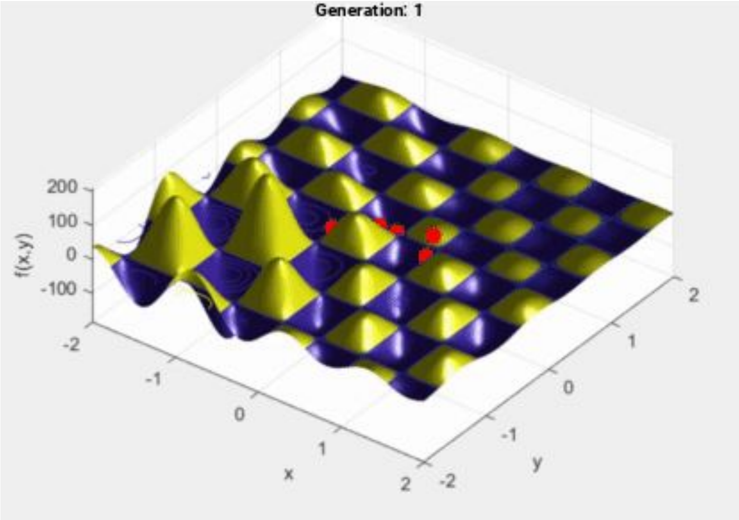
- **AI** technique to solve **optimization** problems
- Inspired by **evolution process** and **natural selection theory**
- Population of individual solutions represented by its **genes**. Selection of best individuals. **Combination** and **mutation**. Iterative process towards an optimal solution.
- Some advantages:
  - Robustness with respect to local maxima or minima
  - Better results in problems that do not adapt properly to traditional optimization techniques



Genetic Algorithms (GAs)

Example:

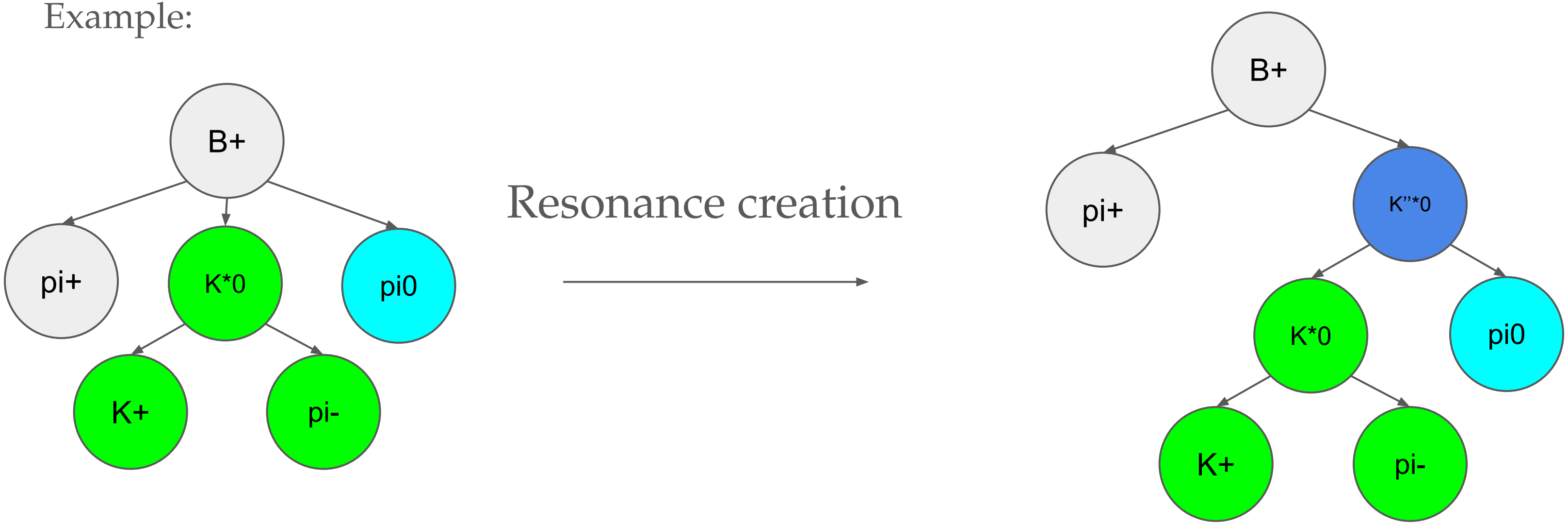
$$f(x,y) = \left(\sum_{n=1}^{5} n \ \cos[n + x(n+1)]\right)\left(\sum_{n=1}^{5} n \ \cos[n + y(n+1)]\right)$$



https://drive.google.com/file/d/1OWyJNH0tnq4um7fEkUvOwZIG2oyjckRg/view?usp=sharing

Resonance creation

- Some genes are randomly selected to create an intermediate resonance
- The charge of the intermediate particle is computed from the selected genes, and an intermediate particle of this charge is randomly suggested
- Example:



Arrival

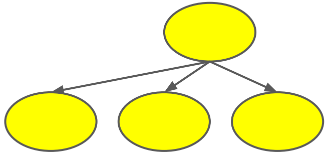
- Introduce into the population random individuals in each generation
  - The purpose is to be able to explore more, and to not depend that much on the initialization of the algorithm
  - Arrivals are only considered in case an individual did not suffer combinations, mutations or a resonance creation ⇒ We are not increasing the population size

Current population genes:

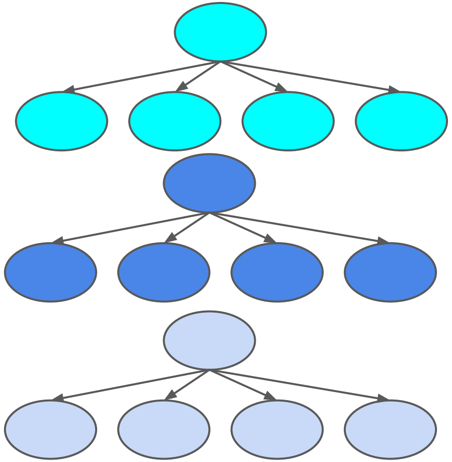
Inheriting from the signal

- With a certain probability the genes of the signal will be cloned instead of cloning an individual from the current population
- This allows to explore more the region of the space near the signal, which is more likely to have problematic backgrounds

Signal:

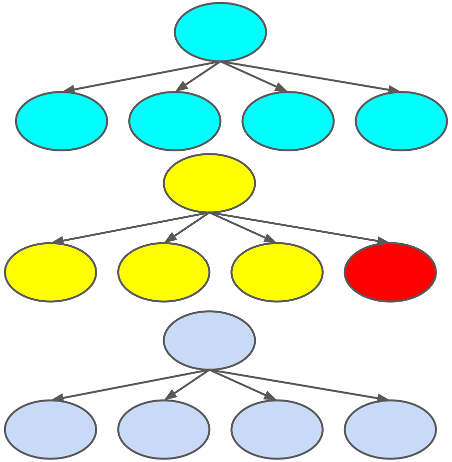


Current population:



⇒

Clone population:

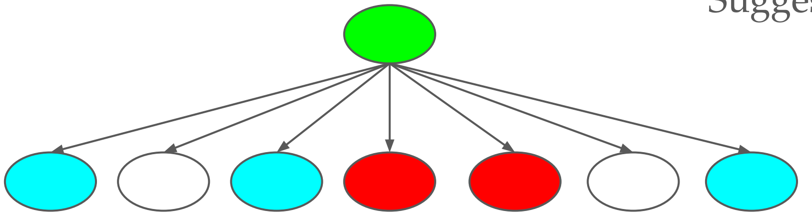


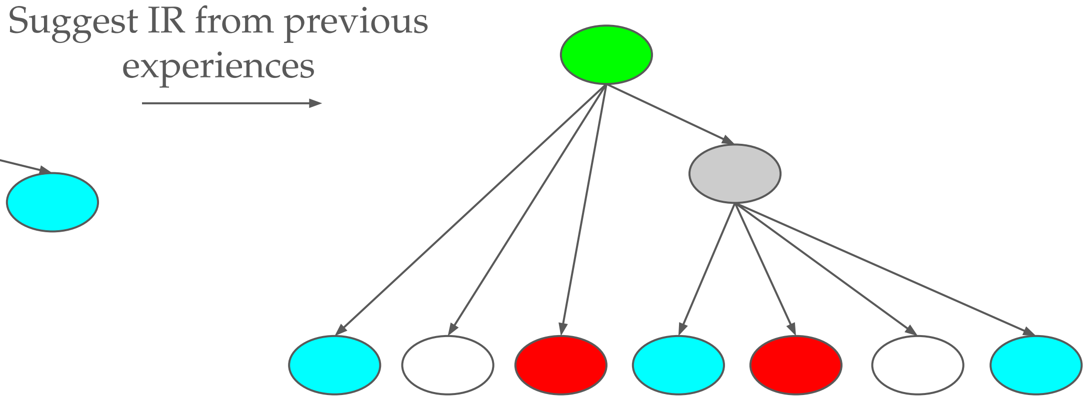
⇒

Apply variation processes to cloned population to obtain offspring

Previous experience learning

- When an individual has a non zero fitness value, and has an intermediate resonance in its genes that was never seen before, the GA “learns” this is a valid intermediate resonance for future games.
- With a certain probability, the genes of the individual are analyzed, and from all intermediate resonances that are possible for these genes one is randomly suggested





Green=mother
Blue=positive final state particle
Red=negative final state particle
White=neutral final state particle
Gray=intermediate particle

Experiment 1

- Description:
  - Set of particles: 4 mother particles, 10 intermediate particles, 18 daughter particles
  - Number of daughter particles in signal: 5
  - Hall of Fame size: 4
  - 2 intermediate resonances at most
  - 0 lost particles at most
  - Technique: Genetic Algorithms
    - Population of 1.000 individuals and evolution of 40 generations
- Results:
  - 50 signals were analyzed. Example: $B^- \rightarrow e^-$ anti-ν_e D^0(π^+ K^- pi^0)
    - **99.5% of backgrounds were found**
  - Space size:
    - Space size: 851.200
    - Number of individuals: 40.000
      - **4.7% of space explored at most**

Experiment 2

- Description:
  - Set of particles: 4 mother particles, ~~10~~ **18** intermediate particles, 18 daughter particles
  - Number of daughter particles in signal: 5
  - Hall of Fame size: 4
  - 2 intermediate resonances at most
  - ~~0~~ **1** lost particles at most
  - Technique: Genetic Algorithms
    - Population of 1.000 individuals and evolution of 40 generations
- Results:
  - 50 signals were analyzed. Example: $B^- \rightarrow e^-$ anti-ν_e $D^0(\pi^+ K^-$ pi$^0)$
    - **71% of backgrounds were found**
  - Space size:
    - Space size: ≃ 310.000.000
    - Number of individuals: 120.000
      - **0.04% of space explored at most**