

# Systematics framework

[Repository Documentation](#)

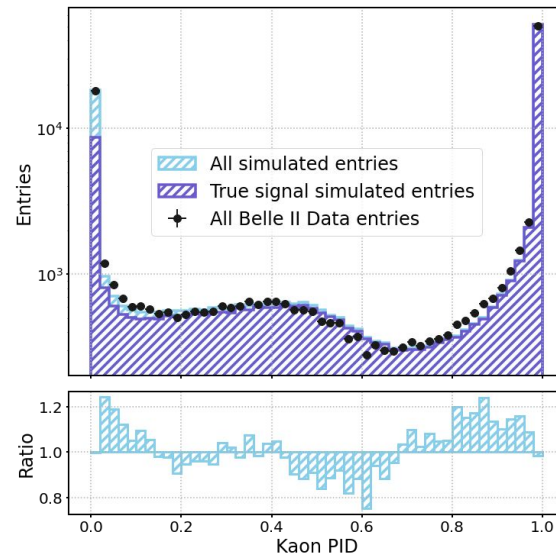
US Belle II Summer Workshop  
2024/06/21

Sviatoslav BILOKIN  
*LMU München*




# Introduction

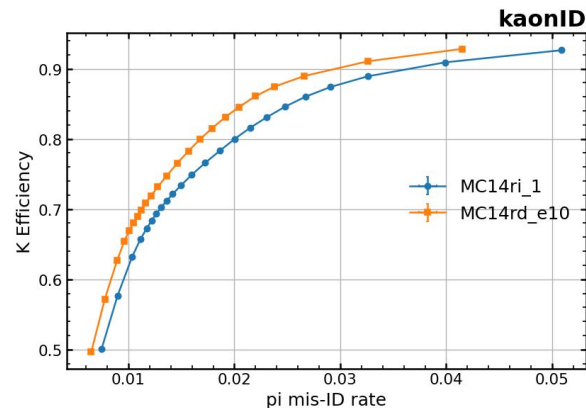
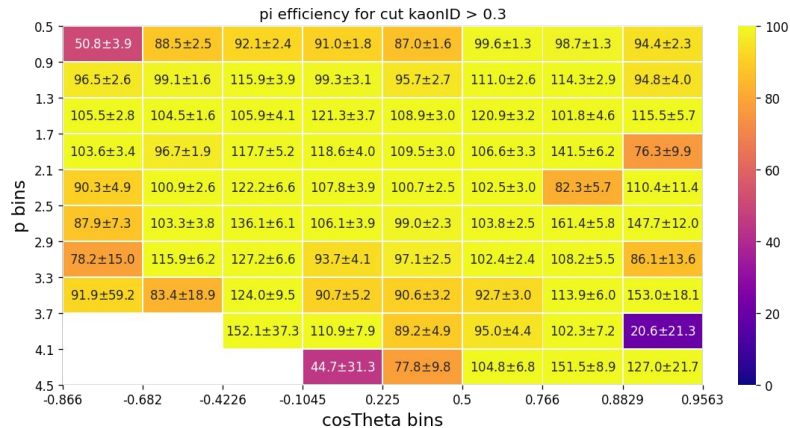
- Belle II simulation is excellent, but not perfect
  - Any selection cuts that are done on MC require data/MC corrections and assignment of systematic uncertainty
- We have developed a framework that unifies and automates the computation of data/MC corrections and systematics
  - Largely inspired by LHCb's [PIDCalib framework](#).
- Our framework has lots of physics modes for systematics
  - Modes for HadronID are fully integrated
  - Other modes are available as ntuples
- Main challenge is background subtraction on real data
  - Fit mass distributions for each weighted performance mode and compute *sWeights* and signal-like histograms on *data*
  - Compute efficiencies, ROC curves, and data/MC weights, etc.
- Framework is located on KEKCC and NAF



- $D^{*+} \rightarrow [D^0 \rightarrow K^- \pi^+] \pi^+$  for K/ $\pi$  ID
- $\Lambda^0 \rightarrow p \pi^-$  for p/ $\pi$  ID, nID
- $K_S \rightarrow \pi^+ \pi^-$  for  $\pi$  ID
- $[\tau \rightarrow 3 \pi \nu] [\tau \rightarrow l \nu \nu]$  for lepton ID
- $e^+ e^- \rightarrow \mu \mu \gamma$  for  $\mu$  ID
- $e^+ e^- \rightarrow \gamma \gamma$  for lepton ID
- $J/\psi \rightarrow l^+ l^-$  for leptonID
- $D^{*+} \rightarrow [D^0 \rightarrow K^- \pi^+ \pi^0] \pi^+$  for  $\pi^0$

# User scripts

- The script folder:
  - systematic\_corrections\_framework/scripts/
- Physics analysis scripts:
  - Data/MC weights `weight_table.py`
  - Efficiency calibration `calibrate_efficiency.py`
  - KDE PID fit `perform_pid_fit.py`
- Performance and validation scripts:
  - Efficiency table `efficiency_table.py`
  - ROC curve `id_vs_misid_curve.py`
  - Histogram `plot_distribution.py`
- Support scripts:
  - Show ntuple DB content `show_db_content.py`
  - Print variables `show_variables.py`
  - Print collections `show_collections.py`
- User scripts can be launched in the following ways:
  - Command line
  - **In the IPython mode**  Useful for workflow management, e.g. snakemake, b2luigi



# Tutorial

- The tutorial will be shown on the  $B^+ \rightarrow K^+ J/\psi$  where  $J/\psi \rightarrow \mu^+ \mu^-$ 
  - Nice illustration of leptonID and hadronID systematics
- Prerequisites:
  - KEKCC access
  - [Port forwarding](#)
  - Setup light-2405-quaxo release
  - Start [Jupyter server](#)
  - Copy the following notebook into your home folder:  
`/group/belle2/dataproduct/Systematics/examples/tutorials/US_BelleII_June2024/tutorial.ipynb`

# Cheatsheet

- Documentation: [link](#)
- Confluence: [link](#)
- Git repository: [link](#)

- **Ntuples:**

- KEKCC: `/group/belle2/dataproduct/Systematics/production/`
- NAF: `/nfs/dust/belle2/group/dataproduct/Systematics/production/`

- **Ntuple structure:**

- Follows the `vu.create_aliases_for_selected()` pattern

- Available dataset collections can be shown by `scripts/show_collections.py`:

- Available variables can be shown by `scripts/show_variables.py`:

- Now includes Charged BDT, kaonIDNN and pionIDNN and track isolation variables

SysCorrFW 0.8.0  
documentation

Contents:

- User scripts
- PID studies
- Variables
- Installation
- Tutorials
- Development
- Available `b2util` task classes
- Available `basf2` modules
- Other ParticleID modules
- Changelog

## Systematic Corrections Framework

Note

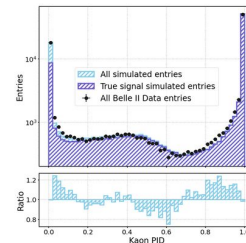
The source code repository has been moved to [DESY GitLab!](#)

Differences between simulation and experimental data are one of the main components of a typical systematic uncertainty of any physics analysis. For example, selection of charged particles by particle identification (PID) will result in different selection efficiencies on the simulation and experimental data samples in most of the cases, the differences are demonstrated in the figure below. To reduce the systematic uncertainty, one can transform the simulated PID distributions or compute the efficiencies directly on data, which is the main purpose of this framework.

Contents:

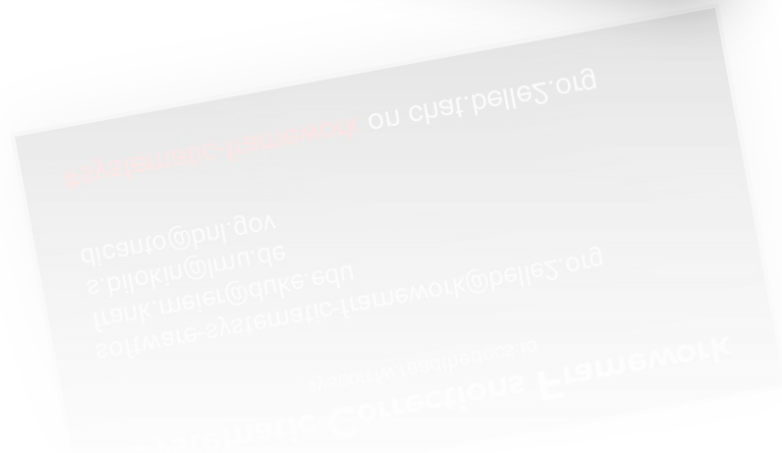
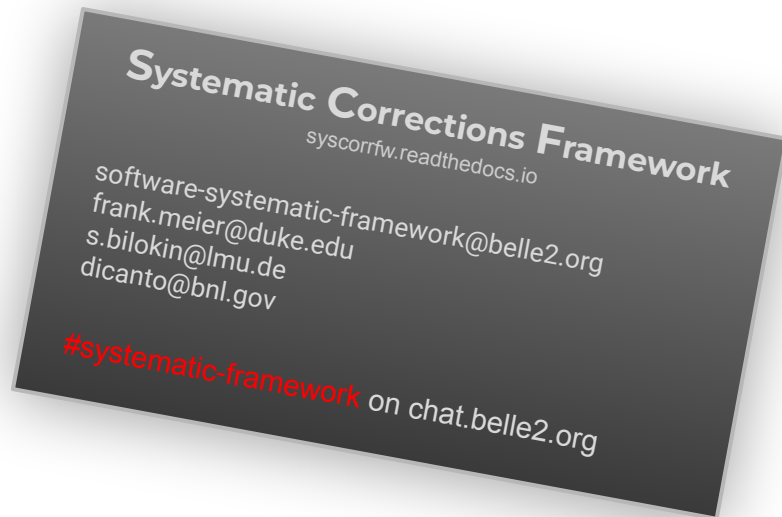
- [User scripts](#)
- [PID studies](#)
- [Variables](#)
- [Installation](#)
- [Tutorials](#)
- [Development](#)
- [Available `b2util` task classes](#)
- [Available `basf2` modules](#)
- [Other ParticleID modules](#)
- [Changelog](#)

### Basics



# Summary

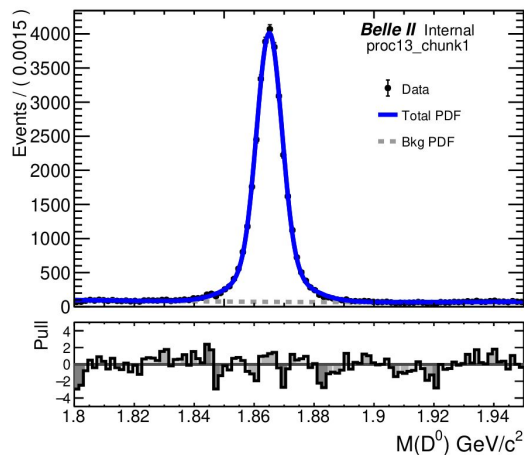
- This framework is designed to automate the performance studies and computation of systematic weights associated to PID
- Progress so far:
  - ✓ Added a possibility to integrate other types of studies
  - ✓ Processed proc13 + all buckets and MC
  - ✓ Introduced PID KDE Fit in LHCb fashion
  - ✓ Duplicate dataset to other servers, i.e. DESY
  - ✓ Create meta-variables in basf2 fashion
  - ✓ Migration to GitLab
  - ✓ Remove nCDCHits cut from HadronID modes
  - ✓ Integration tests
  - Integration of Lepton ID modes is in progress
    - Integration with B2Production framework
    - Integration with basf2 and b2conditiondb



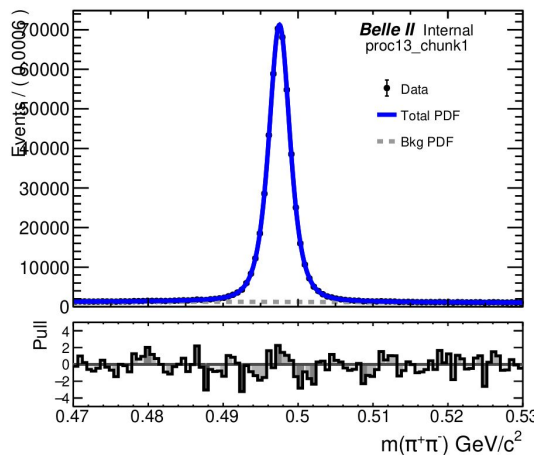
Thank you!

# Status of integrated studies

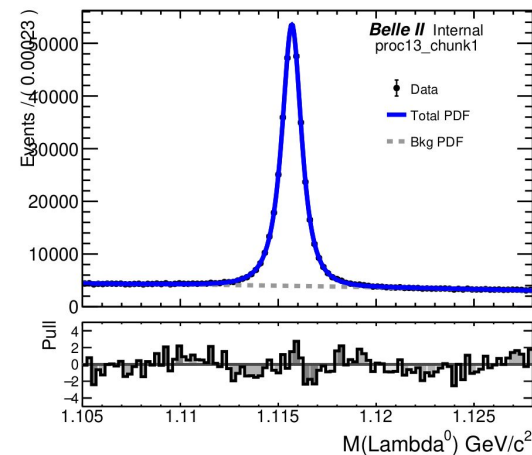
$$D^{*+} \rightarrow [D^0 \rightarrow K^- \pi^+] \pi^+$$



$$K_S \rightarrow \pi^+ \pi^-$$



$$\Lambda^0 \rightarrow p^+ \pi^-$$

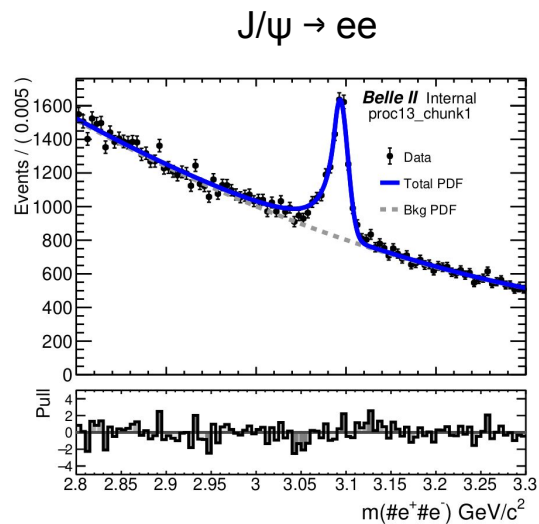
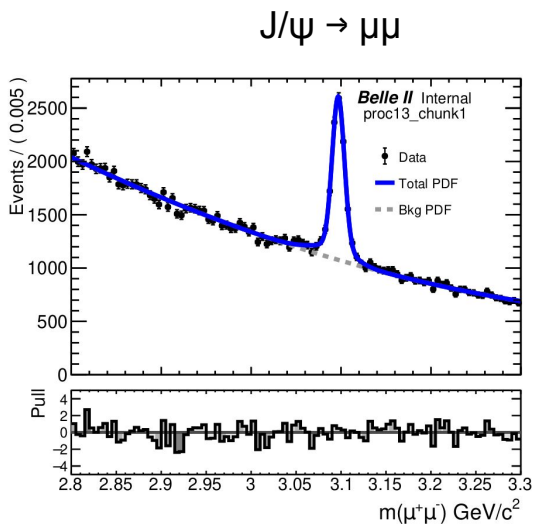


- Mass fits for HadronID studies have been significantly improved
  - Reduced low-multiplicity background and improved signal shape
- The mass fit is used to produce sWeights for background subtraction
- Lambda0 study can be used for nbarID systematics
- Recent changes:
  - Removed old Hadron ID cut of  $n\text{CDCHits} > 20$  for leptonID fake rates

NEW

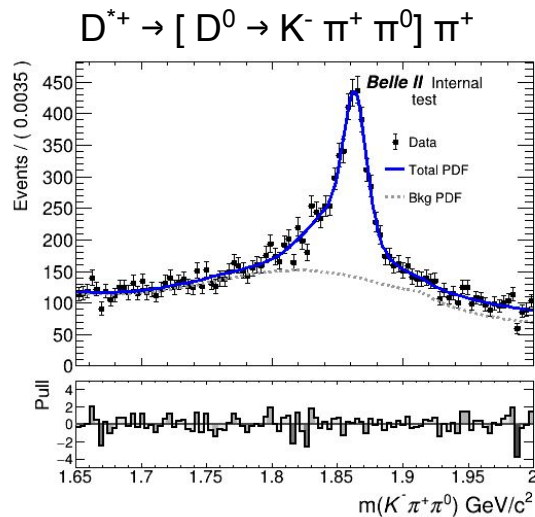


# Status of integrated studies



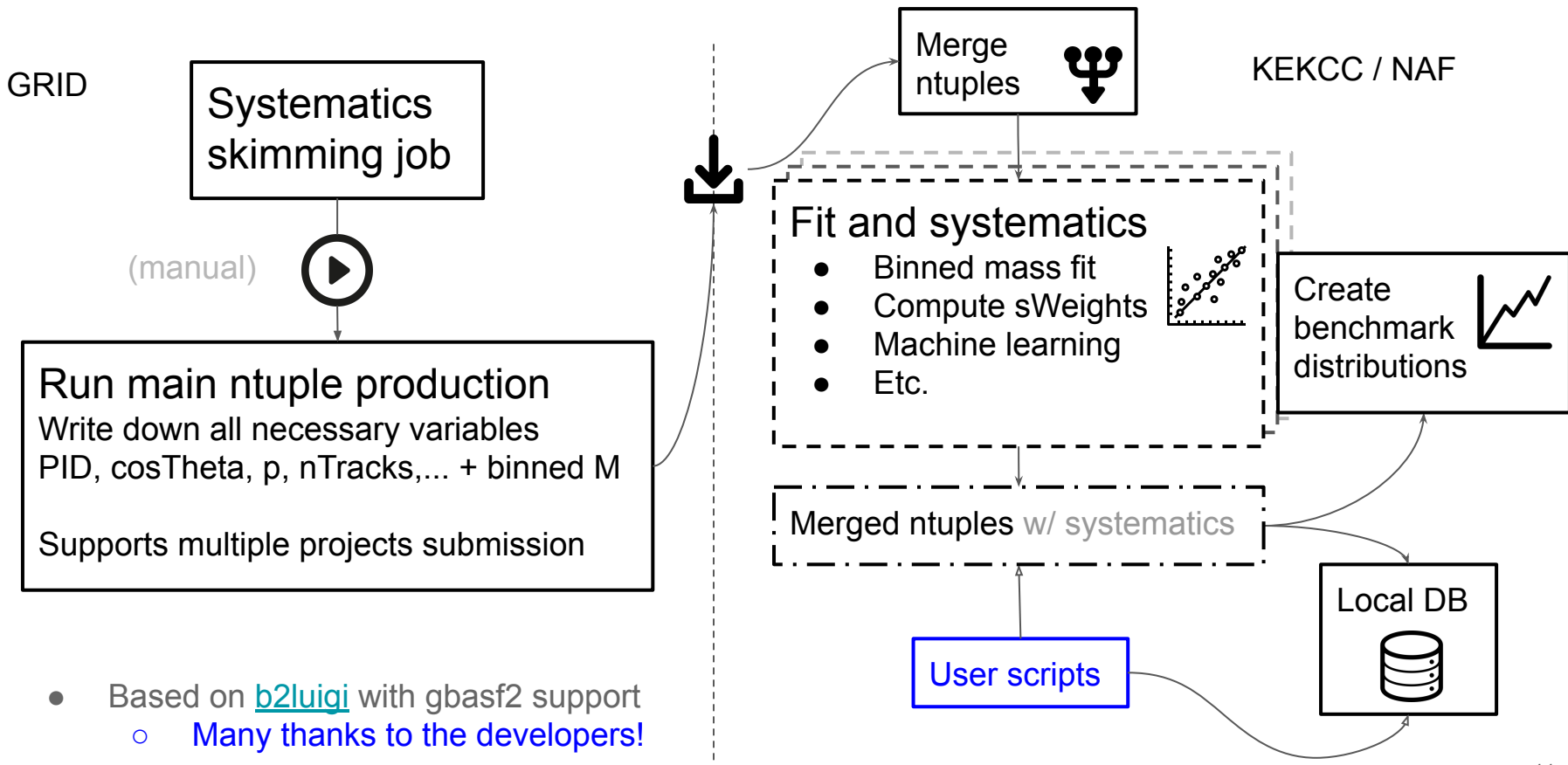
- Mass fits for HadronID studies have been significantly improved
  - Reduced low-multiplicity background and improved signal shape
- The mass fit is used to produce sWeights for background subtraction
- In development:
  - sWeights are not valid for  $p < 1.5$  GeV/c because of correlations in the background, more investigation needed

# Status of integrated studies



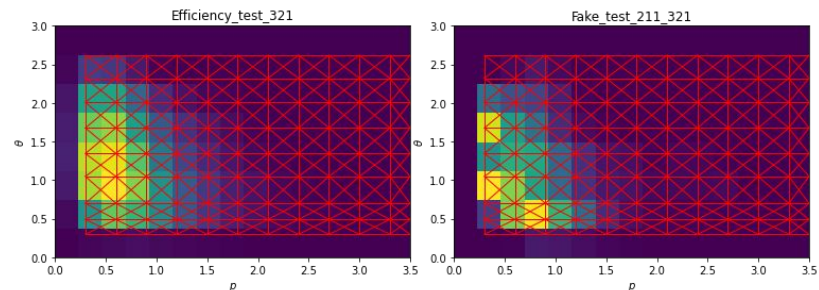
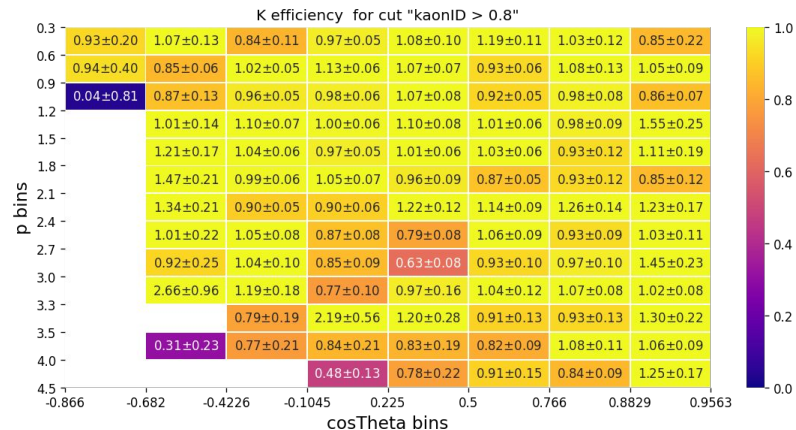
- This study is targeting  $\pi^0$  selection efficiency systematics
- **Ntuple production has been successfully integrated**
  - ECL Cluster variables have been added to support custom MVA
- In development:
  - sWeights are not valid for  $p(\pi^0) < 1.5 \text{ GeV}/c$  because of extremely steep background
  - Measurement of sWeight biases is ongoing

# Basic workflow



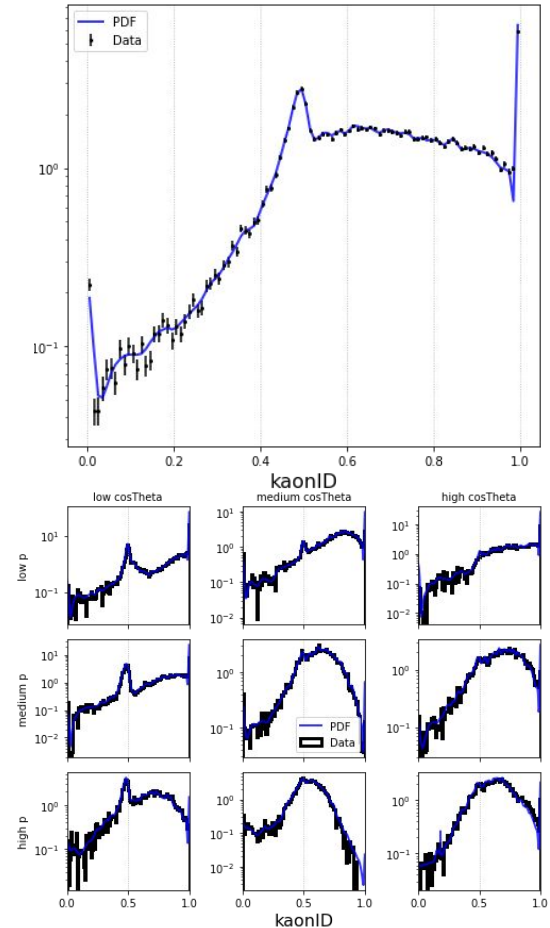
# Computation of Data/MC weights

- One can produce the weights in a notebook on KEKCC / NAF and immediately apply them to MC ntuple
  - Select dataset collections
    - “proc13+prompt” vs “MC15rd\_proc13+prompt”
  - Compute data/MC ratios and convert into weight table format via `create_weights()`
- The application of the weights is done using PIDVar
  - Requires efficiency and fake rate weights
  - Provides uncertainties on MC ntuples
- The weight tables from the framework need to be adjusted for PIDVar
  - E.g. renaming columns, converting column values, etc.
  - In the future PIDVar will be integrated into basf2
    - Adapt it for FEI,  $\pi^0$  systematics
- [Tutorial is online](#)



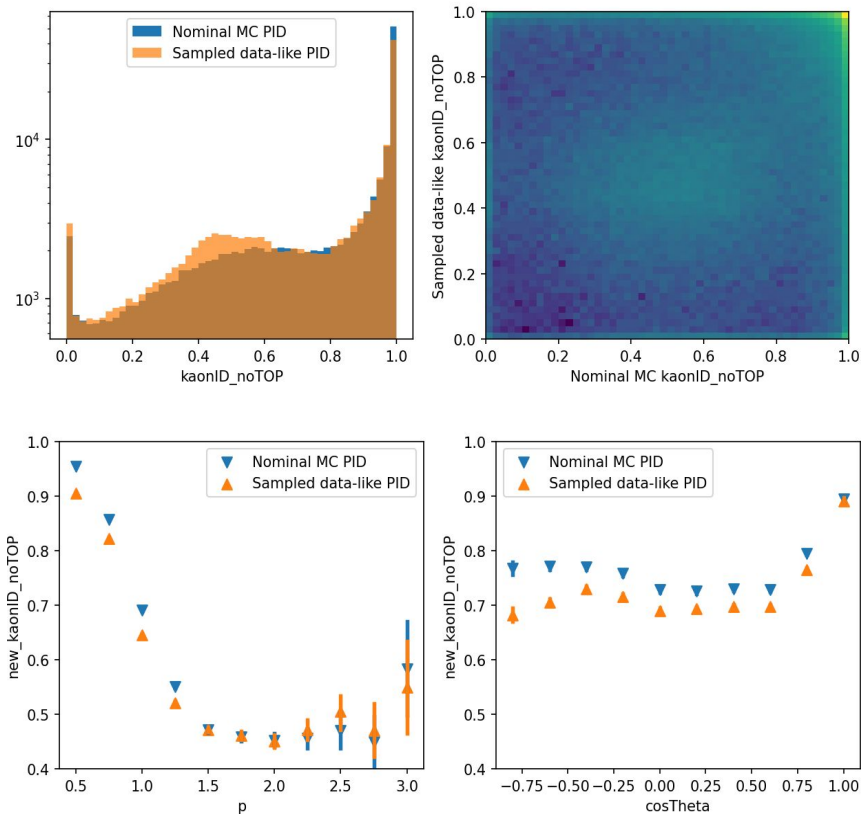
# Multidimensional PID Fit

- An alternative to the reweighting of the histograms, which was first implemented in LHCb
- One can fit real data PID as function of  $(p, \cos\theta, \dots)$  and sample from it using MC  $(p, \cos\theta, \dots)$ 
  - Sampled variable on MC would be a data-like PID
  - Free choice of cut values by the analysts
  - Much more granularity than in the weight tables
    - e.g. 200 bins for PID, 100 bins for  $(p, \cos\theta, \dots)$
  - Sampled variable can be used in ML algorithms
  - Relies on weights for background subtraction and systematics
- Implemented using LHCb [KDE Meerkat](#) library
- Free choice of the fit dimensionality and the variables
- The output of the script is saved as ROOT file
- [Tutorial is online](#)
- Further development:
  - Define the optimal fit parameters
  - Upload output ROOT files to CDB



# PID resampling

- The ROOT files from the PID KDE fit are used to sample the data-like PID on MC
- The resampled PID on MC ntuples will have data-like efficiency and fake rates as well as realistic profiles
- Can be used in MVA for training if the other variables are **not correlated to the original MC PID**
- [Tutorial is online](#)
- Systematic uncertainties can be estimated using an alternative sWeights column

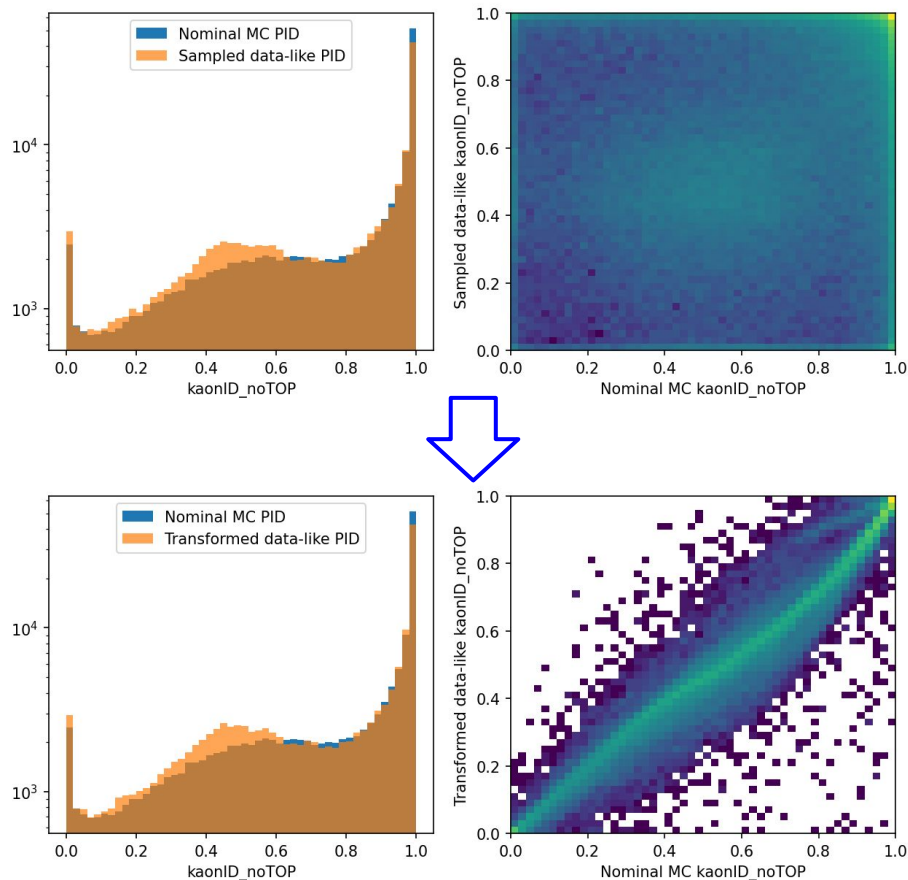


# PID transform

- The resampling algorithm produces new PID variable which is independent from the original MC PID
- Transform algorithm has the same advantages as resampling and its output correlates with the original MC PID
- The inputs are the Data and MC KDE fits and the original PID
- [Tutorial is online](#)
- Further development:



- In some cases, discrepancies have been observed during the validation
- Statistical uncertainties for the methods
- Create basf2 module that can sample or transform MC PID during basf2 runtime



# Status of integrated studies

$[\tau \rightarrow 3 \pi \nu] [\tau \rightarrow l \nu \nu]$

$e^+e^- \rightarrow \mu\mu$  ISR


$e^+e^- \rightarrow \gamma\gamma$

- Tau and low multiplicity modes for Lepton ID systematics
- **Ntuple production has been successfully integrated**
- In development:
  - There is no fittable invariant mass for these modes
    - Classical sWeight approach will not work
  - Original studies were using bin-by-bin MC based estimation of background levels
    - It can be implemented as a standalone b2luigi workflow
  - Trying a machine learning approach to subtract background and to estimate systematics
    - This approach is flexible and can have the same output columns as sWeight-based modes
- [Instructions for study integration are online](#)





# Custom variables

- Framework's functionality was largely extended by introduction of custom framework variables
  - Allows for any complexity of computation
  - Can be used in cut strings
  - kaonID/(pionID+kaonID) is now binaryID(K,pi)
  - One has to use release-07 or new light releases
-  [List of custom variables is online](#)
- As this list will never be complete, we have a mechanism for plug&play variables
  - [Tutorial is online](#)
- Help needed:
  - Expand the list of integrated variables
  - Enable the usage of the variables in all active strings

```
class TrinaryVariable(VariableBase):
    def __init__(self, arguments: list) -> None:
        # name should be the same as in
        # manager.register_variable
        # args_range is a minimum and maximum
        # number of arguments
        super().__init__(name='trinaryID',
                        arguments=arguments,
                        args_range=(3, 3))
        # list of variable names that this
        # meta variable depends on
        self.dependencies = [TYPE_TO_PID[arg]
                            for arg in arguments]

    def run(self, df: pd.DataFrame) -> pd.Series:
        return df[self.dependencies[0]] / \
            (df[self.dependencies[0]]
             + df[self.dependencies[1]]
             + df[self.dependencies[2]])

# Register the user variable in the framework:
manager.register_variable('trinaryID', TrinaryVariable)
```

# Ntuple production workflow

- Create `settings.json` configuration file:

```
{  
  "gbasf2_install_directory" : "~/gbasf2_install",  
  "gbasf2_print_status_updates" : true,  
  "gbasf2_max_retries" : 5,  
  "gbasf2_cputime" : 5,  
  "gbasf2_release" : "release-08-00-00",  
  "gbasf2_download_logs" : false,  
  "particleid_tasks" : "pid_task_list.yaml"  
}
```

- Run Dataset Searcher and write down the LPNs to a `.list` file
- Create a workflow configuration file
- Launch command:
  - `python3 -m syscorr fw`
- The final weighted ntuples contain 2 sWeights columns:
  - A nominal column and a systematic one
  - Enables computation of systematic uncertainties

```
---  
# ===== #  
# Hadron ID task submission file #  
# ===== #  
-  
  path: "my_lpns.list"  
  proc: "bucket14"  
  short_name: "Dst_b14"  
  model: "Dst"  
  cms_energy: "4S"  
-  
  path: "my_lpns.list"  
  proc: "bucket14"  
  short_name: "L0_b14"  
  model: "Lambda0"  
  cms_energy: "4S"
```

# Fixed cut weights workflow

- This new workflow produces data/MC correction tables which are suitable for ParticleWeighting module and PIDVar
- Features:
  - Compact configuration file
  - Any number of efficiency cuts on global and/or binary Hadron ID
  - Tables can have any dimensionality and binning
  - Arbitrary preselection cuts
  - Weights can be produced in parallel
  - [Documentation is online](#)
- **Systematic uncertainties are available**
- **Help needed:**
  - Create a possibility to automatically upload the weights to CDB

```
---  
# ===== #  
# Hadron ID weight configuration file #  
# ===== #  
weight_dir: 'fixed_weights/'  
remove tmp files: True  
weight_cfg_list:  
-  
  prefix_name: Rdtmc_v1  
  efficiency_particle_type: 'K'  
  fakerate_particle_type: 'pi'  
  binning: [[0.5, 2.5, 4.5],  
            [-0.8, 0.2, 0.9563]]  
  track_variables: [ "p", "cosTheta" ]  
  cuts: [ "kaonID > 0.2", "kaonID > 0.8" ]  
  precuts: [ "", "charge > 0", "charge < 0" ]  
  mc_proc_query: [ "MC14ri_1" ]  
  data_proc_query: [ "procl2e7" ]
```

- Launch command:
  - `python3 -m syscorr fw -w fixed_weights`

# Refit workflow for HadronID

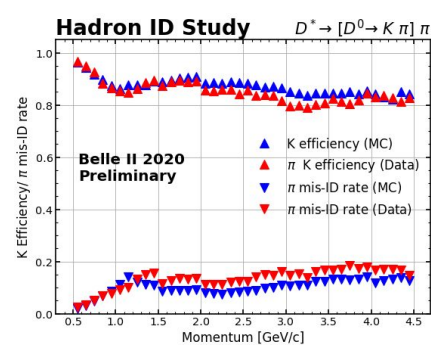
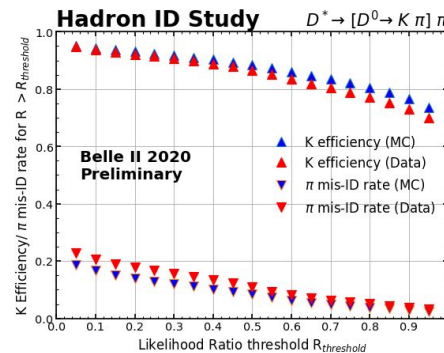
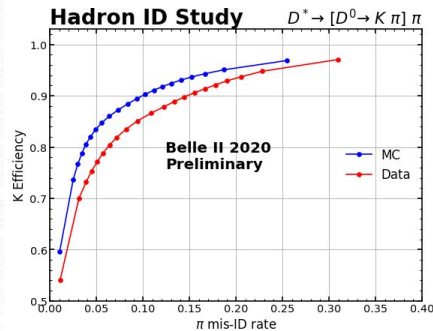
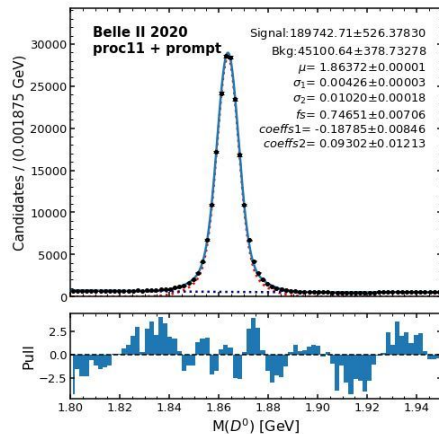
- This new workflow reproduces sWeights in already existing ntuples in case if a PDF model has been updated
- Features:
  - Compact configuration file
  - One can select a particular model to update
  - The ntuple files will acquire a new suffix and their sWeights column will be replaced
  - Workflow will also reproduce all benchmark plots
- Will be applied when we will update Lambda0 fit

- Launch command:
  - `python3 -m syscorr fw -w refit`

```
---  
# ===== #  
# Hadron ID refit configuration file #  
# ===== #  
old_suffix: ""  
new_suffix: "_v1"  
models: ["Lambda0", "Dst"]
```

# D\* K/ $\pi$ ID study

- D\* reconstruction note:
  - S. Sandilya BELLE2-NOTE-PH-2019-048
- Default cuts in the framework
  - impact parameters  $dr < 2$ ,  $abs(dz) < 4$  and CDC hits  $> 20$ ;
  - $p_{D^*}^{cms} > 2.5$  GeV;
  - $0.1439$  GeV  $< \Delta M < 0.1469$  GeV;
  - $1.8$  GeV  $< M(D^0) < 1.95$  GeV.
- Model for  $M(D^0)$ 
  - Signal: two Gaussian functions with a common mean;
  - Background: the second order Chebychev function.
- The full set of global PID variables (pionID, kaonID, etc) and the likelihoods for each mass hypothesis and detector is stored.



# $\Lambda^0$ $p/\pi$ ID study

- $\Lambda$  reconstruction note
  - B. Scavino BN-2020-027
- Two different skim types
  - Analysis skim:  $0.6 < p_p / p_\Lambda < 1.0$ ,  $\text{flightSignificance} > 3.0$ ,  $\text{protonID} > 0.1$ ,  $\cos(\alpha) > 0.99$
  - HLT skim:  $\text{flightSignificance} > 10.0$ ,  $(p_p - p_\pi)/(p_p + p_\pi) > 0.41$
  - Both skims slightly different than our selection, analysis skim is similar to Todd's studies:  $n\text{CDCHits} > 20$ ,  $p_p / p_\Lambda > 0.6$ ,  $\text{flightSignificance} > 2$
- ✓ Performance of sWeights relative to MC truth matching has been validated

