Nuclei wagon for MPDRoot

V. Kireyeu

07.12.23

Introduction

Wagon description

Efficiencies

Results

Summary

Introduction

The new ”wagon” for the light nuclei (d, He) analysis must be implemented within the MpdRoot train-like analysis chain. The nuclei wagon should:

- Be **highly configurable** to avoid the unnecessary source code recompiling.
- Be **highly automated** for the same reasons.
- Be **well documented**.
- Provide the **phase-space distributions** for the particles of interest.
- Provide the **TPC, ToF, PID, DCA efficiencies and PID contamination** within the same phase-space bins as the particles distributions **for the final results** corrections.

The first version of the ”nuclei” MpdRoot wagon will be presented in this talk.

Wagon description

Wagon structure and logic

- Initialization: settings a read from the JSON-file, histograms are booked for each:
  - particle
  - centrality bin
  - PID method (MC PDG, evPID wagon)
- Event processing:
  - Events are checked for the "event quality".
  - Tracks are checked for the "track quality".
  - Centrality bin is selected.
  - Particles are identified (MC, evPID).
  - Histograms for the identified histograms are filled.
- Helper subroutines for the configuration reading, "quality" checks, centrality check, particles selection etc.

Configuration: global, event cuts

```
"Verbose": "1",
"N_MPD_PID_Particles": "8",
"do_MC": "1",
"do_evPID": "1",
"Events": {
    "PrimaryVertexZ": "130",
    "Centrality": [[0, 10], [10, 20], [20, 30], [30, 40]]
},
```

Configuration: track quality, PID

```
"Tracks": {
    "NHits": "20",
    "NSigmaDCAx": "2",
    "NSigmaDCAy": "2",
    "NSigmaDCAz": "2",
    "LowPtCut": "0.05"
},
"PID": {
    "TPCSigma": "2",
    "TOFSigma": "2",
    "TOFDphiSigma": "3",
    "TOFDzSigma": "3"
},
```

Configuration: particles of interest

```
"Particles": {
    "p": {
        "PDG": "2212",
        "Mass": "0.938",
        "Enum": "3",
        "tpcLowMomentum": "0.2",
        "tpcHighMomentum": "2.8",
        "pt_bins": [320, 0.0, 8.0],
        "eta_bins": [320, -4.0, 4.0]
    },
```

Configuration: particles of interest

```
"d": {
    "PDG": "1000010020",
    "Mass": "1.876",
    "Enum": "4",
    "tpcLowMomentum": "0.2",
    "tpcHighMomentum": "2.8",
    "pt_bins": [160, 0.0, 8.0],
    "eta_bins": [160, -4.0, 4.0]
},
```

Usage

Dependencies:

- Branches: MCTrack, TpcKalmanTrack, ZdcDigi, Vertex, MPDEvent, TOFMatching.
- Wagons: evCentrality, evPID.

Usage (add these lines to your ”train” macro (e.g. ‘RunAnalyses.C‘)):

```
MpdNuclei taskNuclei("taskNuclei","taskNuclei","NucleiAna.json");
man.AddTask(&taskNuclei);
```

Histograms naming scheme

Example: **hv__eff_pdg_primary_nhits_dca_tof**

- **hv** – histograms vector
- **eff** – ”efficiency” histograms
- **pdg** – PID by MC
- **primary** – primary by MC
- **nhits** – with nhits cut
- **dca** – with dca cut
- **tof** – has ToF matchging

Each single histogram in this vector:
h__eff_pdg_primary_nhits_dca_tof_%s_centrality%d

- **h** – single histogram
- **%s** – particle name from the JSON configuration file (”p”, ”d”, etc)
- **%d** – centrality bin number (0, 1, etc)

Histograms naming scheme

Example: **hv__pteta_evpid**

- **hv** – histograms vector
- **pteta** – phase-space histograms ”p_T *vs* η”
- **evpid** – PID by evPID wagon

Each single histogram in this vector: **h__pteta_%s_evpid_centrality%d**

- **h** – single histogram
- **%s** – particle name from the JSON configuration file (”p”, ”d”, etc)
- **%d** – centrality bin number (0, 1, etc)

Postprocessing

For the test purpose one can run the MpdRoot analysis train on the NICA cluster – in this case it would be a good idea to run tasks in parallel, e.g. 1000 parallel jobs, each job process 20000 events.

As the output one will have 1000 files with efficiency and phase-space histograms.

These histograms must be concatenated into the single file with the ”hadd” program:

```
$ hadd final_file.root /some/directory/*.root
```

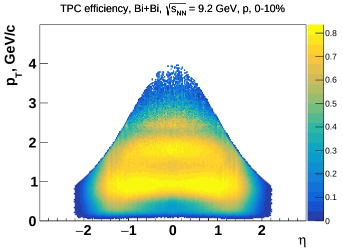
Postprocessing

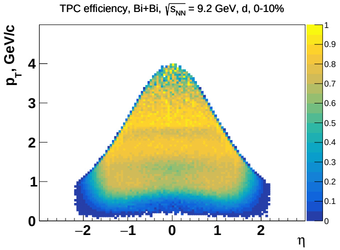
Now, the final single file can be processed.

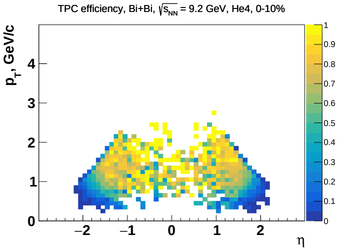
```
TH2D *hResult   = (TH2D*) inFile -> Get(Form("h__pteta_%s_evpid_centrality%d", pname, c_bin)) -> Clone("hResult");
TH2D *hEfficiency = nullptr;
TH2D *hNumerator = nullptr;
TH2D *hDenominator = nullptr;

hNumerator   = (TH2D*) inFile -> Get(Form("h__eff_pdg_primary_nhits_dca_%s_centrality%d", pname, c_bin)) -> Clone("hNumerator");
hDenominator = (TH2D*) inFile -> Get(Form("h__eff_pdg_primary_%s_centrality%d", pname, c_bin))           -> Clone("hDenominator");
hEfficiency = (TH2D*) hNumerator -> Clone("hEfficiency");
hEfficiency -> Divide(hNumerator, hDenominator);
hResult -> Divide(hEfficiency); // TPC efficiency

hNumerator   = (TH2D*) inFile -> Get(Form("h__eff_pdg_primary_nhits_dca_tof_%s_centrality%d", pname, c_bin)) -> Clone("hNumerator");
hDenominator = (TH2D*) inFile -> Get(Form("h__eff_pdg_primary_nhits_dca_%s_centrality%d", pname, c_bin))     -> Clone("hDenominator");
hEfficiency -> Divide(hNumerator, hDenominator);
hResult -> Divide(hEfficiency); // ToF efficiency

hNumerator   = (TH2D*) inFile -> Get(Form("h__eff_pdg_nhits_dca_tof_pid_%s_centrality%d", pname, c_bin)) -> Clone("hNumerator");
hDenominator = (TH2D*) inFile -> Get(Form("h__eff_pdg_nhits_dca_tof_%s_centrality%d", pname, c_bin))     -> Clone("hDenominator");
hEfficiency -> Divide(hNumerator, hDenominator);
hResult -> Divide(hEfficiency); // PID efficiency

hNumerator   = (TH2D*) inFile -> Get(Form("h__eff_primary_nhits_dca_tof_pid_%s_centrality%d", pname, c_bin)) -> Clone("hNumerator");
hDenominator = (TH2D*) inFile -> Get(Form("h__eff_nhits_dca_tof_pid_%s_centrality%d", pname, c_bin))         -> Clone("hDenominator");
hEfficiency -> Divide(hNumerator, hDenominator);
hResult -> Divide(hEfficiency); // DCA efficiency
```

Efficiencies

TPC efficiency





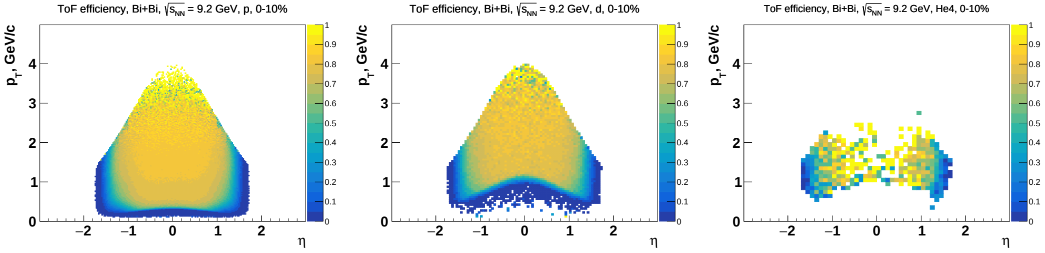


$$TPC\ efficiency = \frac{hv\_\_eff\_pdg\_primary\_nhits\_dca}{hv\_\_eff\_pdg\_primary}$$

hv__eff_pdg_primary_nhits_dca – PID by MC, primary by MC, with nhits cut, with dca cut. The low p_T cut is also here.
hv__eff_pdg_primary – PID by MC, primary by MC. There is no low p_T cut.

Courtesy to A. Mudrokh

ToF efficiency

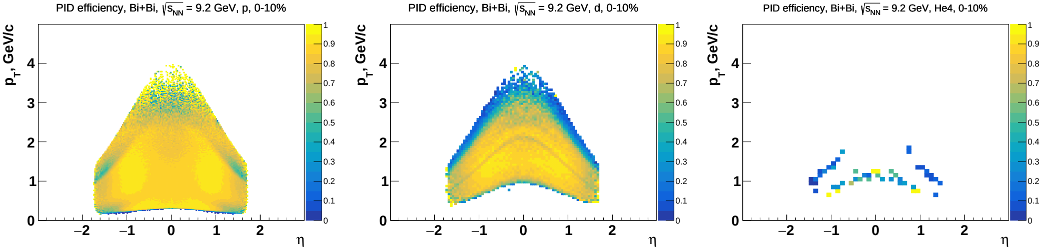


$$ToF\ efficiency = \frac{hv\_\_eff\_pdg\_primary\_nhits\_dca\_tof}{hv\_\_eff\_pdg\_primary\_nhits\_dca}$$

hv__eff_pdg_primary_nhits_dca_tof – PID by MC, primary by MC, with nhits cut, with dca cut, has ToF matching.
hv__eff_pdg_primary_nhits_dca – PID by MC, primary by MC, with nhits cut, with dca cut.

Courtesy to A. Mudrokh

PID efficiency


PID efficiency, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 0-10%

PID efficiency, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 0-10%

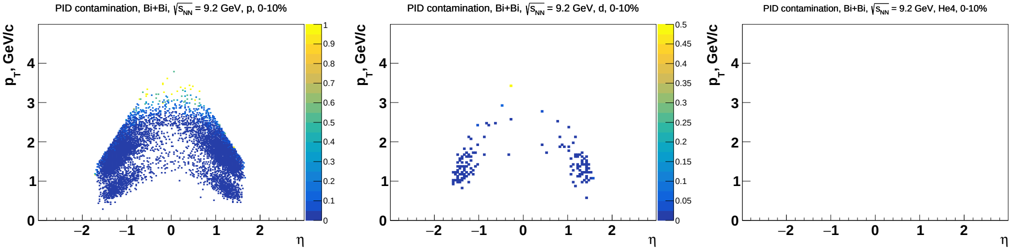
PID efficiency, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 0-10%

$$PID\ efficiency = \frac{hv\_\_eff\_pdg\_nhits\_dca\_tof\_pid}{hv\_\_eff\_pdg\_nhits\_dca\_tof}$$

hv__eff_pdg_nhits_dca_tof_pid – PID by MC, with nhits cut, with dca cut, has ToF matching, PID by wagon = PID by MC.
hv__eff_pdg_nhits_dca_tof – PID by MC, with nhits cut, with dca cut, has ToF matching.

Courtesy to A. Mudrokh

PID contamination



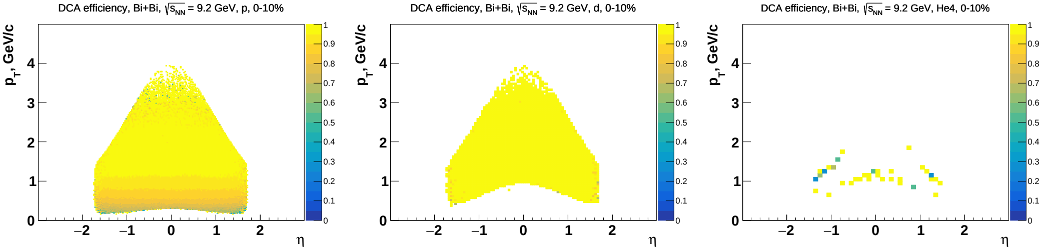
$$PID\ contamination = \frac{hv\_\_eff\_pdg\_nhits\_dca\_tof\_wpid}{hv\_\_eff\_pdg\_nhits\_dca\_tof\_pid}$$

hv__eff_pdg_nhits_dca_tof_wpid – PID by MC, with nhits cut, with dca cut, has ToF matching, PID by wagon $\neq$ PID by MC.

hv__eff_pdg_nhits_dca_tof_pid – PID by MC, with nhits cut, with dca cut, has ToF matching, PID by wagon = PID by MC.

Courtesy to A. Mudrokh

DCA efficiency



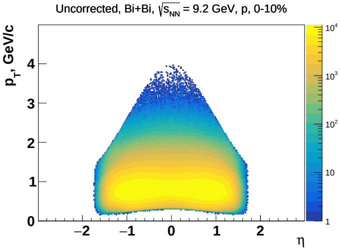
$$DCA\ efficiency = \frac{hv\_\_eff\_primary\_nhits\_dca\_tof\_pid}{hv\_\_eff\_nhits\_dca\_tof\_pid}$$

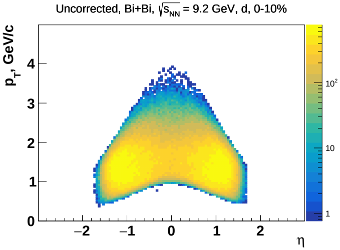
hv__eff_primary_nhits_dca_tof_pid – PID by wagon, primary by MC, with nhits cut, with dca cut, has ToF matching.
hv__eff_nhits_dca_tof_pid – PID by wagon, with nhits cut, with dca cut, has ToF matching.

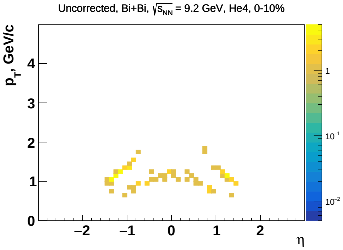
Courtesy to A. Mudrokh

Results

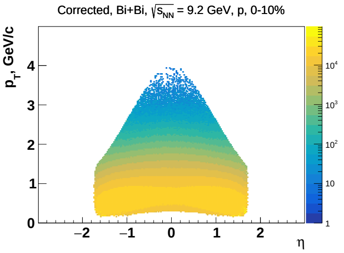
Uncorrected results

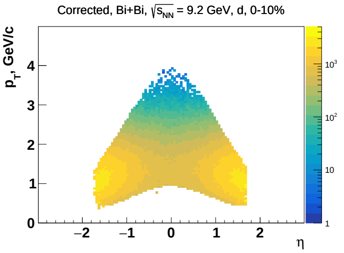

Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 0-10%

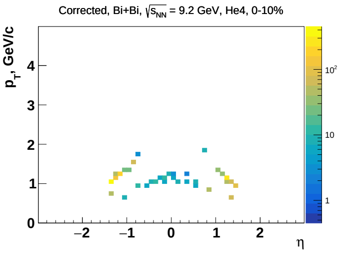

Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 0-10%


Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 0-10%

Corrected results

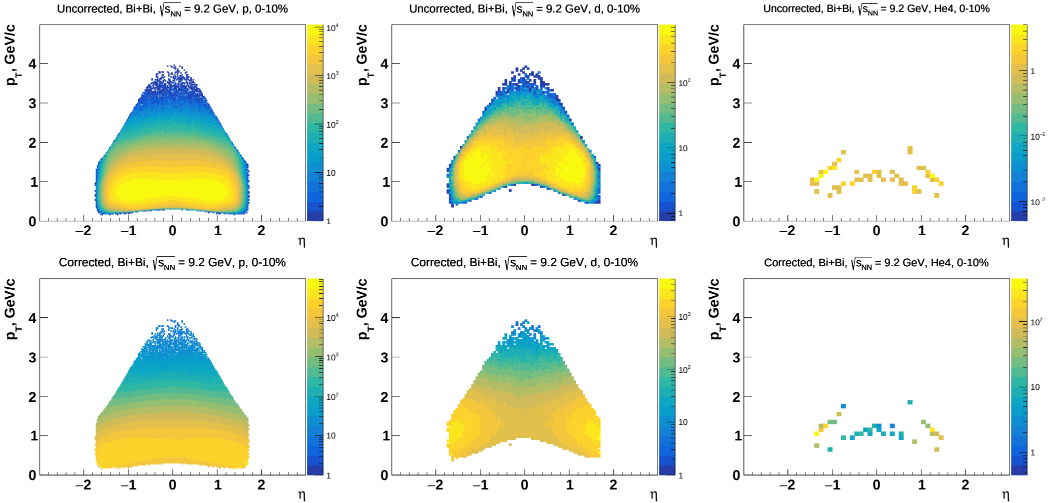






$$Result = \frac{Uncorrected \cdot (1 - PID\ \ contamination)}{TPC\ \ efficiency \cdot ToF\ \ efficiency \cdot PID\ \ efficiency \cdot DCA\ \ efficiency}$$

Uncorrected and corrected results: 0-10%



Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 0-10%

Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 0-10%

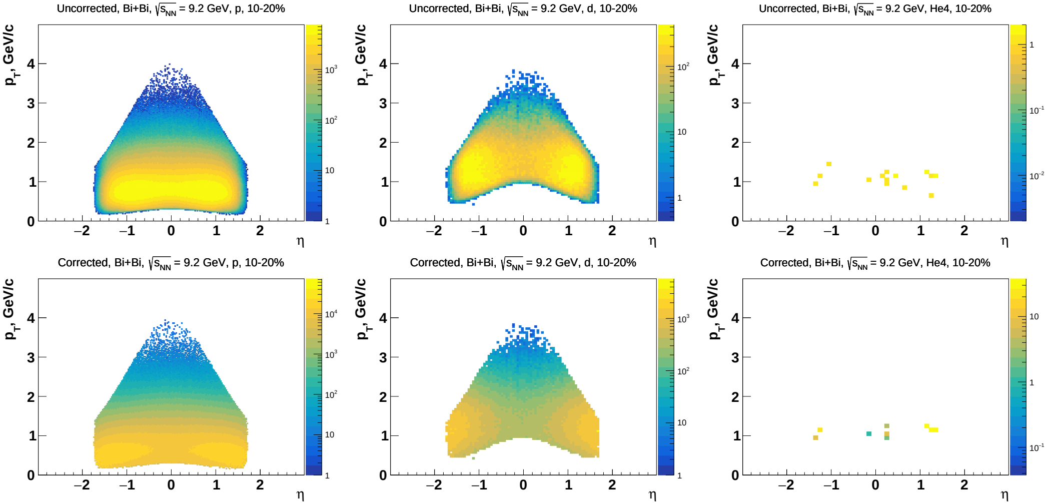
Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 0-10%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 0-10%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 0-10%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 0-10%

Uncorrected and corrected results: 10-20%



Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 10-20%

Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 10-20%

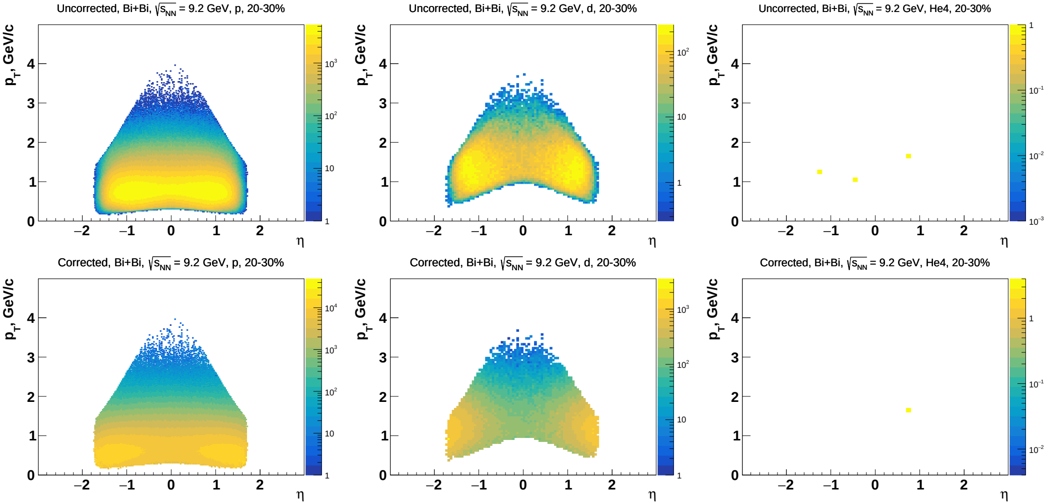
Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 10-20%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 10-20%

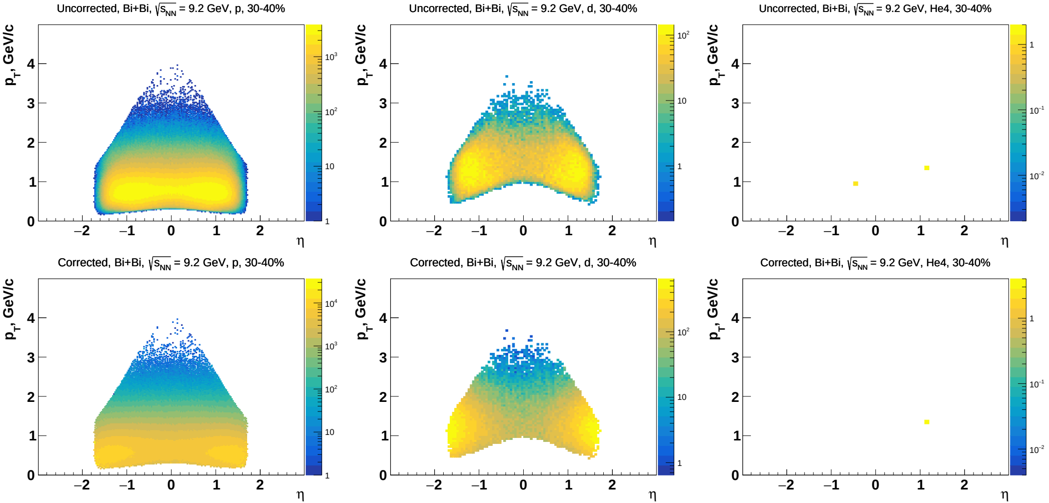
Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 10-20%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 10-20%

Uncorrected and corrected results: 20-30%



Uncorrected and corrected results: 30-40%



Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 30-40%

Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 30-40%

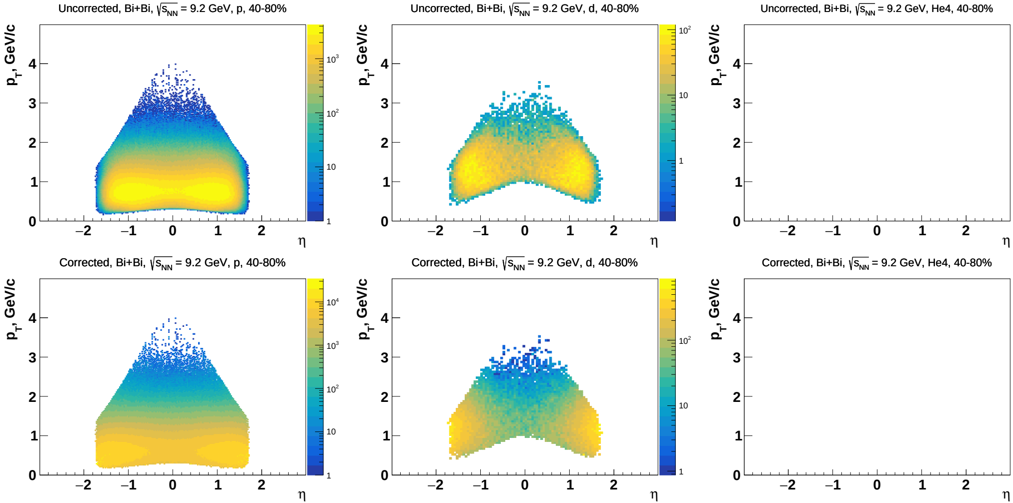
Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 30-40%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 30-40%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 30-40%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 30-40%

Uncorrected and corrected results: 40-80%



Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 40-80%

Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 40-80%

Uncorrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 40-80%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, p, 40-80%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, d, 40-80%

Corrected, Bi+Bi, $\sqrt{s_{NN}}$ = 9.2 GeV, He4, 40-80%

Summary

The first version of the "Nuclei" wagon is presented:

- The wagon uses the JSON-formatted input file to handle all possible settings and automatically create histograms for the defined particles.
- Only phase-space histograms (p_T vs η) are included.
- Different efficiencies are calculated within same phase-space bins:
  - TPC efficiency
  - ToF efficiency
  - PID efficiency
  - DCA efficiency
  - PID contamination
- TPC, ToF, PID, DCA efficiencies and PID contamination are used for the final results corrections.
- The Doxygen-style documentation for the wagon is provided.

Current **proposals**:

- Push the "nuclei" wagon into the "dev" version of the MpdRoot.
- Revise the definitions of efficiencies.
- Move from the "p_T/η" phase-space to the "p_T/y" for the final results.
- Merge the bulk spectra and nuclei wagons into one?

Thank you for your attention!

This presentation was prepared using LaTeX with the Beamer package on Overleaf.