

# An Error Analysis Toolkit for Binned Counting Experiments

Ben Messerly<sup>1,2,\*</sup>, Rob Fine<sup>3,4,\*\*</sup>, and Andrew Olivier<sup>3,\*\*\*</sup>, on behalf of the MINERvA Collaboration

<sup>1</sup>University of Pittsburgh

<sup>2</sup>University of Minnesota

<sup>3</sup>University of Rochester

<sup>4</sup>Los Alamos National Laboratory

**Abstract.** We introduce the MINERvA Analysis Toolkit (MAT), a utility for centralizing the handling of systematic uncertainties in HEP analyses. The fundamental utilities of the toolkit are the `MnvhNd`, a powerful histogram container class, and the systematic `Universe` classes, which provide a modular implementation of the many universe error analysis approach. These products can be used stand-alone or as part of a complete error analysis prescription. They support the propagation of systematic uncertainty through all stages of analysis, and provide flexibility for an arbitrary level of user customization. This extensible solution to error analysis enables the standardization of systematic uncertainty definitions across an experiment and a transparent user interface to lower the barrier to entry for new analyzers.

## 1 Introduction

Despite the importance and complexity of systematic uncertainty evaluation in HEP, there is no community-wide standard for the storage and propagation of uncertainty through analysis infrastructures. At the experiment level, an error analysis system must be a central consideration during the design of analysis infrastructure. In the absence of such a system, consideration of systematic uncertainty is incumbent on individual analyzers and can be relegated to the final stages of analysis after substantial design choices have already been made. This impedes the effectual integration of error analysis and represents a substantial barrier to entry for new analyzers. To this point, analyzers within an experiment are often left to implement solutions to this independently of one another, creating an inefficiency of effort and increased opportunity for inconsistency and mistakes.

We present a solution to this problem, the MINERvA Analysis Toolkit (MAT), which can be adopted to streamline an experiment's approach to evaluating systematic uncertainty. The MAT (1) implements the many-universe error method, (2) is built on a unique and powerful histogram object (`MnvhNd`), and (3) offers experiment-wide standardization of systematics and simultaneous customizability and extensibility via `Universe` classes. This toolkit was developed within the MINERvA analysis environment where it has facilitated dozens of neutrino cross section measurements over the last decade [1–34] and forms the basis for MINERvA's

---

\*e-mail: [mess@umn.edu](mailto:mess@umn.edu)

\*\*e-mail: [finer@fnal.gov](mailto:finer@fnal.gov)

\*\*\*e-mail: [aolivier@ur.rochester.edu](mailto:aolivier@ur.rochester.edu)

data preservation effort [35]. The MAT was designed to be adopted directly, in part or in whole, by any HEP experiment. It is directly applicable to any ROOT-based [36] analysis environment in which simulated physics interactions are evaluated on an event-by-event basis. We also believe that the underlying approach and some of the classes which underpin the MAT's functionality could be implemented in any software environment for a binned counting experiment in any field of study.

In this paper, we introduce in detail the design of the MAT, provide examples for its current and future use within MINERvA, and describe how it could be adopted more widely. Section 2 introduces the "many universe" approach to evaluating systematic uncertainty, which is a prerequisite for this toolkit. Sections 3 and 4 describe the technical implementations of the `MvHnD` and `Universe` classes, and describe how MAT analyses are structured, including pseudocode examples. Finally, Section 5 discusses MAT integration into different experimental analysis environments and benefits of adopting the MAT wholesale.

## 2 "Many Universe" Error Analysis

Modern experimental HEP is inextricably coupled to complex simulations that are designed to replicate the physics we measure. These simulations are parameterized by variables that correspond to the design of an experiment and the related physics processes. In general, the output of these simulations is a collection of individual interactions (events), which are subject to the same reconstruction and analysis used for data. The observables of an analysis are binned as functions of kinematic variables and stored in histograms. When analyzed in aggregate over many events, the distribution of simulated event kinematics can be compared to data to judge the efficacy of the simulation. In this manner data is used to fine-tune the simulation and ultimately to improve the underlying physics models.

In the "many universe" approach to evaluating systematic uncertainty, aspects of the simulation and reconstruction are shifted to reflect distinct sources of uncertainty. An analysis is repeated many times, and in each instance the effect of a particular shift is propagated through subsequent stages of the simulation-reconstruction-analysis chain. The particular configuration of the simulation and reconstruction in any individual instance defines a "universe". Two universes may be distinct from one another, for example, because of the choice of values for parameters within the beamline simulation or within a track reconstruction algorithm. The choice of nominal values (or "best guesses") for all parameters in the simulation and reconstruction defines the "Central Value" (CV) universe and forms the basis against which data measurements are compared, and against which uncertainties are evaluated. Each systematic universe is constructed to isolate the impact of shifting a particular aspect of the CV universe in a measured way, and the uncertainty on a measurement corresponding to each individual source is determined based on the spread of outcomes across two or more universes.

It can be constructive to distinguish between two conceptually different classes of shifts away from the CV. The first, *vertical shifts*, arise from sources of systematic uncertainty which do not directly affect any kinematic variable, and their effect is only to modify the weight a particular event is given in an analysis. Vertical shifts cause the contents of a histogram bin to be increased or decreased but do not lead to event migration between bins, and therefore will never lead to events migrating into or out of a selected sample. Because of this, the effect of a vertical shift of a far-upstream aspect of the simulation can be characterized downstream in the analysis by changing the relative weighting of events based on their kinematics. This "reweighting" enables evaluation of vertical systematic shifts without the need to repeat computationally intensive stages of the simulation. The second, *lateral shifts*, arise from sources of systematic uncertainty that do directly affect a kinematic variable and therefore can cause events to migrate between bins and into and out of a selected sample. A

lateral shift may cause an event which does not pass a selection cut in the CV universe to instead pass in the shifted universe, or vice versa. The impact of a lateral shift on the final result must be determined by repeating the portion of the simulation-reconstruction-analysis chain downstream of the shift, and cannot be evaluated via reweighting.

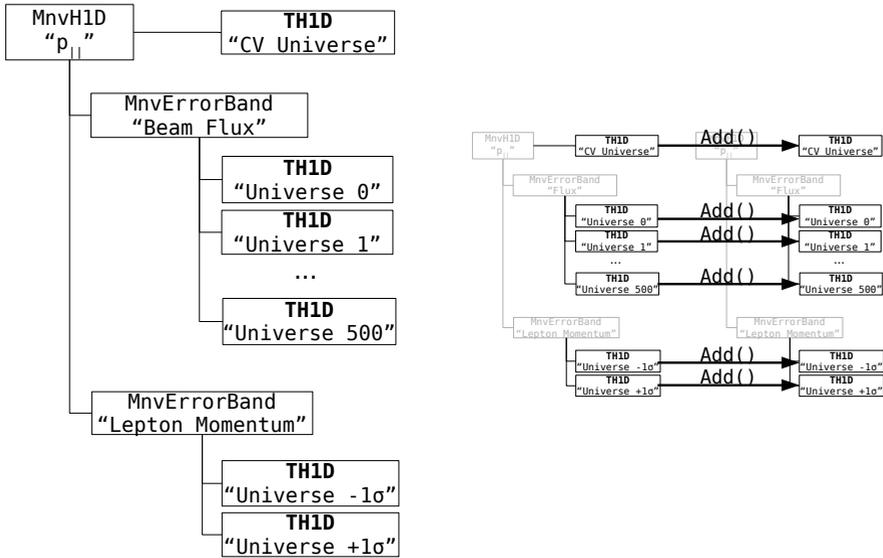
In practice, the value of a systematic shift is determined *a priori*, and often corresponds to the standard deviation of a Gaussian distribution. For example, a dedicated study of muon momenta might inform a  $\pm 2\%$  shift applied to muons on an event-by-event basis to represent the uncertainty associated with their reconstruction. In this case, the impact of this uncertainty on an observable is calculated using the variance between the  $+2\%$ - and  $-2\%$ -shifted universes in each bin of the analysis. Many uncertainties are adequately represented by two universes because the pair of universes reflects a simple underlying distribution (usually assumed to be Gaussian). Others motivate a large number of universes to encapsulate the effect of simultaneously varying multiple parameters within a more complex model. For example, neutrino beamline fluxes are predicted using notoriously complex simulations, in which the correlations among internal parameters are nontrivial. In this case, an ensemble of universes is used, and in each multiple parameters are varied according to their individual statistical distributions. The impact of this uncertainty on an observable is then calculated using the variance between all of the universes in each bin of the analysis.

Within the MAT, individual systematic universes are represented by `Universe` classes, and universes corresponding to the same source of uncertainty are grouped into "error bands". The `MnvHnD` class, described in detail in the following section, provides the interface to propagate an analysis histogram in many systematic universes simultaneously. The uncertainty on an observable stored in an `MnvHnD` can be calculated on-demand at any stage of the analysis by using the spread across universes to construct a covariance matrix corresponding to any individual systematic uncertainty. The covariance matrix corresponding to any individual systematic uncertainty, or to the total uncertainty, can be reported, or the diagonal entries of the covariance matrix can be used to produce vertical error bars on a histogram. This approach to reporting uncertainty is common in HEP, but the many universe technique does not exclude alternative error treatments.

### 3 The `MnvHnD` Class

A successful implementation of the many universe approach requires that an analysis be repeated many times. In the MAT, this is achieved using an `MnvHnD` object, which stores the central value and systematic universe histograms for an analysis observable. With all histograms stored in a single object, analysis calculations can proceed simultaneously in all universes. The interface of the `MnvHnD` is implemented by broadcasting the `THnD` interface from ROOT to each systematic universe, so uncertainties are automatically propagated correctly when `MnvHnDs` are `Add()`d, `Divide()`d, or `Scale()`d. An `MnvHnD` can `Write()` itself to a ROOT file to persist an analysis result with everything needed to produce a covariance matrix, defer further processing to a separate program, or separate plot formatting from computationally intensive analysis routines.

An `MnvHnD` encapsulates a parallel histogram for each systematic universe as shown in Figure 1. To populate a histogram in a particular universe, an analyzer need only tell an `MnvHnD` which `Universe` a `Fill()` value belongs to (as in line 12 of Code Example 1). The covariance matrix for any group of `Universes` can then be calculated on demand. An uncertainty is usually evaluated using two or more `Universes` grouped into an error band (implemented in the MAT as the `MnvErrorBand` class). Covariance matrices are calculated for each error band individually, and the total covariance matrix on a result is equal to the sum of all of the



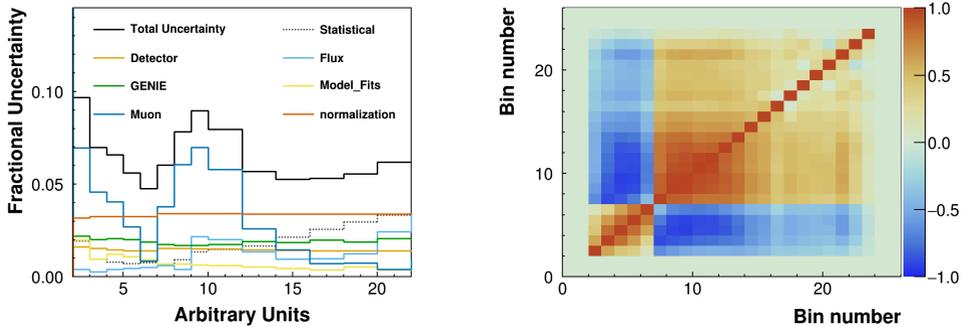
**Figure 1.** Left: An `MnvH1D` encapsulates one histogram for each Universe. Universes are grouped into `MnvErrorBands`. The `MnvH1D` is itself the CV TH1D. Right: The `MnvH1D` propagates uncertainties by performing TH1D operations like `Add()` in each Universe.

individual covariance matrices. In this way, the variance of each bin is added in quadrature as expected for uncorrelated uncertainties.

The MAT provides tools to visualize the data and uncertainties stored in an `MnvH1D`. Because each error band is propagated independently, it is straightforward to see how each source of uncertainty contributes to the total uncertainty of a distribution. Figure 2 shows an example of an uncertainty breakdown as a function of an arbitrary kinematic variable, and the associated total correlation matrix. One of the key advantages to the MAT is that, through the bookkeeping provided by the `MnvH1D`, a user retains access to an observable in all systematic universes throughout every stage of their analysis. This enables, for example, detailed studies comparing the relative impact of uncertainties on different measurements.

## 4 Systematic Universe Classes

The MAT implements the many universe approach to error analysis, described in Section 2, using the `MnvH1D` to store histograms and a `Universe` interface to abstract the concept of systematic universes. In a typical analysis, the user defines a CV `Universe` which includes methods to calculate all required event kinematics and weights (using information from the event record), and systematic `Universe` classes model alternatives to the default analysis by overriding the behavior of the CV `Universe`. The methods of the CV `Universe` serve as a base interface for all other `Universe`s, with each systematic `Universe` overriding one or more of the CV methods. In the case of vertical systematic shifts, one or more methods which calculate an event’s weights are overridden, and in the case of lateral systematic shifts, one or more methods which calculate an event’s kinematics are overridden. An analysis includes a loop over all `Universe`s



```

1 hist->DrawErrorSummary(); // [left]
2 errMatrix = hist->GetCorrelationMatrix();
3 errMatrix.Draw(); // [right]
    
```

**Figure 2.** The MAT provides data visualization tools that operate on an `MnvHnD`. Left: An error budget is constructed by extracting the diagonal entries of distinct `MnvErrorBands`’ covariance matrices to show the relative contributions of independent sources of uncertainty. Right: The total correlation matrix is constructed by adding together (and normalizing) the covariance matrices of all `MnvErrorBands`.

(CV and systematic), and the event kinematics and weights in each `Universe` are used to fill histograms in the corresponding `MnvErrorBands` of `MnvHnDs`.

Code Example 1 shows an example of an event loop in a MAT-based analysis. Within the loop over independent events is a loop over systematic `Universes`. In the MAT’s simplest form, the full event selection is repeated for each `Universe` (see code starting on line 9). Compared to the CV analysis, each systematic `Universe` might, on an event-by-event basis, change the result of applying a cut (line 9) or calculating an event kinematic (line 10) or weight (line 11).

```

1 double PassesCuts(const ExampleUniverse& univ) {
2     return univ.GetLeptonEnergy() < 20;
3 }
4
5 MnvH1D energyHistogram(systematics);
6
7 for(const auto& event: eventRecord) {
8     for(const ExampleUniverse& univ: systematics) {
9         if(PassesCuts(univ)) {
10            const double energy = univ.GetEnergy();
11            const double weight = univ.GetWeight();
12            energyHistogram.FillUniverse(univ, energy, weight);
13        }
14    }
15 }
    
```

**Code Example 1.** An example of an analysis event loop in the MAT. Within the loop over independent events is a loop over systematic `Universes`. In each `Universe`, cuts are applied and event kinematics and weights are calculated. The `PassesCuts()` function needs no special logic to handle systematic uncertainties that may affect it. Rather than fill a `TH1D`, the user instead fills an `MnvH1D`.

Code Example 2 shows an example of a `Universe`. This class serves both as the interface that all systematics implement and as the CV implementation itself. The MAT is written in C++, so weights and observables are virtual functions that can be overridden by derived classes. `BaseUniverse` is the parent class to all `Universes`, and can provide experiment management and optimization tools.

```
1 class ExampleUniverse: public BaseUniverse
2 {
3     public:
4         //Constructor and destructor details...
5
6         virtual double GetLeptonEnergy() const {
7             return m_tree->Get("lepton_energy");
8         }
9
10        virtual double GetRecoilEnergy() const {
11            return m_tree->Get("recoil_energy");
12        }
13
14        virtual double GetEnergy() const {
15            return GetLeptonEnergy() + GetRecoilEnergy();
16        }
17
18        virtual double GetTrackPIDScore() const {
19            return m_tree->Get("track_PID_score");
20        }
21
22        virtual double GetWeight() const {
23            return 1;
24        }
25
26    private:
27        TreeWrapper* m_tree;
28 };
```

**Code Example 2.** Example of a Universe, including methods to calculate event kinematics and weights. This class serves both as the Universe interface for an analysis and as the CV Universe implementation. Every observable that could be affected by a systematic uncertainty is implemented as a virtual function. An observable can be shifted by overriding its virtual function as done in Code Example 4.

Systematic Universes are implemented as mixins, as in Code Example 3, so that uncertainties central to an experiment are implemented the same way by all analyzers. In this example, `ResonantPionUniverse<>` is a template for a class that derives from a CV Universe with a `GetWeight()` function. This class template only needs to know about the weight it shifts. When we construct the list of Universe pointers to loop over, the base Universe class carries the rest of the CV interface into `ResonantPionUniverse<Universe>`. Similarly, in Code Example 4 the `GetLeptonEnergy()` observable is shifted while an arbitrary Universe interface is left intact. If these Universes were plugged into the systematics container in Code Example 1, `CalibrationLeptonEnergyUniverse` (Code Example 4) would change which events make it past line 9, and `ResonantPionUniverse` (Code Example 3) would change the shape of the histogram stored in the `TH1D` of the corresponding error band in `energyHistogram`. `energyHistogram` itself is an `Mnvh1D` which contains the CV histogram and two `MnvErrorBands`.

The separation of concerns between physics calculations and the evaluation of systematic uncertainties brings some broad advantages to MAT-based analyses. An analysis routine that fills `MnvHnDs` is modeled by the same code for each Universe. Analyzers only need to study one path of control to understand what quantity each histogram contains. A developer focused on a particular systematic uncertainty only needs to read a Universe's implementation rather than hunt for blocks of code scattered throughout the analysis routine. Version control systems automatically provide a concise summary of how a systematic uncertainty algorithm has developed over time. Furthermore, any MAT analysis program is modular and does not need to be changed to introduce a new systematic uncertainty.

```
1 template <class UNIVERSE>
2 class ResonantPionUniverse: public UNIVERSE
3 {
4     public:
5         //Constructors to match UNIVERSE go here
6
7         double GetWeight() const override {
8             return
9             this->IsResonantPion()?UNIVERSE::GetWeight():1.1*UNIVERSE::GetWeight();
10        };
11};
```

**Code Example 3.** The event weight is scaled up 10% for events which are truly resonant pion interactions in this Universe. This is an example of a "vertical" systematic shift.

```
1 template <class UNIVERSE>
2 class CalibrationLeptonEnergyUniverse: public UNIVERSE
3 {
4     public:
5         //Constructors to match UNIVERSE go here
6
7         double GetLeptonEnergy() const override {
8             return UNIVERSE::GetLeptonEnergy() + 50; //MeV
9         }
10 };
```

**Code Example 4.** The event's lepton energy is increased by 50 MeV in this Universe. This is an example of a "lateral" systematic shift.

## 5 MAT Adoption

The MAT's flagship tools, the `MNVHND` and `Universe` classes, are not reliant on particular technologies, but are merely powerful aids to many-universe binned analyses. They work well together, but are also standalone classes and can each be adopted individually to support an experiment's systematics infrastructure. The `MNVHND` on its own can simplify analysis calculations in many universes, provide on-demand covariance matrices, enable visualization of individual error bands (as shown in Figure 2), and serve as a standard histogram container for experiments. The `Universe` classes can standardize systematics treatment across an experiment and isolate systematic impacts to the single simulation parameter or analysis quantity that they affect.

Beyond these tools, the MAT has an intended approach to analysis, which, when adopted, unlocks the full potential of the toolkit. This approach, demonstrated in Code Example 1, uses a single event-universe loop to measure an analysis quantity and collect into an `MNVHND` all information needed to calculate its systematic errors. This approach simplifies analysis flow, gives a central role to systematic uncertainties, and makes transparent the particular many-universe strategy employed. This approach, however, is best-used in concert with certain experiment-wide choices regarding data organization and analysis infrastructure design.

It would be difficult to capture the full range of possible conditions in an experiment's analysis environment and their consequences on MAT implementation, so we consider only MINERvA here as an illustrative case. On MINERvA, the common reconstruction pass over raw data is relatively minimal. Each analyzer thus has access to low-level variables, and runs their own reconstruction (for example, specialized tracking) on top of the general reconstruction to produce a custom tuple with new branches tailored to their analysis needs. This design choice is well-suited to the MAT approach because access to low-level objects allows users to dynamically study new sources of uncertainty that affect low- or high-level variables alike.

In general, sources of systematic uncertainty can shift quantities throughout the simulation-reconstruction-analysis chain. Some are most conveniently expressed as a shift in a high-level, downstream analysis variable, which can be directly encoded in a `Universe` class (see, for example, Code Example 4). Others may most conveniently shift a low-level simulation variable with consequences downstream on the analysis that are too-complicated to summarize in terms of high-level variables (see Code Example 5). After quantifying the new source of uncertainty and the point in the simulation-reconstruction-analysis chain where it should be introduced, MINERvA analyzers can then share across the experiment a corresponding `Universe` class and (if needed) a reconstruction prescription to produce systematically shifted higher-level branches.

```
1 template <class UNIVERSE>
2 class TargetMassUniverse: public UNIVERSE
3 {
4     public:
5         //Constructors to match UNIVERSE go here
6
7         double GetTrackPIDScore() const override {
8             return m_tree->Get("track_PID_score_target_mass_shift");
9         }
10 };
```

**Code Example 5.** The uncertainty in target mass has a complex impact on a track's particle identification score. The shift is evaluated during reconstruction and its effect on particle identification is stored as a tuple branch, `track_PID_score_target_mass_shift`.

MINERvA's infrastructure choices, individual reconstruction and access to low-level objects, were feasible within MINERvA's processing and data storage constraints, and they enable MAT's event-universe loop to be a practical one-stop-shop for filling the CV and systematically-altered histograms of the `MNVHND`. These choices further enable agile development of experiment-wide systematic errors in which users are able to express systematics in their most convenient form (as shifts in either high- or low-level variables).

It is not practical in every case to provide the user access to the lowest-level parameters which may be shifted by a source of systematic uncertainty. Interaction event generators, for example, provide event weights which are relatively high-level expressions of an underlying shift on a model parameter. MINERvA has found a good compromise between providing users direct access to some low-level variables for studying new systematics and offering higher-level aggregated shifts as weights to avoid computationally expensive re-simulation. For any compromise, in order to make use of MAT's single-loop analysis approach, it is critical that experiments make available to the user a representation of a systematic shift that they can propagate through their analysis.

## 6 Conclusion

In section 1, the need for common error analysis tools and a top-down approach to error analysis was motivated and the MAT was introduced. Section 2 introduced the abstract concept of universes and their role in the many universe error analysis method. Sections 3 and 4 described MAT's foundational tools, the `MNVHND` and `Universe` classes, and it showed an example of typical MAT analysis flow. Finally, section 5 discussed the adoption of all or part of the MAT, and the important experiment-wide choices that impact its potential benefits.

There are extensions to the MAT which build on the functionality we have presented. In addition to providing the analysis foundation for dozens of MINERvA cross section measurements, the MAT is now serving MINERvA's data preservation effort. For eventual analysis of preserved MINERvA data, measurements in terms of new observables is challenging be-

cause the expertise needed to assign systematic uncertainties to those observables may no longer exist. A MAT analysis solves this problem automatically for any combination of preserved observables. MINERvA has also built on a central principle of the MAT and had some success recasting new models as reweights of existing models [26]. MAT analyses can introduce reweights of the underlying model the same way a vertical systematic universe is implemented.

As experiments grow, accessible and flexible analysis systems will become increasingly important for maximizing physics output. The MAT lowers the barrier to entry for analyzers by providing them a working error analysis system, thereby allowing users to spend less time writing code and more time thinking about physics. MAT's systematics standardization further reduces code duplication and potential for mistakes.

The MINERvA collaboration is preparing to publicly release the MAT this year, and continues active development to expand its capabilities. The MAT has not yet been used to perform a neutrino oscillation analysis or to apply a wide variety of machine learning methods, but we see no obstacles to those applications. Users of the MAT are currently exploring the idea of sharing systematics inter-experimentally. For example, we believe that broad adoption of the MAT could facilitate shared implementation of GENIE [37] model uncertainties in the neutrino interactions community. Outside of HEP, MAT's ideas could be implemented in new technologies and in any field which performs a many-universe binned analysis.

## References

- [1] A. Filkins et al. (MINERvA), Phys. Rev. **D101**, 112007 (2020), 2002.12496
- [2] D. Coplowe et al. (MINERvA), Phys. Rev. **D102**, 072007 (2020), 2002.05812
- [3] M.F. Carneiro et al. (MINERvA), Phys. Rev. Lett. **124**, 121801 (2020), 1912.09890
- [4] T. Cai et al. (MINERvA), Phys. Rev. **D101**, 092001 (2020), 1910.08658
- [5] T. Le et al. (MINERvA), Phys. Rev. **D100**, 052008 (2019), 1906.08300
- [6] E. Valencia et al. (MINERvA), Phys. Rev. **D100**, 092001 (2019), 1906.00111
- [7] P. Stowell et al. (MINERvA), Phys. Rev. **D100**, 072005 (2019), 1903.01558
- [8] M. Elkins et al. (MINERvA), Phys. Rev. **D100**, 052002 (2019), 1901.04892
- [9] D. Ruterbories et al. (MINERvA), Phys. Rev. **D99**, 012004 (2019), 1811.02774
- [10] X.G. Lu et al. (MINERvA), Phys. Rev. Lett. **121**, 022504 (2018), 1805.05486
- [11] R. Gran et al. (MINERvA), Phys. Rev. Lett. **120**, 221805 (2018), 1803.09377
- [12] C.E. Patrick et al. (MINERvA), Phys. Rev. **D97**, 052002 (2018), 1801.01197
- [13] A. Mislivec et al. (MINERvA), Phys. Rev. **D97**, 032014 (2018), 1711.01178
- [14] O. Altinok et al. (MINERvA), Phys. Rev. **D96**, 072003 (2017), 1708.03723
- [15] M. Betancourt et al. (MINERvA), Phys. Rev. Lett. **119**, 082001 (2017), 1705.03791
- [16] L. Ren et al. (MINERvA), Phys. Rev. **D95**, 072009 (2017), [Addendum: Phys. Rev.D97,no.1,019902(2018)], 1701.04857
- [17] C.M. Marshall et al. (MINERvA), Phys. Rev. Lett. **119**, 011802 (2017), 1611.02224
- [18] J. Devan et al. (MINERvA), Phys. Rev. **D94**, 112007 (2016), 1610.04746
- [19] L. Aliaga et al. (MINERvA), Phys. Rev. **D94**, 092005 (2016), [Addendum: Phys. Rev.D95,no.3,039903(2017)], 1607.00704
- [20] Z. Wang et al. (MINERvA), Phys. Rev. Lett. **117**, 061802 (2016), 1606.08890
- [21] C.L. McGivern et al. (MINERvA), Phys. Rev. **D94**, 052005 (2016), 1606.07127
- [22] C.M. Marshall et al. (MINERvA), Phys. Rev. **D94**, 012002 (2016), 1604.03920
- [23] J. Wolcott et al. (MINERvA), Phys. Rev. Lett. **117**, 111801 (2016), 1604.01728

- [24] J. Mousseau et al. (MINERvA), Phys. Rev. **D93**, 071101 (2016), 1601.06313
- [25] J. Park et al. (MINERvA), Phys. Rev. **D93**, 112007 (2016), 1512.07699
- [26] P.A. Rodrigues et al. (MINERvA), Phys. Rev. Lett. **116**, 071802 (2016), [Addendum: Phys. Rev. Lett.121,no.20,209902(2018)], 1511.05944
- [27] J. Wolcott et al. (MINERvA), Phys. Rev. Lett. **116**, 081802 (2016), 1509.05729
- [28] T. Le et al. (MINERvA), Phys. Lett. **B749**, 130 (2015), 1503.02107
- [29] T. Walton et al. (MINERvA), Phys. Rev. **D91**, 071301 (2015), 1409.4497
- [30] A. Higuera et al. (MINERvA), Phys. Rev. Lett. **113**, 261802 (2014), 1409.3835
- [31] B. Eberly et al. (MINERvA), Phys. Rev. **D92**, 092008 (2015), 1406.6415
- [32] B.G. Tice et al. (MINERvA), Phys. Rev. Lett. **112**, 231801 (2014), 1403.2103
- [33] G.A. Fiorentini et al. (MINERvA), Phys. Rev. Lett. **111**, 022502 (2013), 1305.2243
- [34] L. Fields et al. (MINERvA), Phys. Rev. Lett. **111**, 022501 (2013), 1305.2234
- [35] R. Fine, B. Messerly, K.S. McFarland (MINERvA), Snowmass 2021 Letter of Intent (2020), 2009.04548
- [36] R. Brun, F. Rademakers, Nucl. Instrum. Meth. A **389**, 81 (1997)
- [37] C. Andreopoulos et al., Nucl. Instrum. Meth. A **614**, 87 (2010), 0905.2517