NuGraph2

A Graph Neural Network for 3D Reconstruction in Liquid Argon Time Projection Chambers

V Hewes

18th May 2022

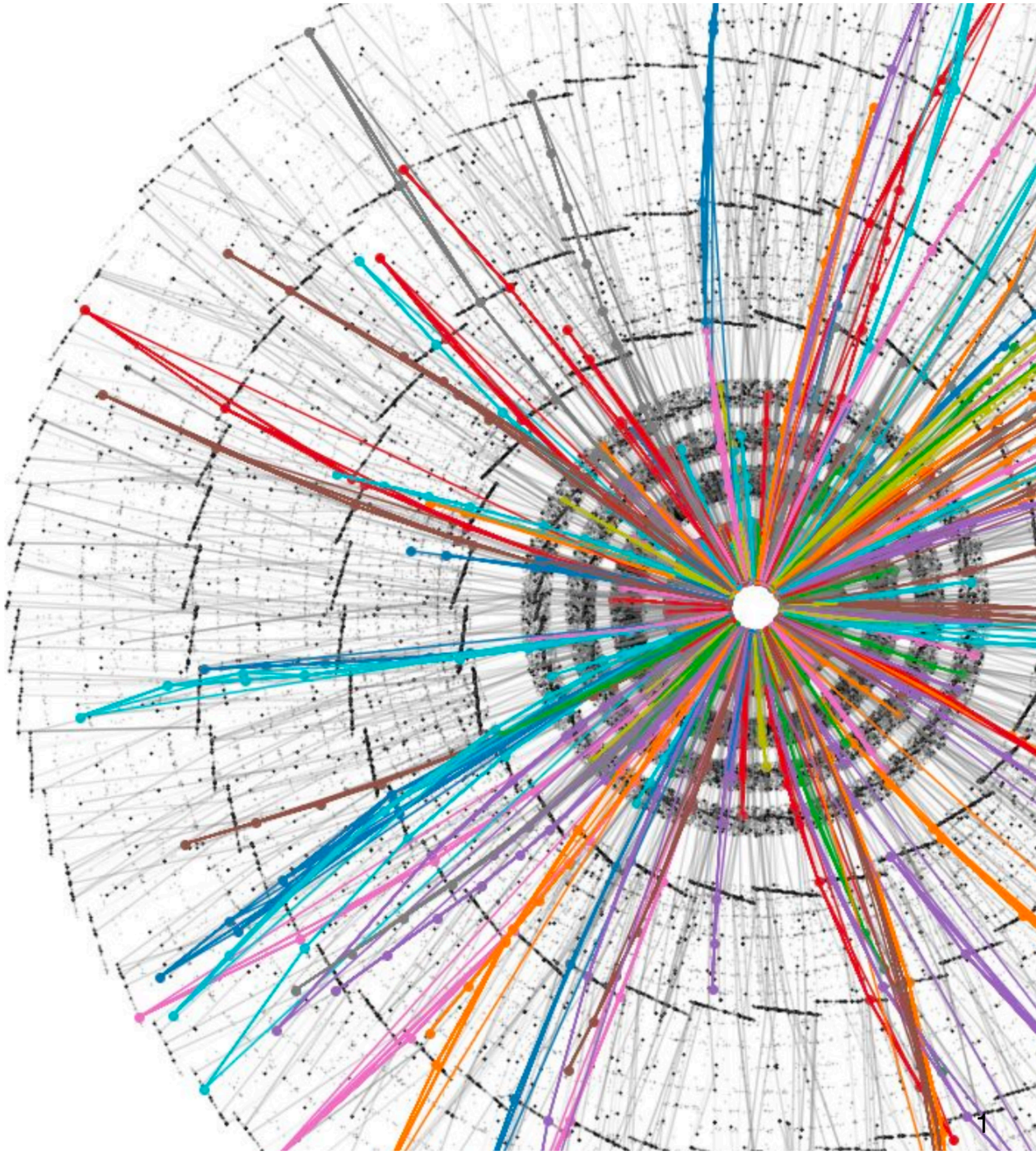
DUNE FD sim/reco meeting

Introduction

- Developing Graph Neural Network (GNN) reconstruction for LArTPCs as part of the Exa.TrkX collaboration.
- This effort has been ongoing for ~5 years, originally motivated by reconstructing atmospheric ν_τ interactions in the DUNE far detector.
- Branched out to other detector technologies, ie. MicroBooNE, with the aim of developing general-purpose reconstruction tools.
- Last DUNE update was exactly two years ago, May 18th 2021!

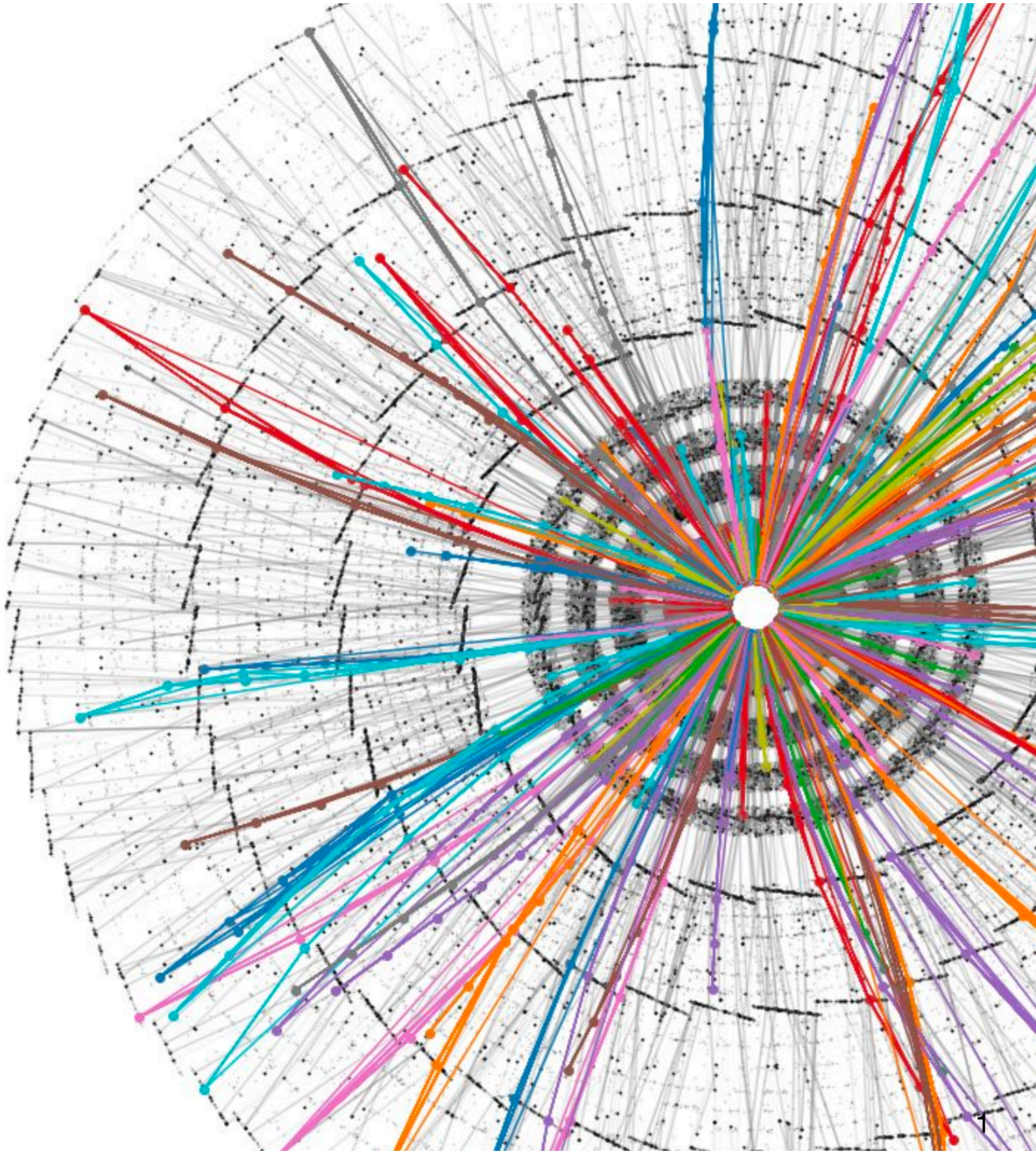
Exa.TrkX

- Exa.TrkX is a collaboration developing next-generation Graph Neural Network (GNN) reconstruction for HEP:



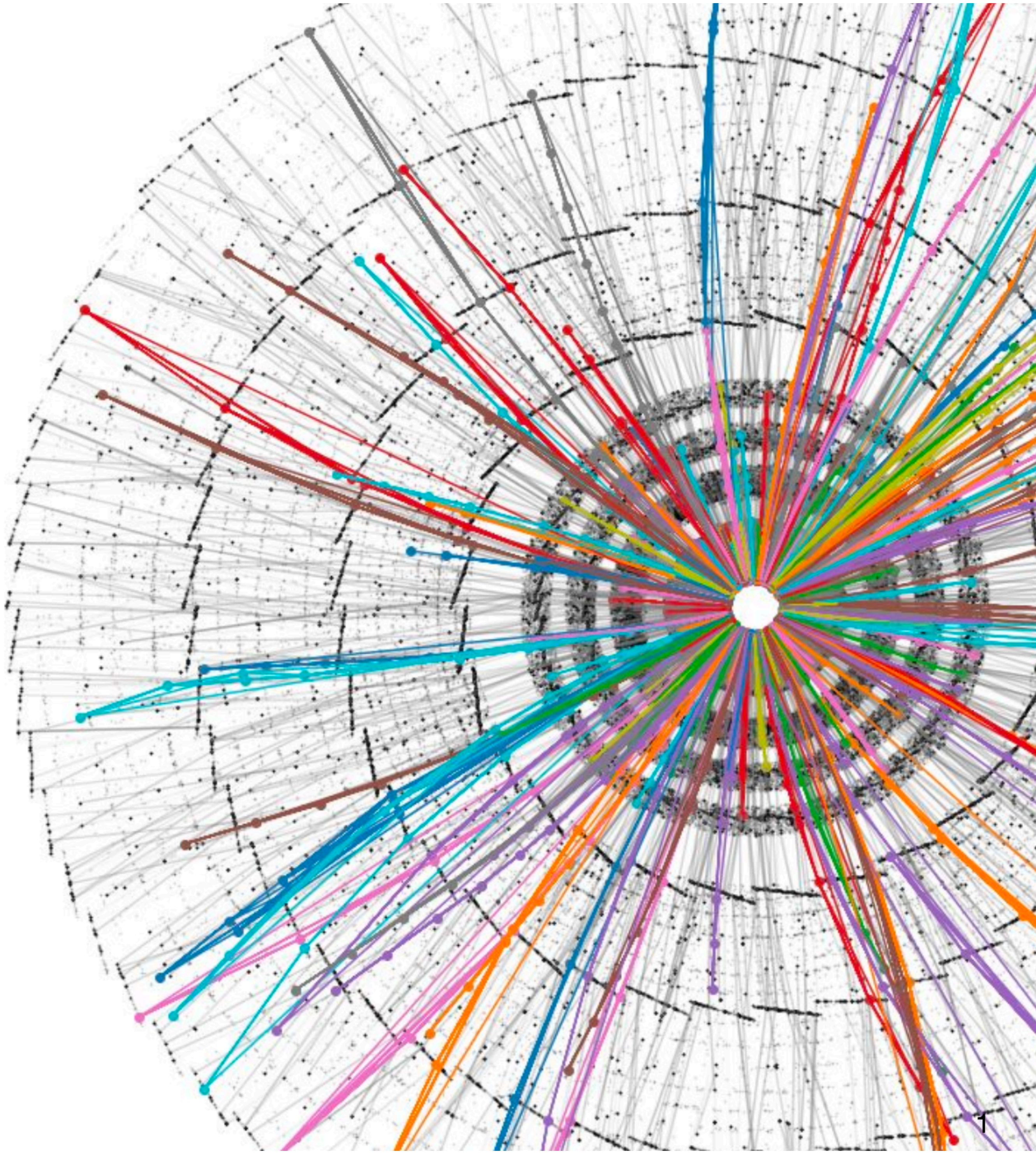
Exa.TrkX

- Exa.TrkX is a collaboration developing next-generation Graph Neural Network (GNN) reconstruction for HEP:
  - **Energy Frontier**
    - Expand on HEP.TrkX's prototype GNN for HL-LHC.
    - Incorporate into ATLAS's simulation and validation chain.



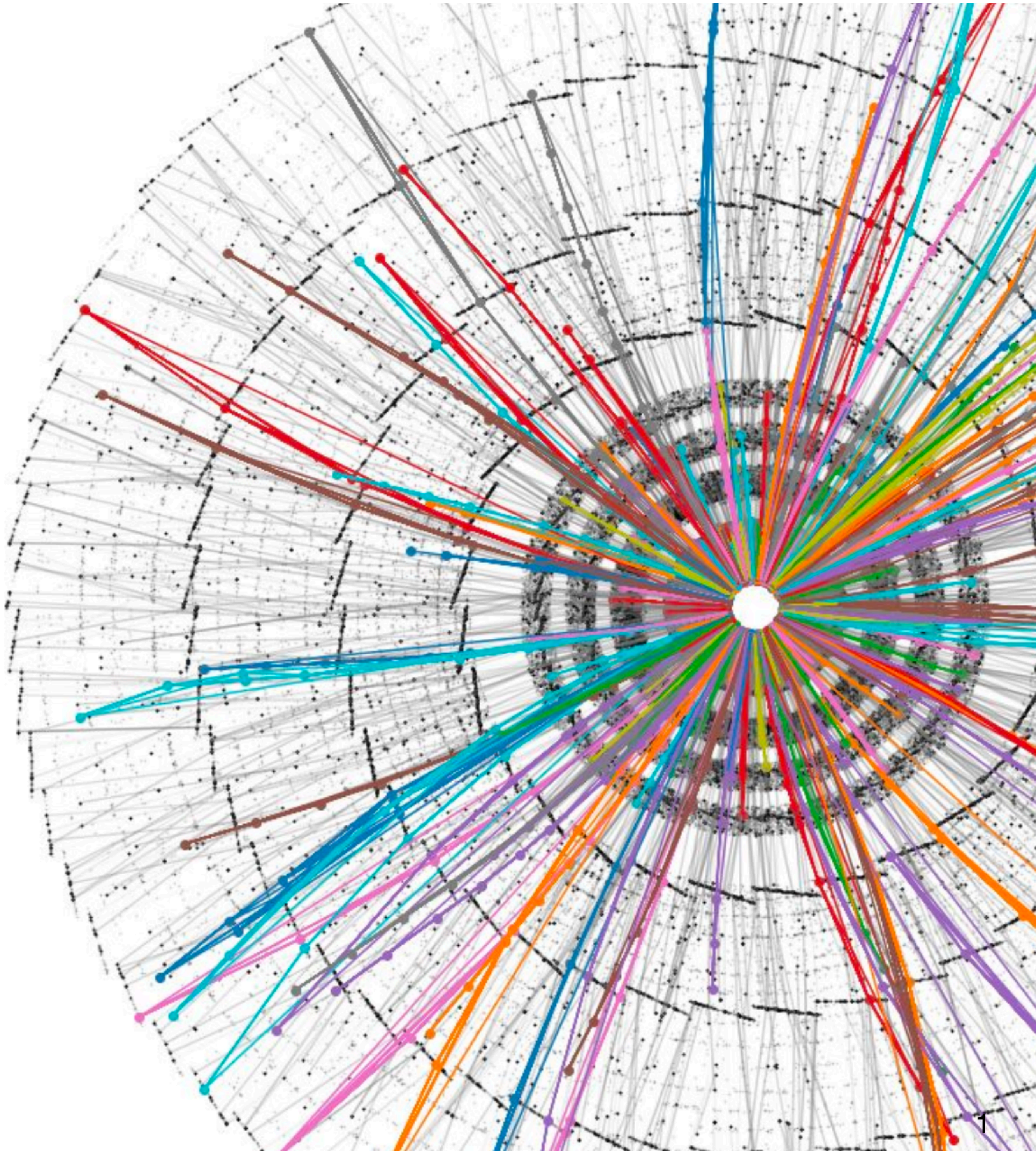
Exa.TrkX

- Exa.TrkX is a collaboration developing next-generation Graph Neural Network (GNN) reconstruction for HEP:
  - **Energy Frontier**
    - Expand on HEP.TrkX's prototype GNN for HL-LHC.
    - Incorporate into ATLAS's simulation and validation chain.
  - **Intensity Frontier**
    - Explore viability of HEP.TrkX network for neutrino physics.
    - Develop GNN-based reconstruction for Liquid Argon TPCs.

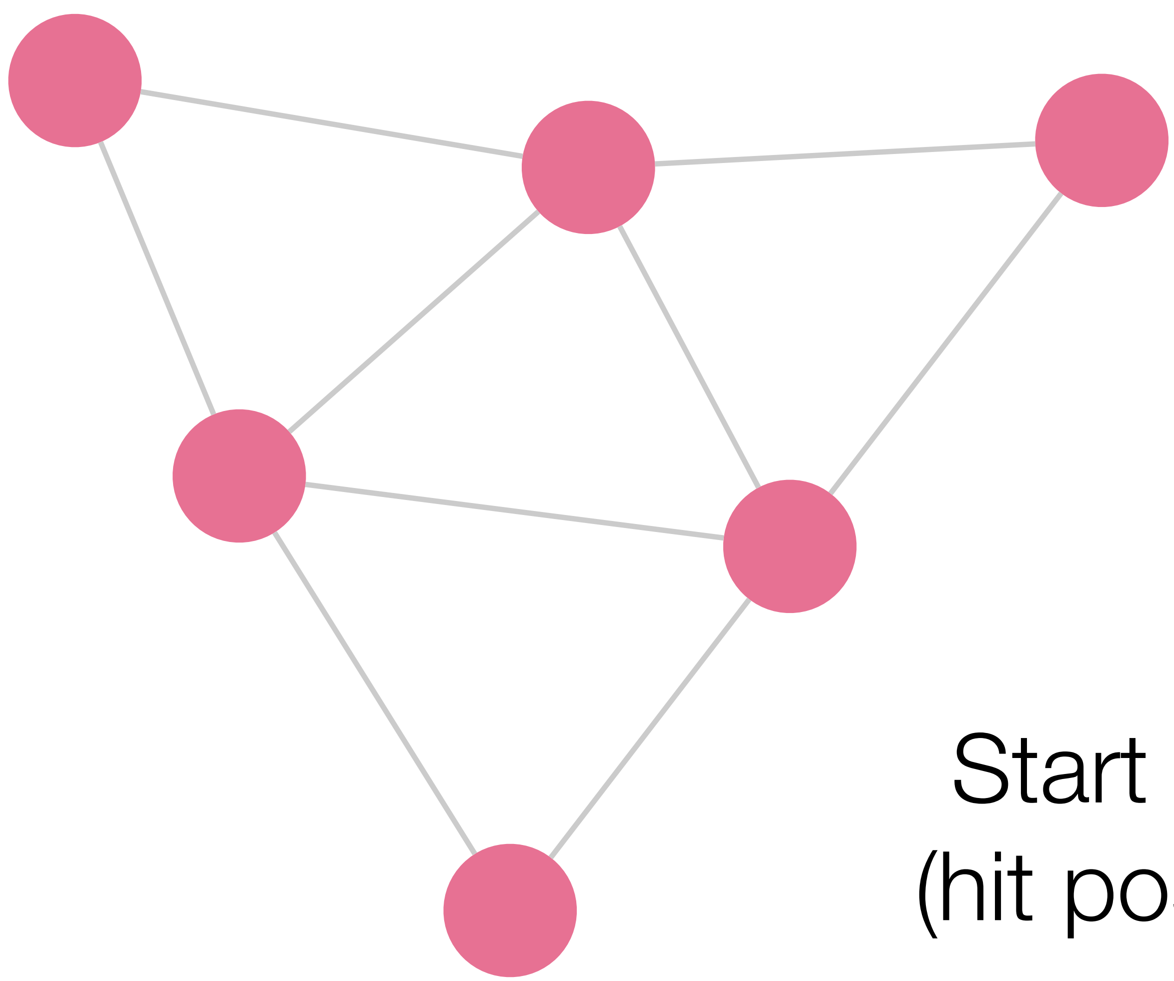


Exa.TrkX

- Exa.TrkX is a collaboration developing next-generation Graph Neural Network (GNN) reconstruction for HEP:
  - **Energy Frontier**
    - Expand on HEP.TrkX's prototype GNN for HL-LHC.
    - Incorporate into ATLAS's simulation and validation chain.
  - **Intensity Frontier**
    - Explore viability of HEP.TrkX network for neutrino physics.
    - Develop GNN-based reconstruction for Liquid Argon TPCs.
- **See Paulo Calafiura's overview talk!**

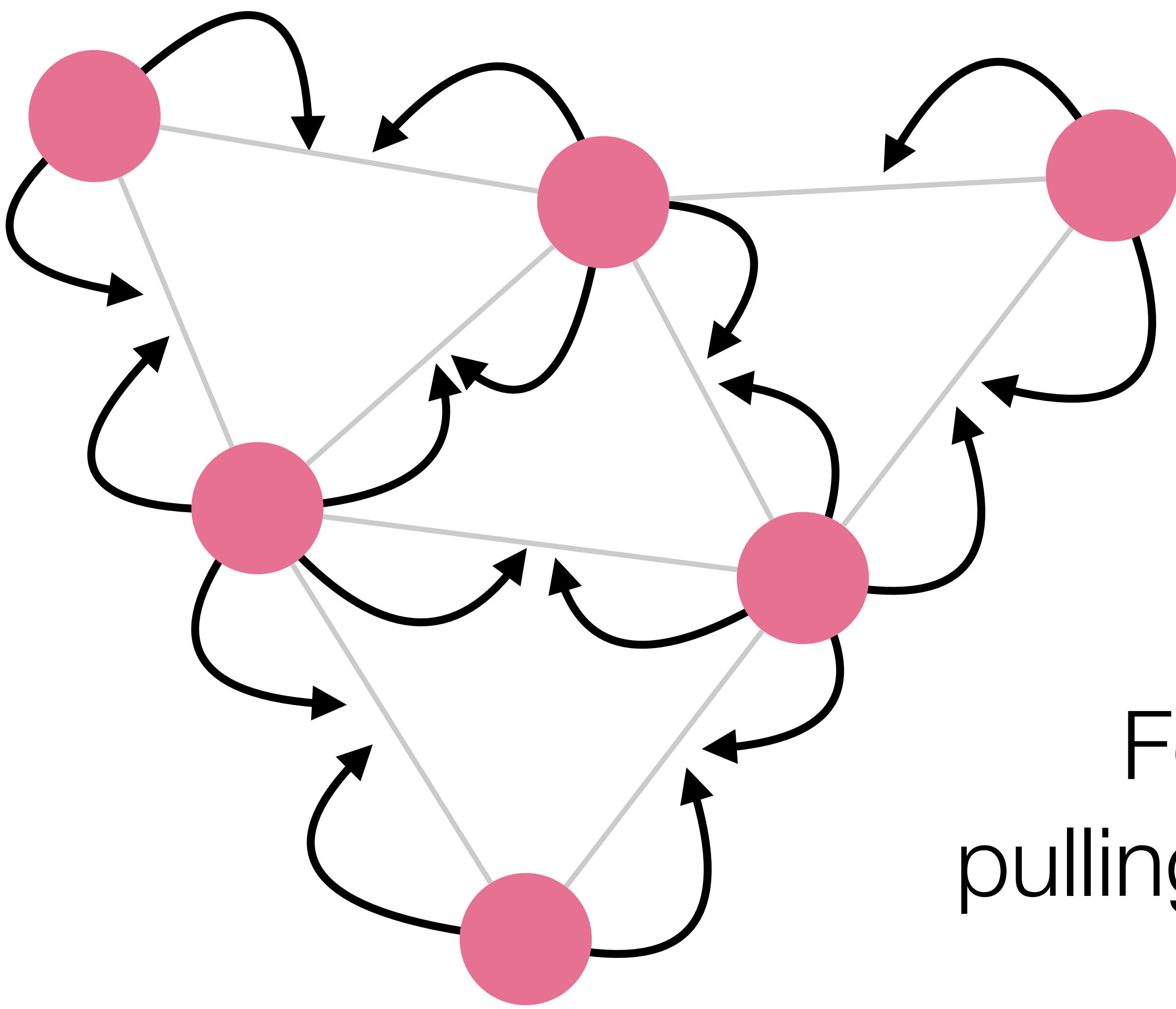


Message-passing network



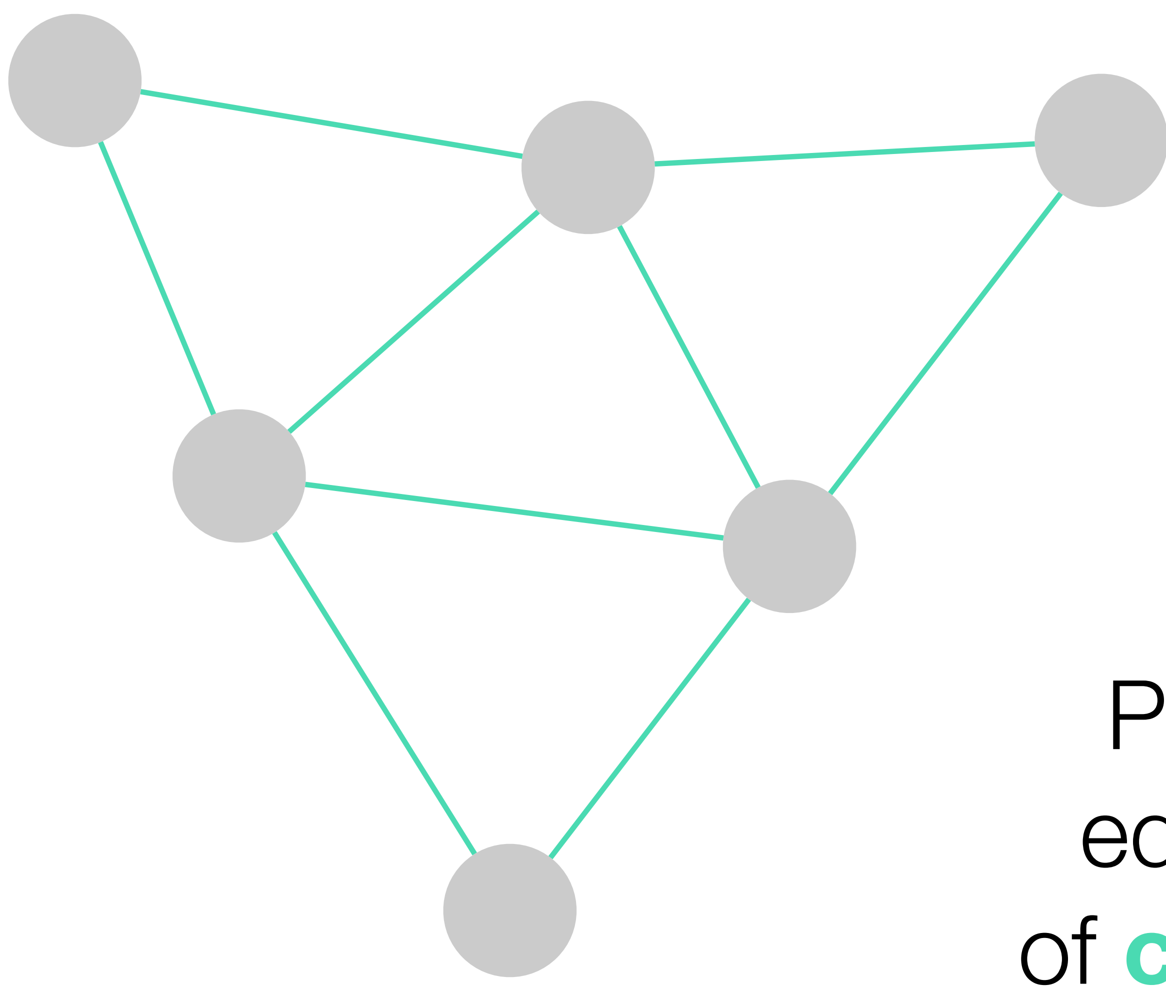
Start with graph **node features**
(hit position, amplitude, RMS, etc)

Message-passing network



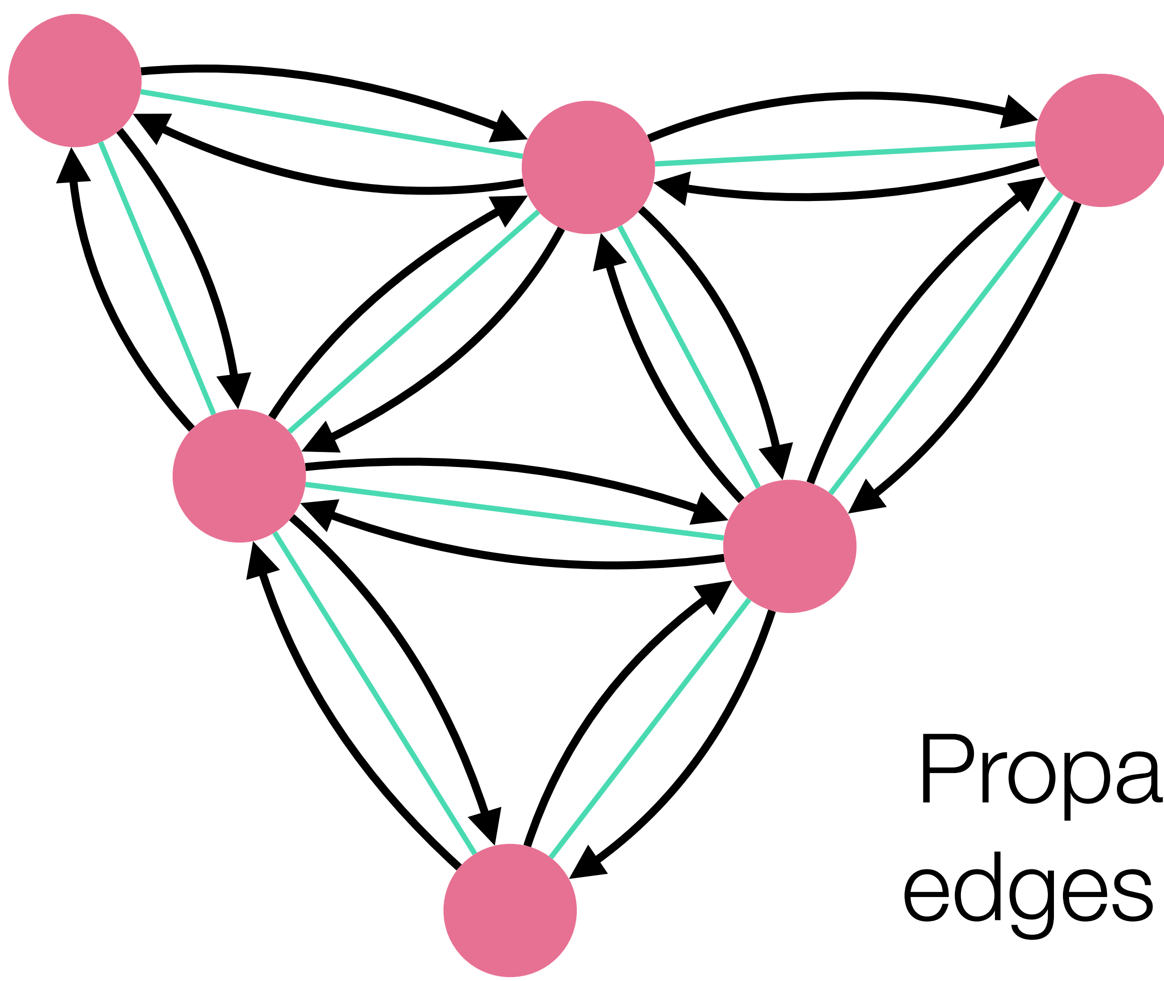
Form **edge features** by pulling in features from incoming and outgoing nodes

Message-passing network



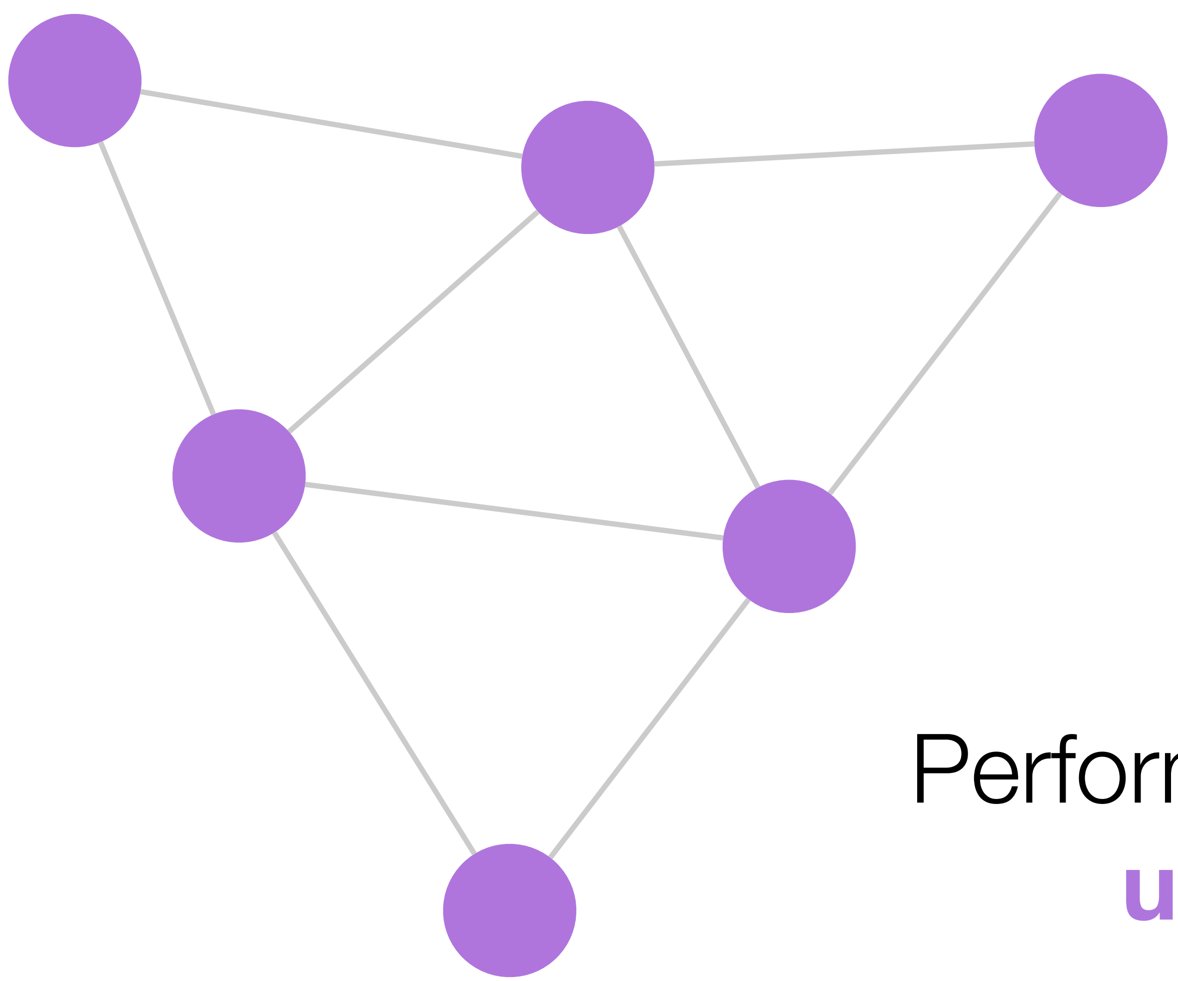
Perform convolutions on edge scores to form a set of **class-wise probabilities**

Message-passing network



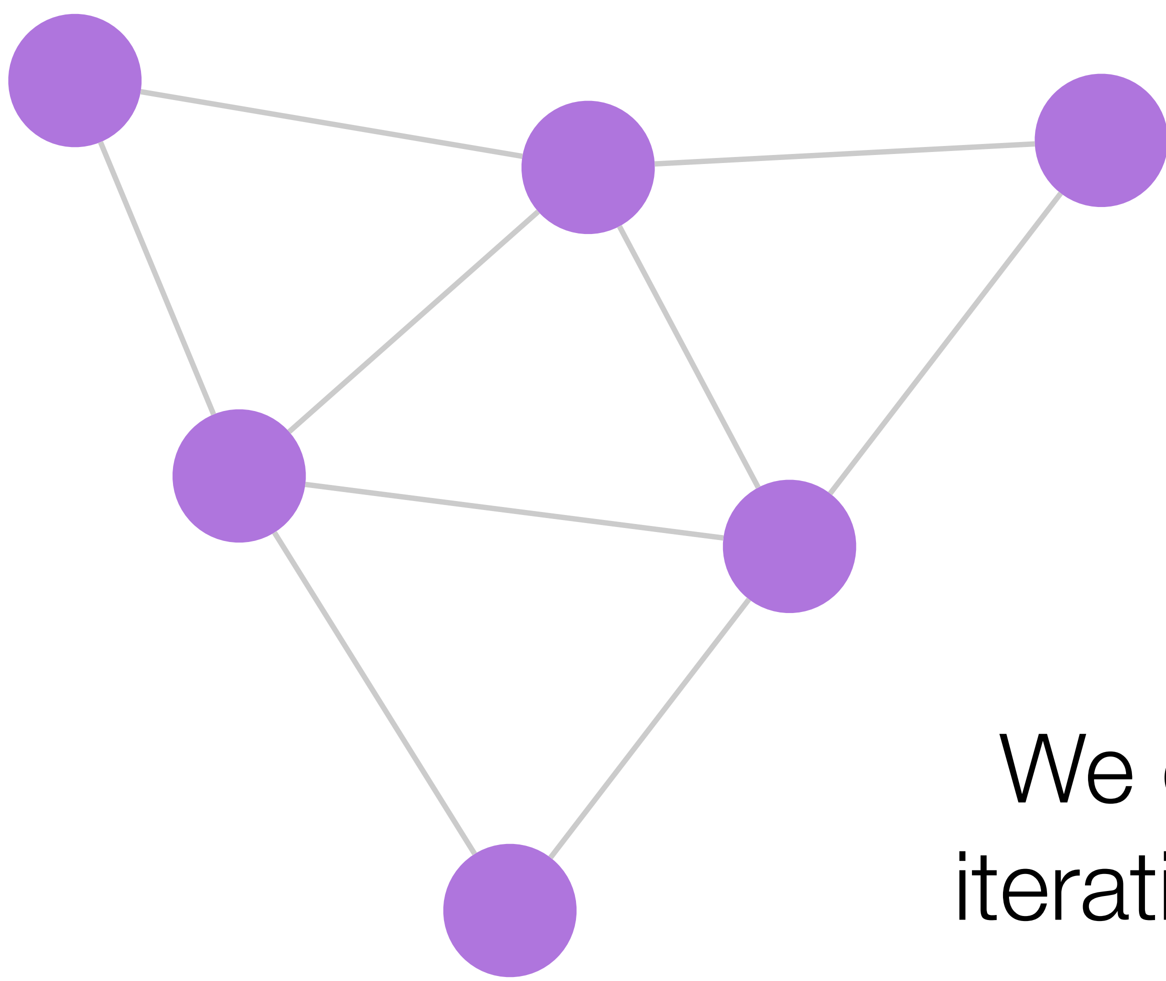
Propagate **node features** across edges, weighting by **edge scores**

Message-passing network



Perform convolutions on nodes to **update node features**

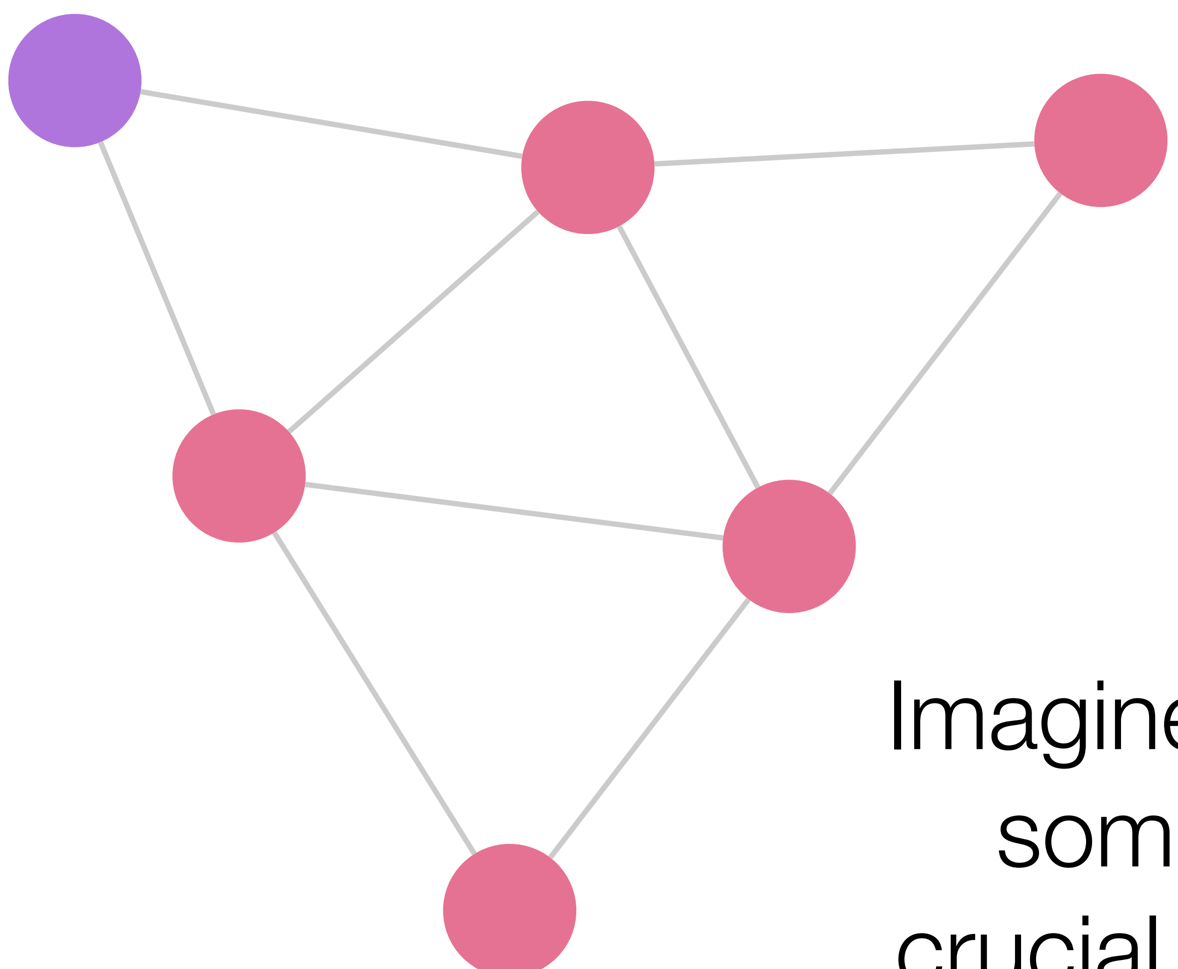
Message-passing network



We can repeat this procedure iteratively to **"pass messages"**

Message-passing network

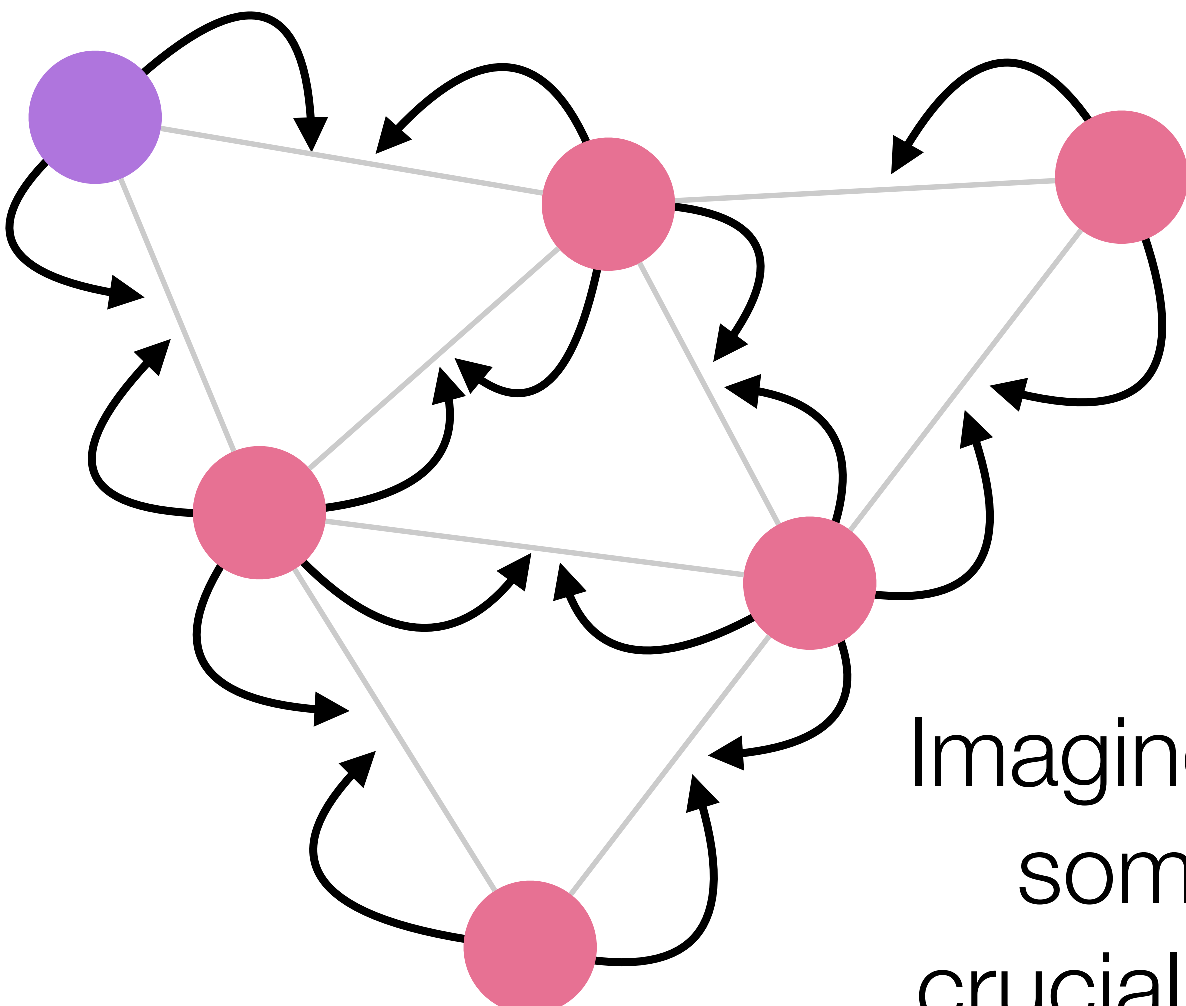
0 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

Message-passing network

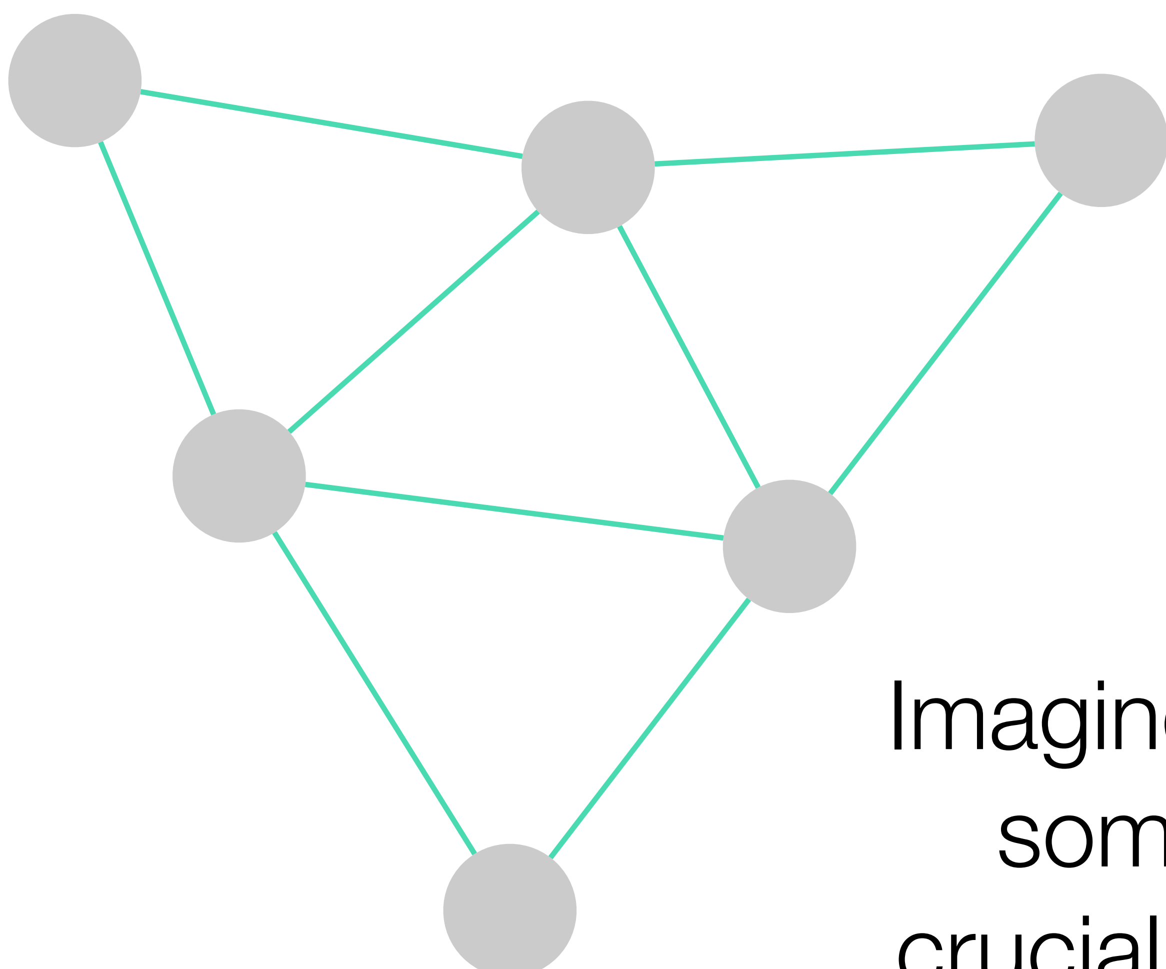
0 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

Message-passing network

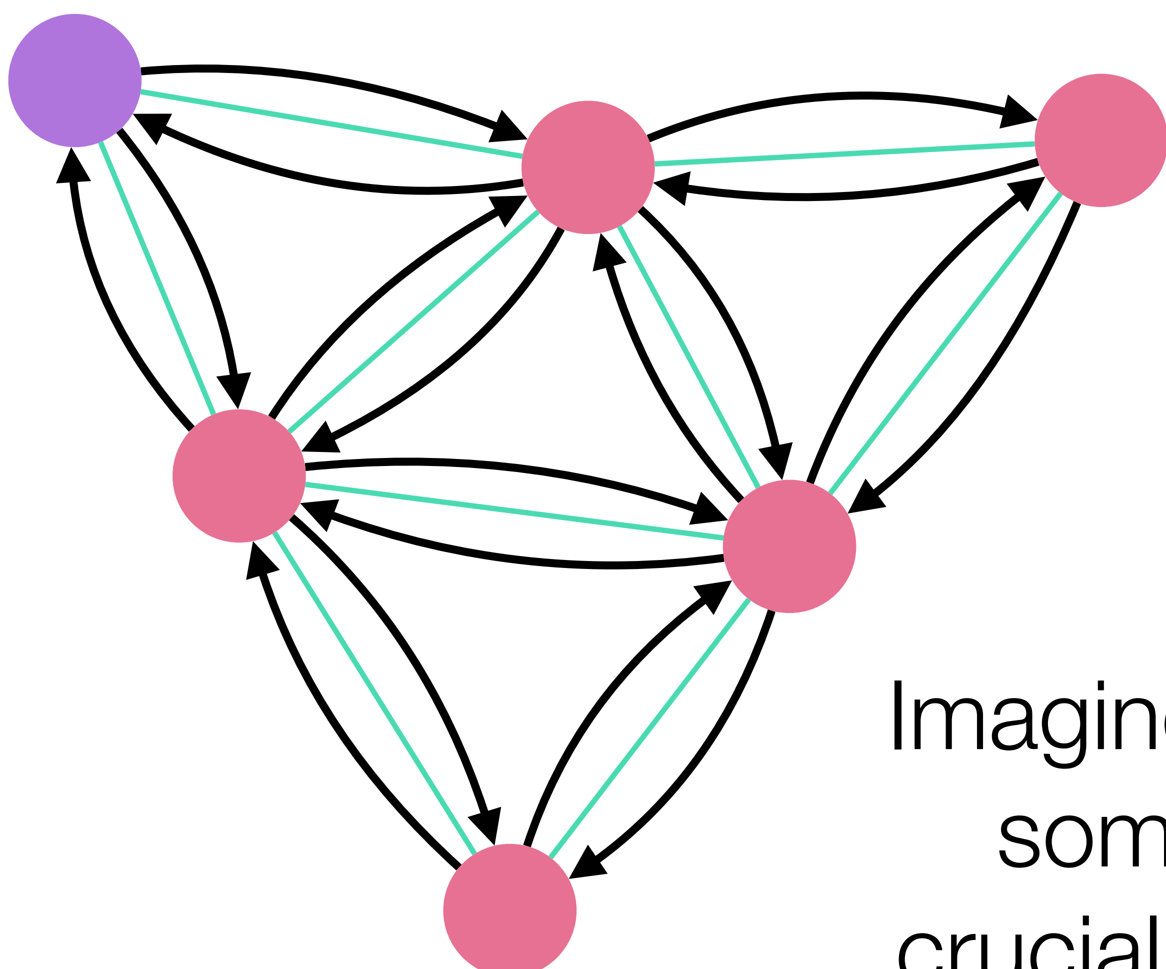
0 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

Message-passing network

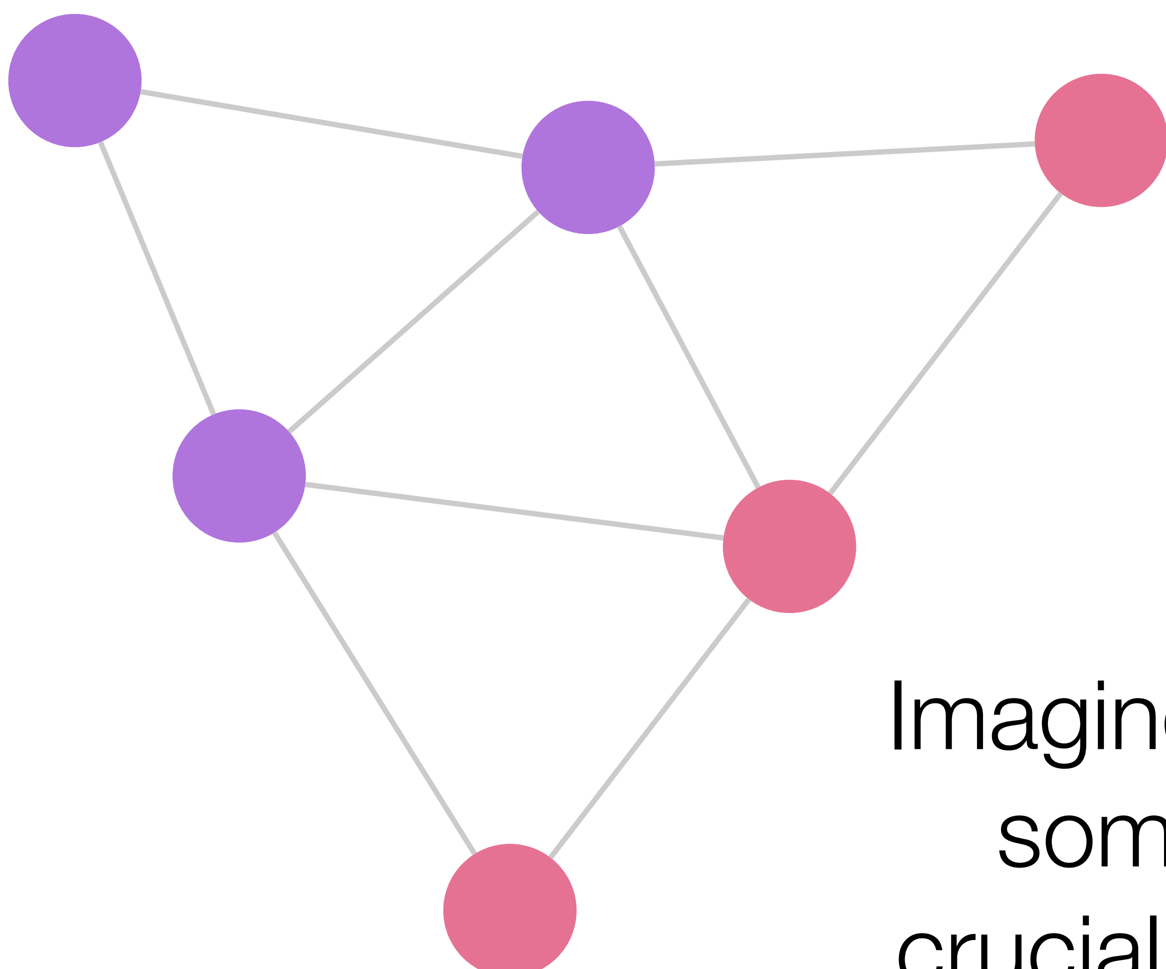
0 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

Message-passing network

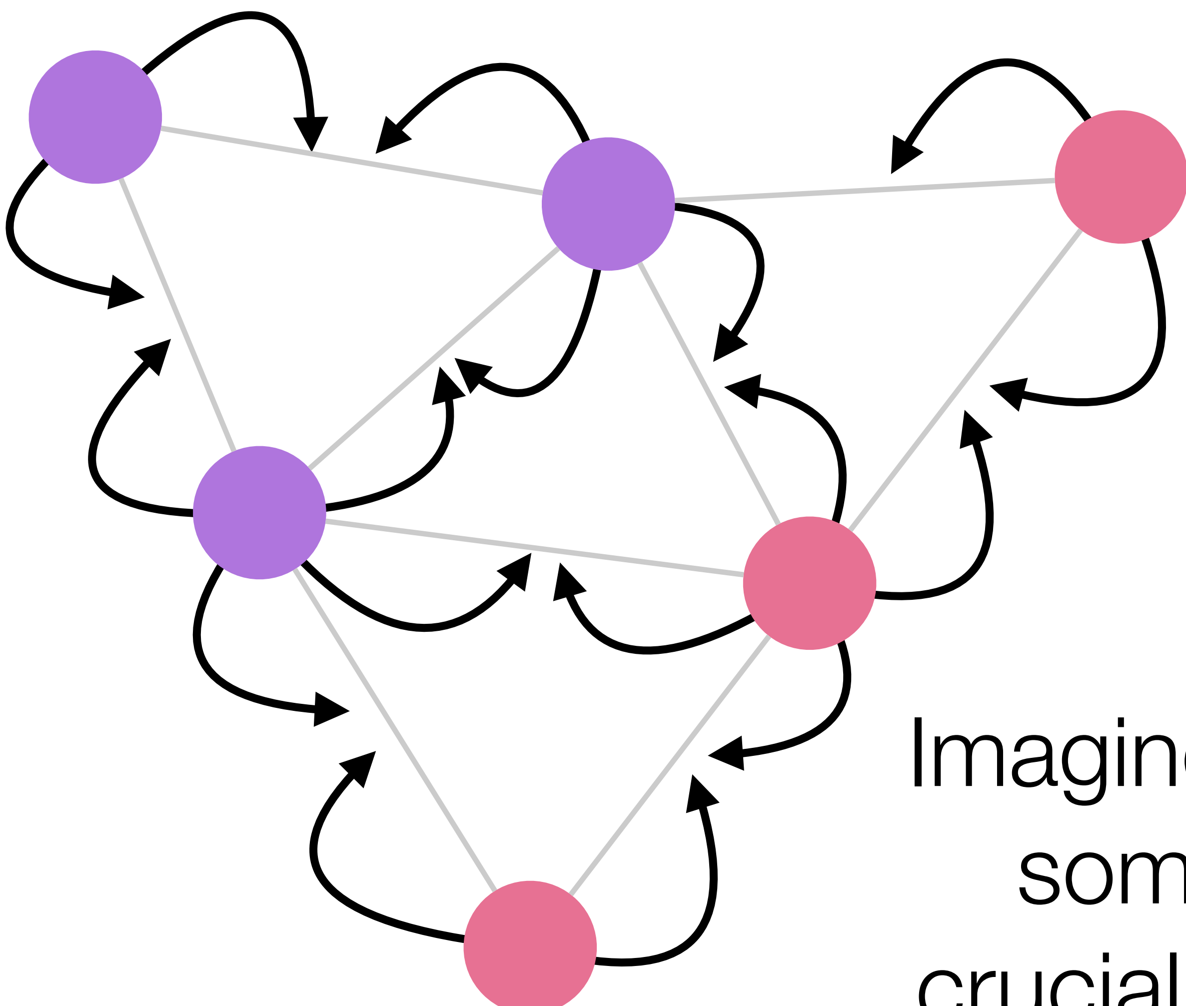
1 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

Message-passing network

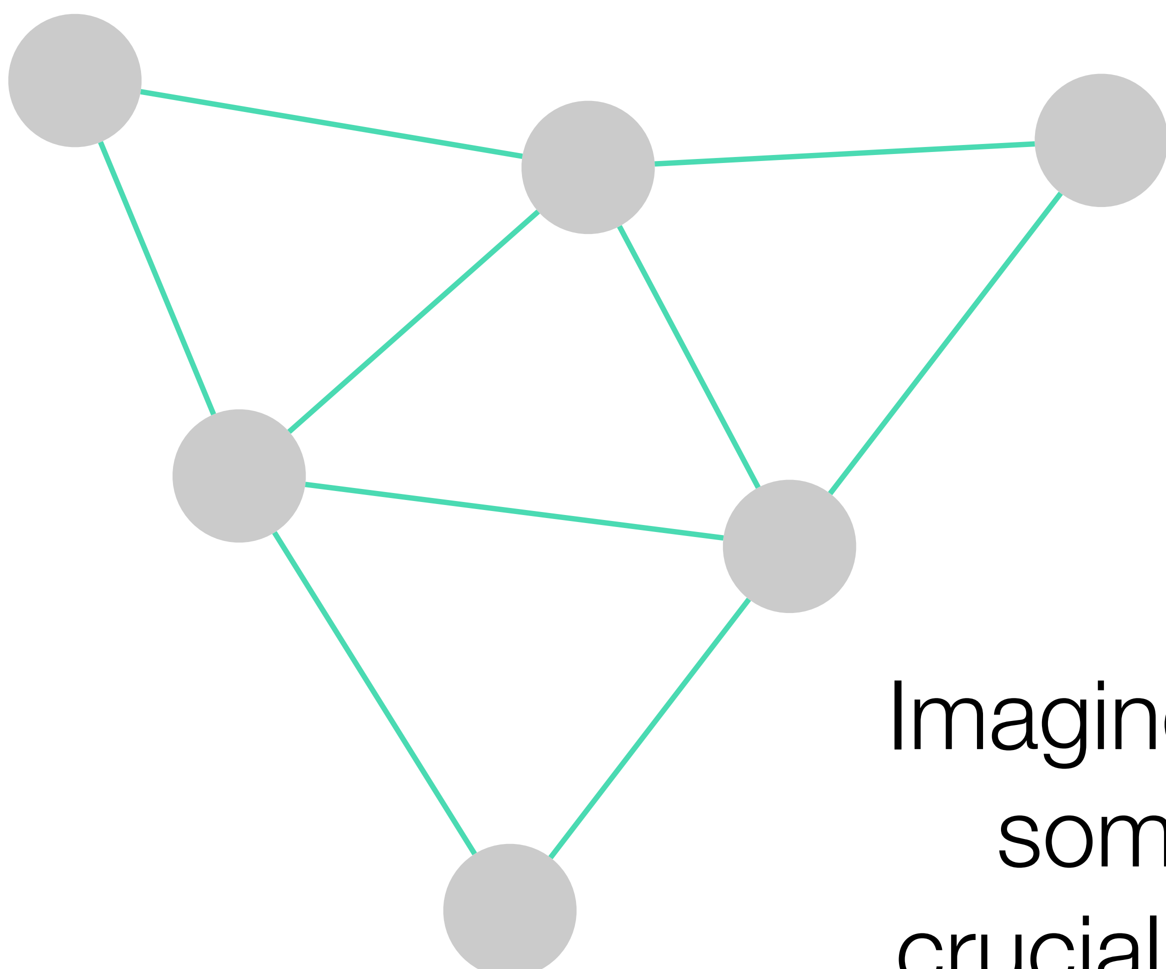
1 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

Message-passing network

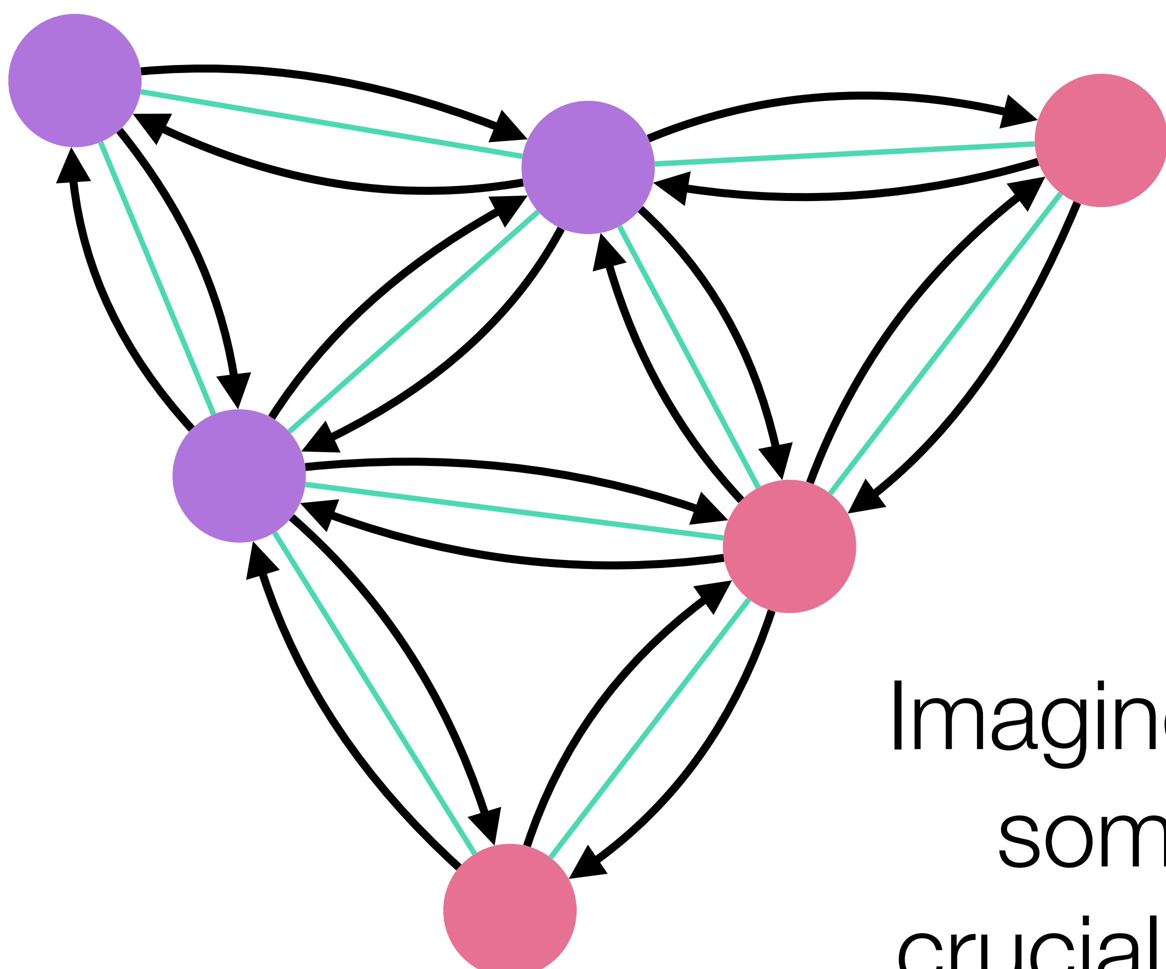
1 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

Message-passing network

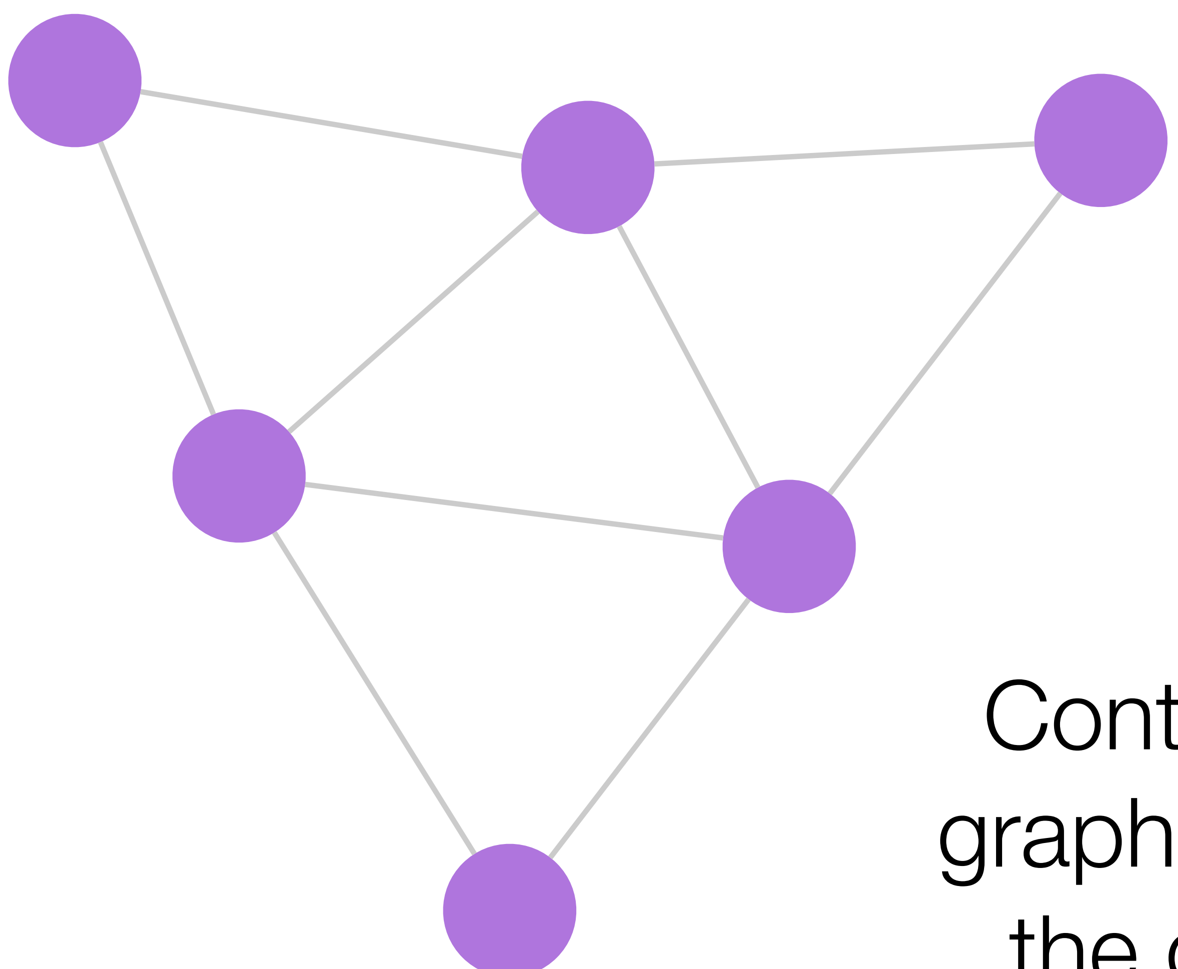
1 iterations



Imagine that one graph node holds some **key information** that's crucial for understanding the event

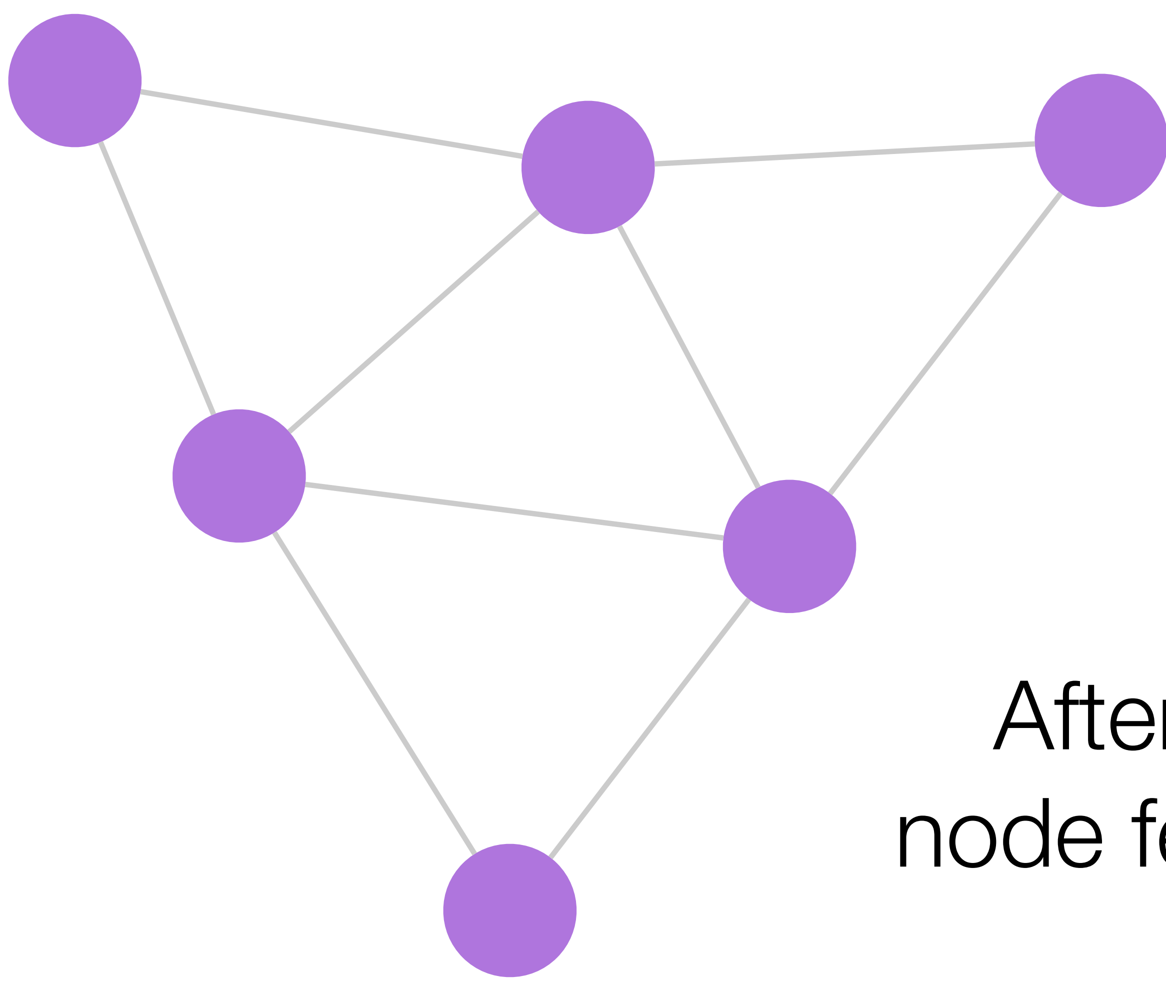
Message-passing network

2 iterations



Context information from each graph node is dispersed through the graph with each iteration.

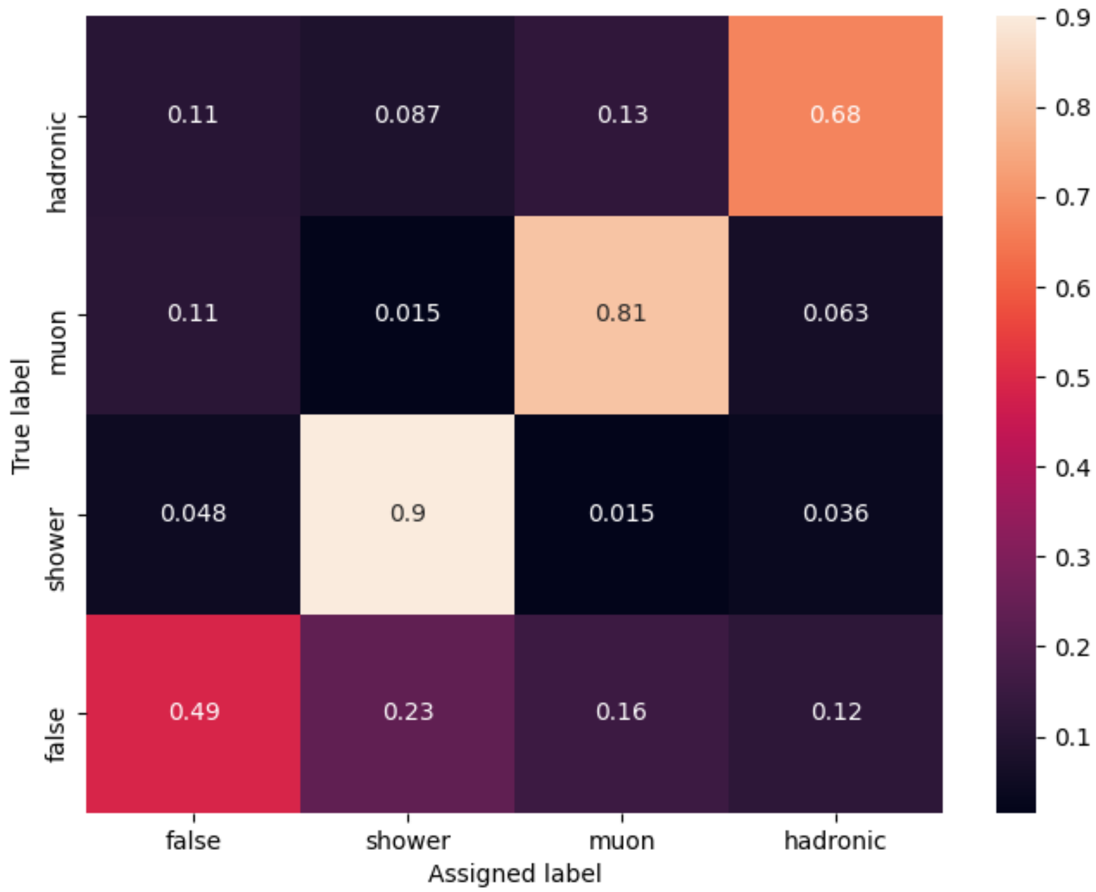
Message-passing network

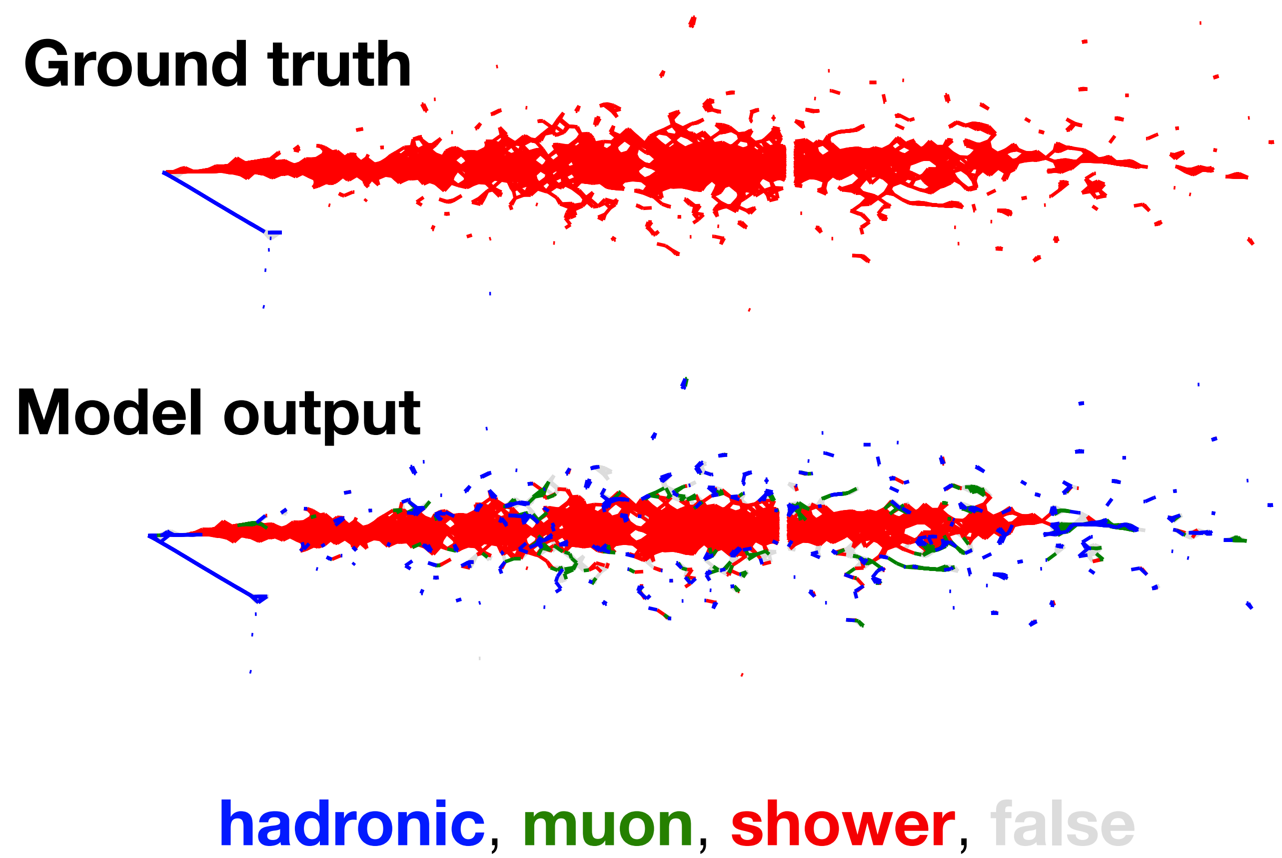


After multiple iterations, graph node features can be convolved to **label graph nodes**

NuGraph1

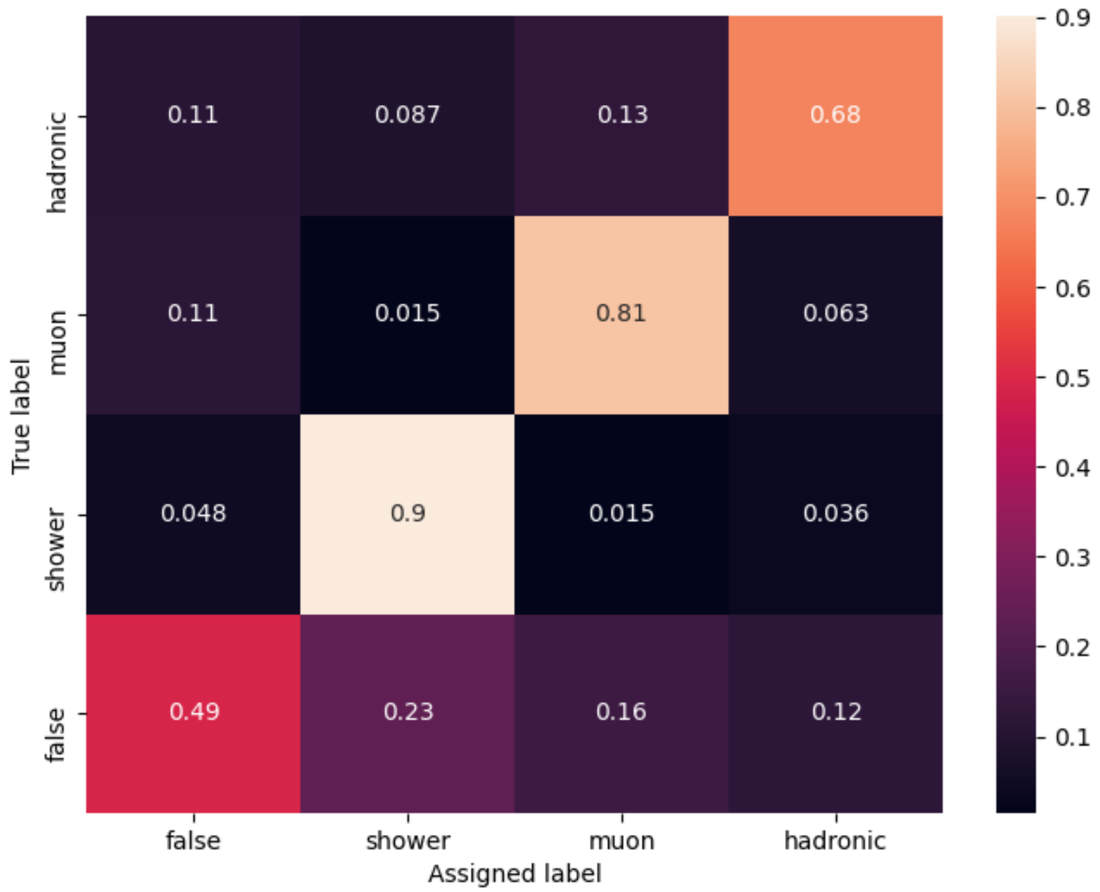
- First proof-of-concept model achieved 84% accuracy in **classifying graph edges**.
  - Reasonable performance on showers, struggled to correctly identify type of track.
  - See arxiv:2103.06233.

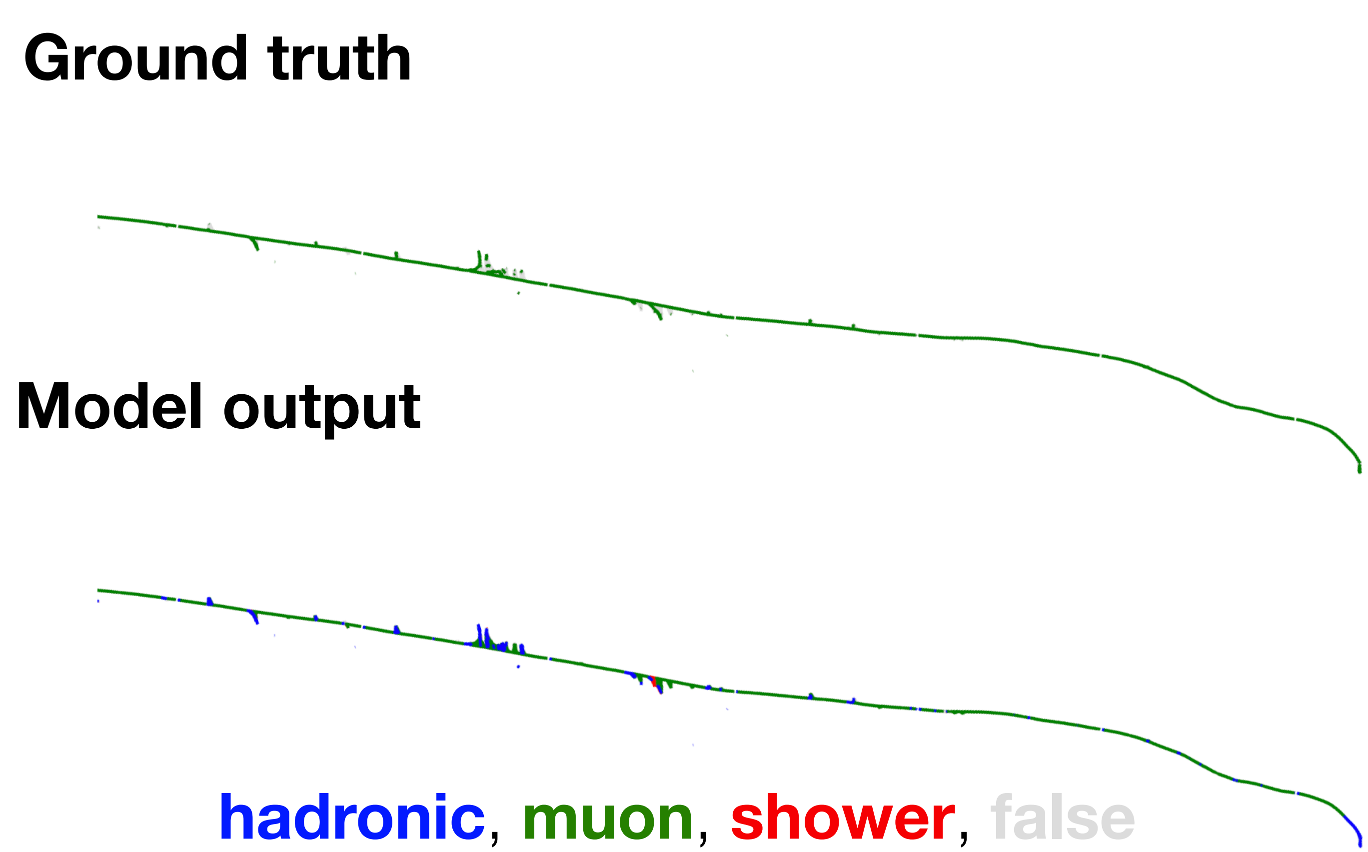




NuGraph1

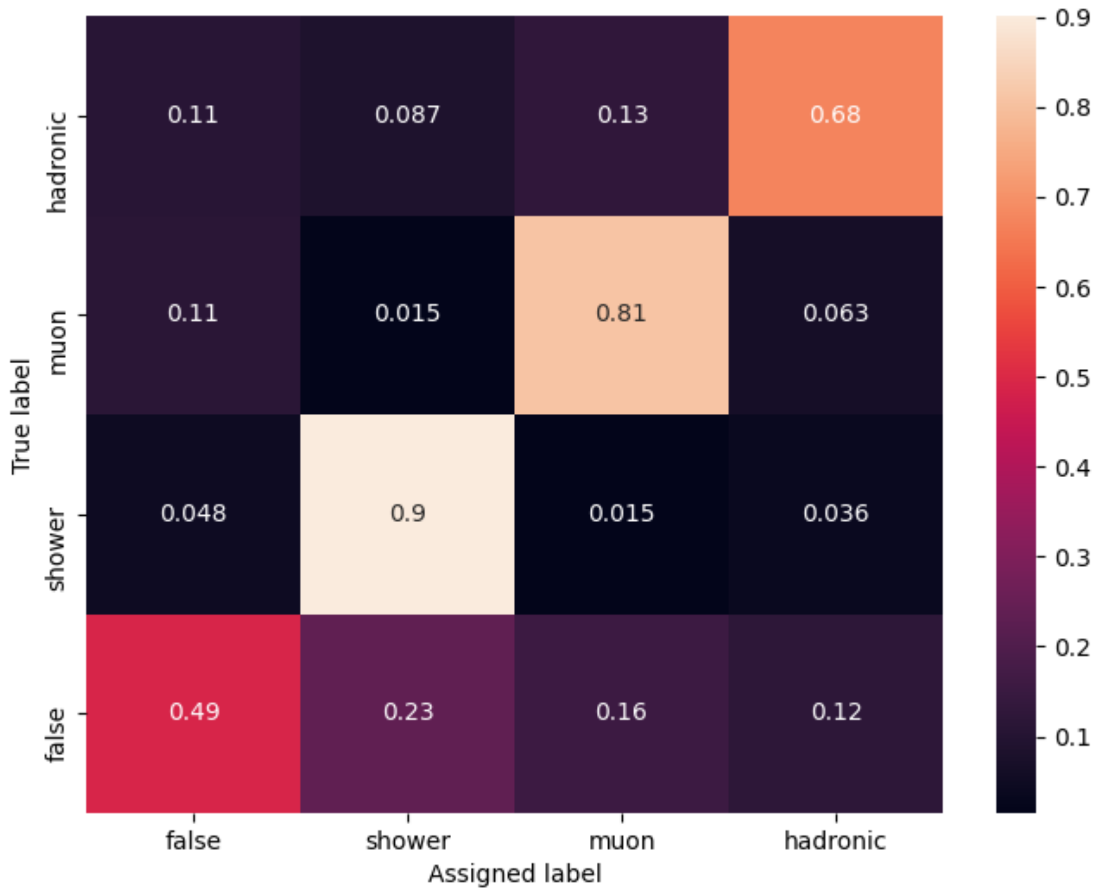
- First proof-of-concept model achieved 84% accuracy in **classifying graph edges**.
  - Reasonable performance on showers, struggled to correctly identify type of track.
  - See arxiv:2103.06233.

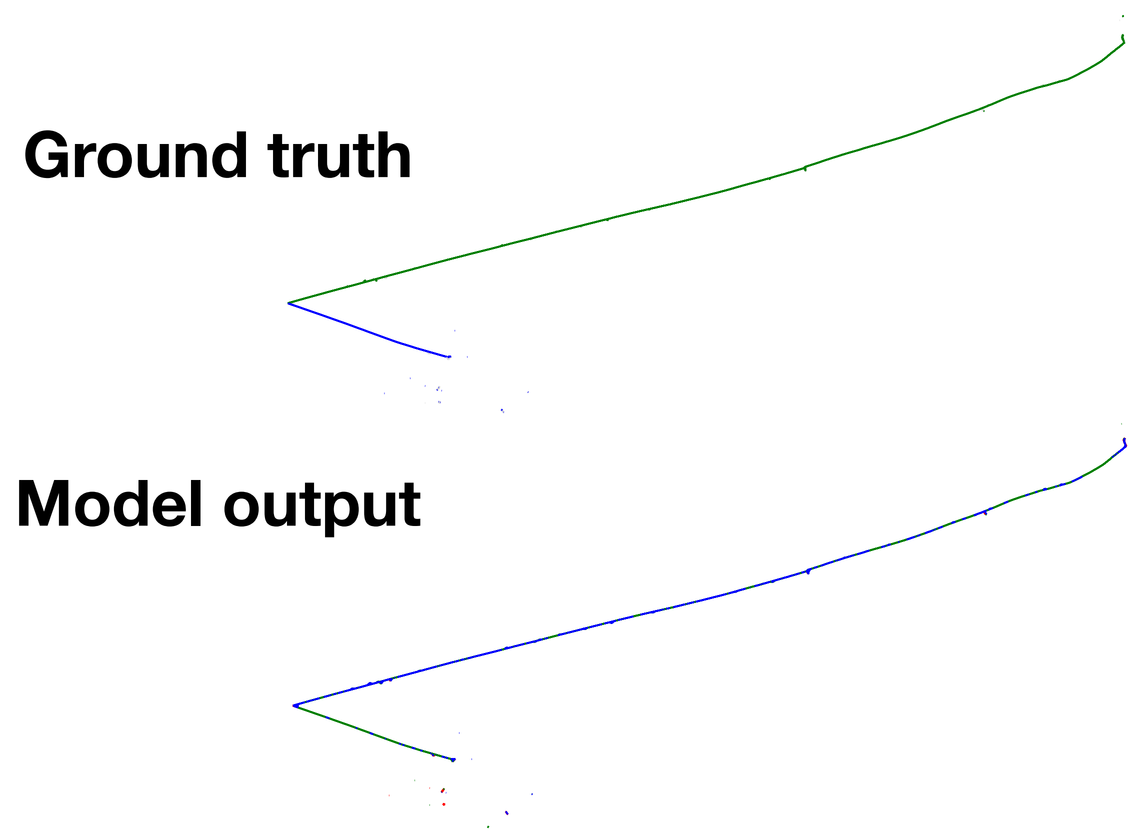




NuGraph1

- First proof-of-concept model achieved 84% accuracy in **classifying graph edges**.
  - Reasonable performance on showers, struggled to correctly identify type of track.
  - See arxiv:2103.06233.





hadronic, **muon**, **shower**, **false**

Second-generation model

- **Second-generation model** incorporated a wide range of improvements over first proof-of-concept.

- Move from **edge classification** to **node (ie. hit) classification**.
  - Graph edge classification for track forming is a natural choice for LHC detectors, where sequential layers provide a natural constraint on edges.
  - Dense LArTPC environment provides no such constraints, and number of edges explodes.

University of CINCINNATI

Second-generation model

- **Second-generation model** incorporated a wide range of improvements over first proof-of-concept.

- Move from **edge classification** to **node (ie. hit) classification**.
  - Graph edge classification for track forming is a natural choice for LHC detectors, where sequential layers provide a natural constraint on edges.
  - Dense LArTPC environment provides no such constraints, and number of edges explodes.

- Introduce **more sophisticated semantic labelling** which considers a wider variety of particle types, ie. EM showers, Michel electrons, diffuse EM activity.

Second-generation model

- **Second-generation model** incorporated a wide range of improvements over first proof-of-concept.

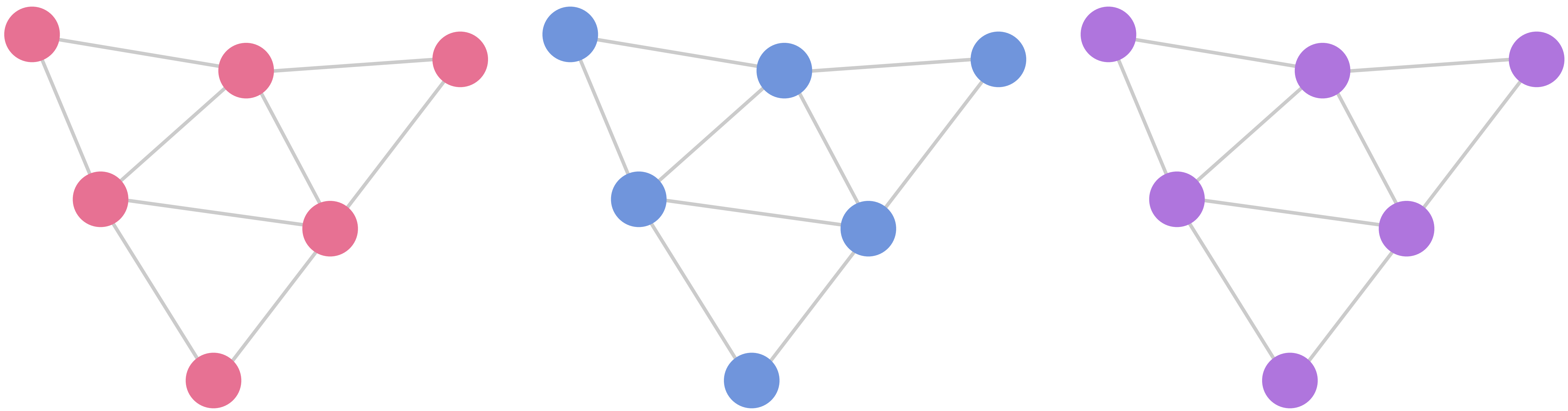
- Move from **edge classification** to **node (ie. hit) classification**.
  - Graph edge classification for track forming is a natural choice for LHC detectors, where sequential layers provide a natural constraint on edges.
  - Dense LArTPC environment provides no such constraints, and number of edges explodes.

- Introduce **more sophisticated semantic labelling** which considers a wider variety of particle types, ie. EM showers, Michel electrons, diffuse EM activity.

- Build a model which **classifies all views simultaneously**, instead of classifying each detector views as an independent event.
  - Furthermore, allow **information exchange between 2D views** to break degeneracies.

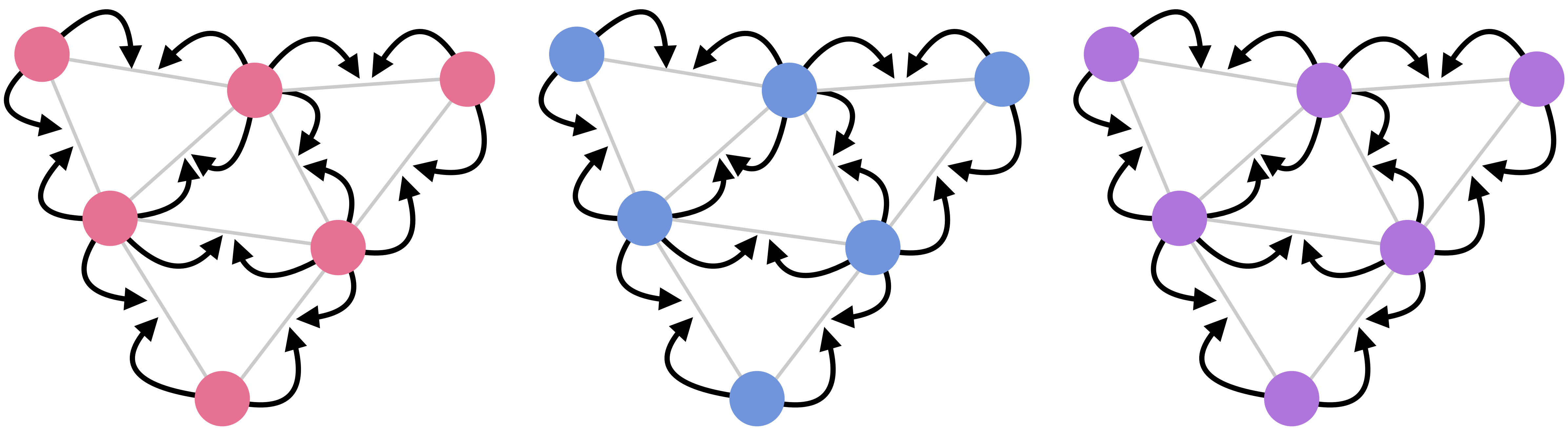
3D Nexus convolutions

- Perform message-passing independently in each detector view.



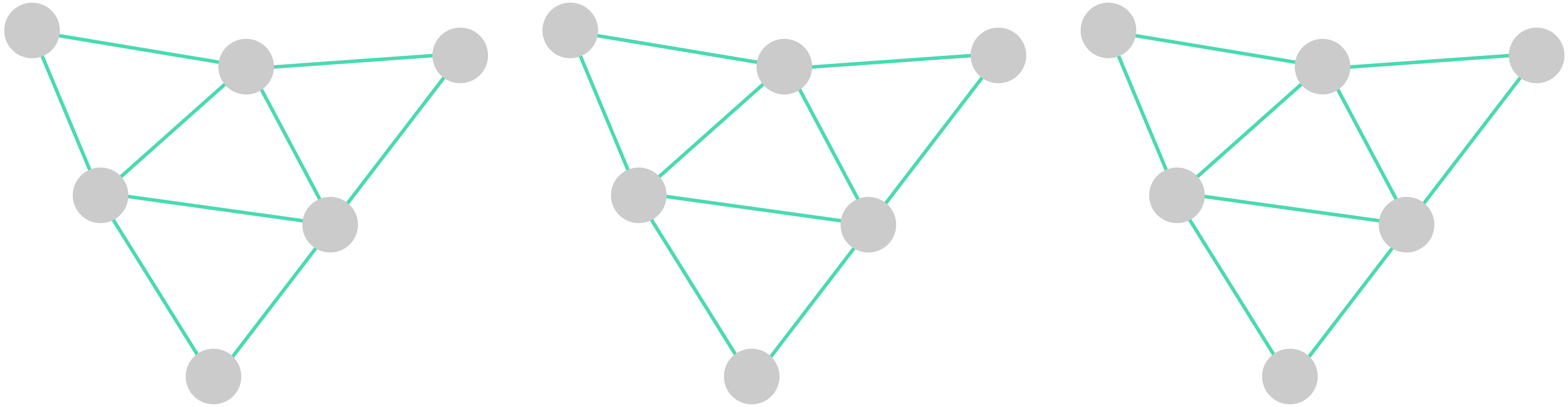
3D Nexus convolutions

- Perform message-passing independently in each detector view.



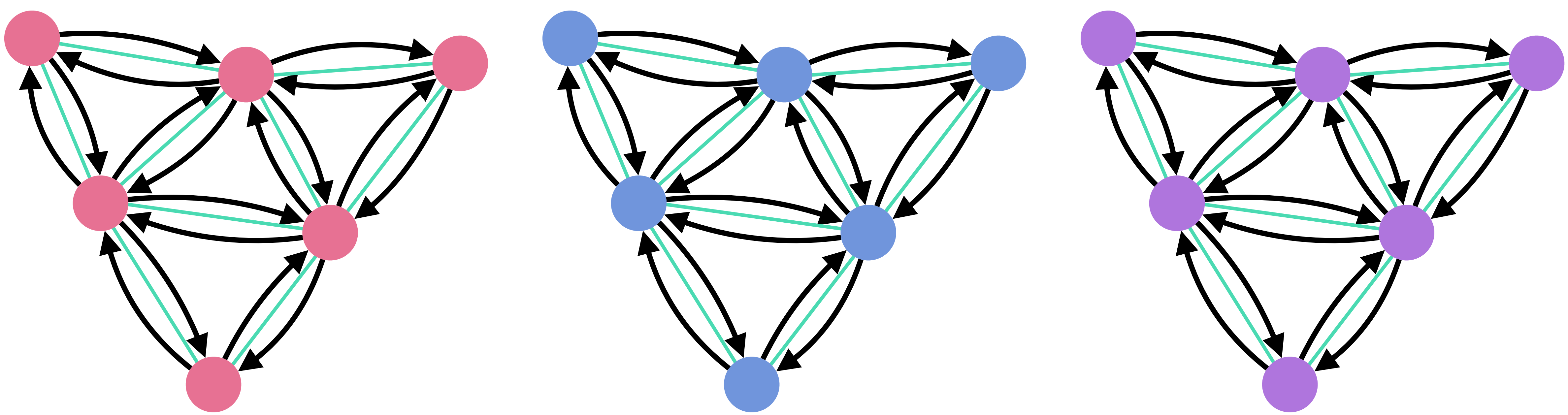
3D Nexus convolutions

- Perform message-passing independently in each detector view.



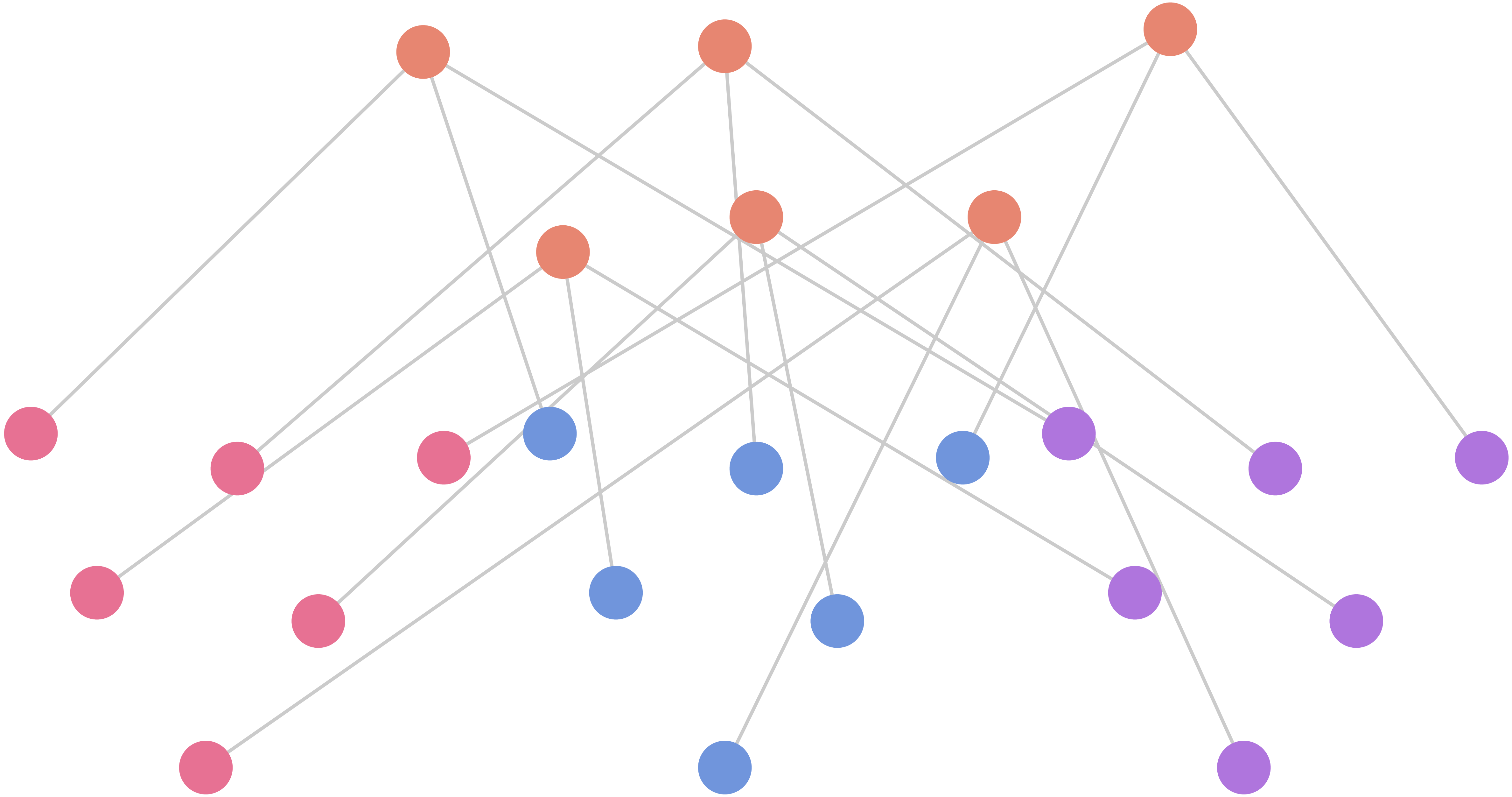
3D Nexus convolutions

- Perform message-passing independently in each detector view.



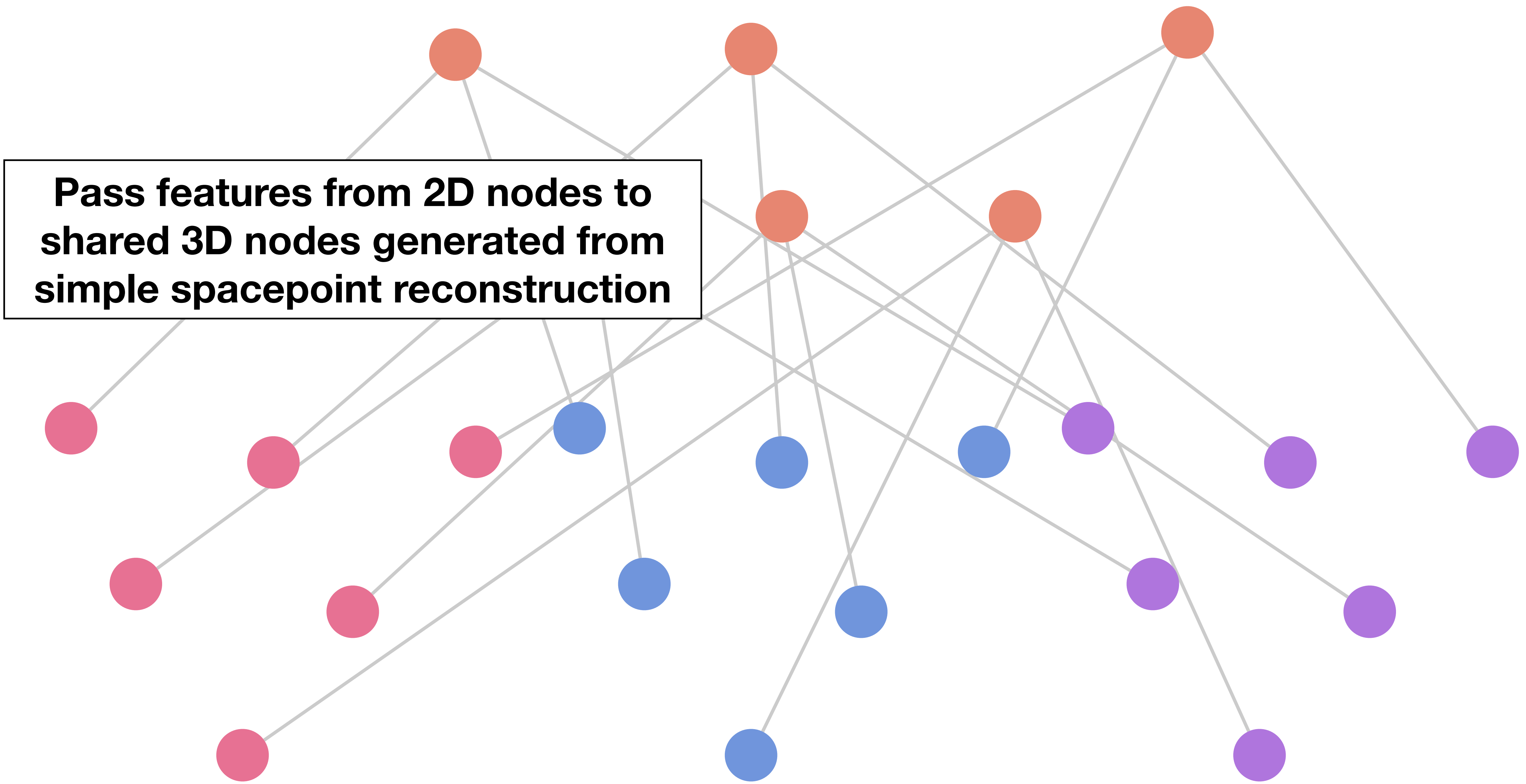
3D Nexus convolutions

- Add additional 3D step to the standard message-passing loop.



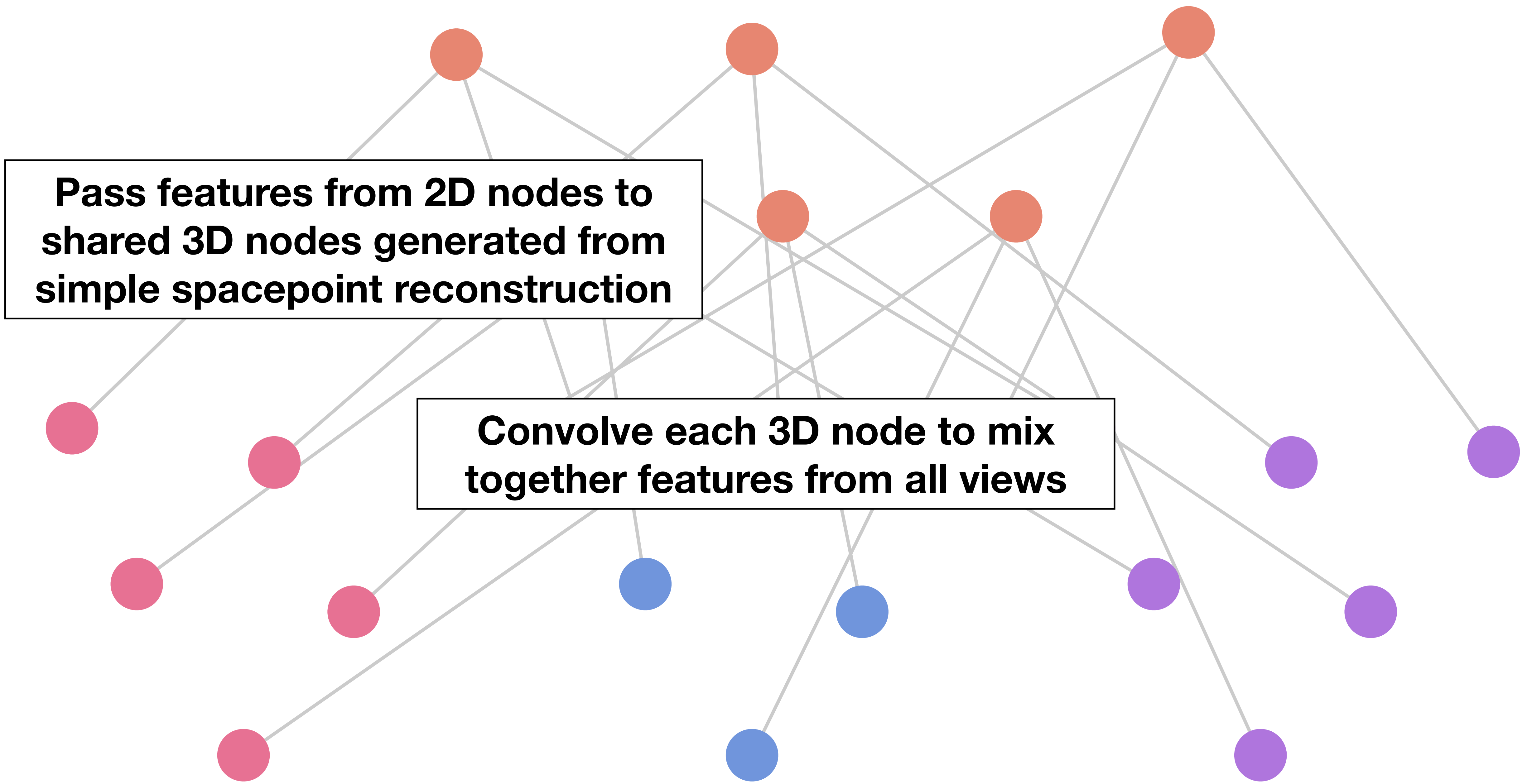
3D Nexus convolutions

- Add additional 3D step to the standard message-passing loop.



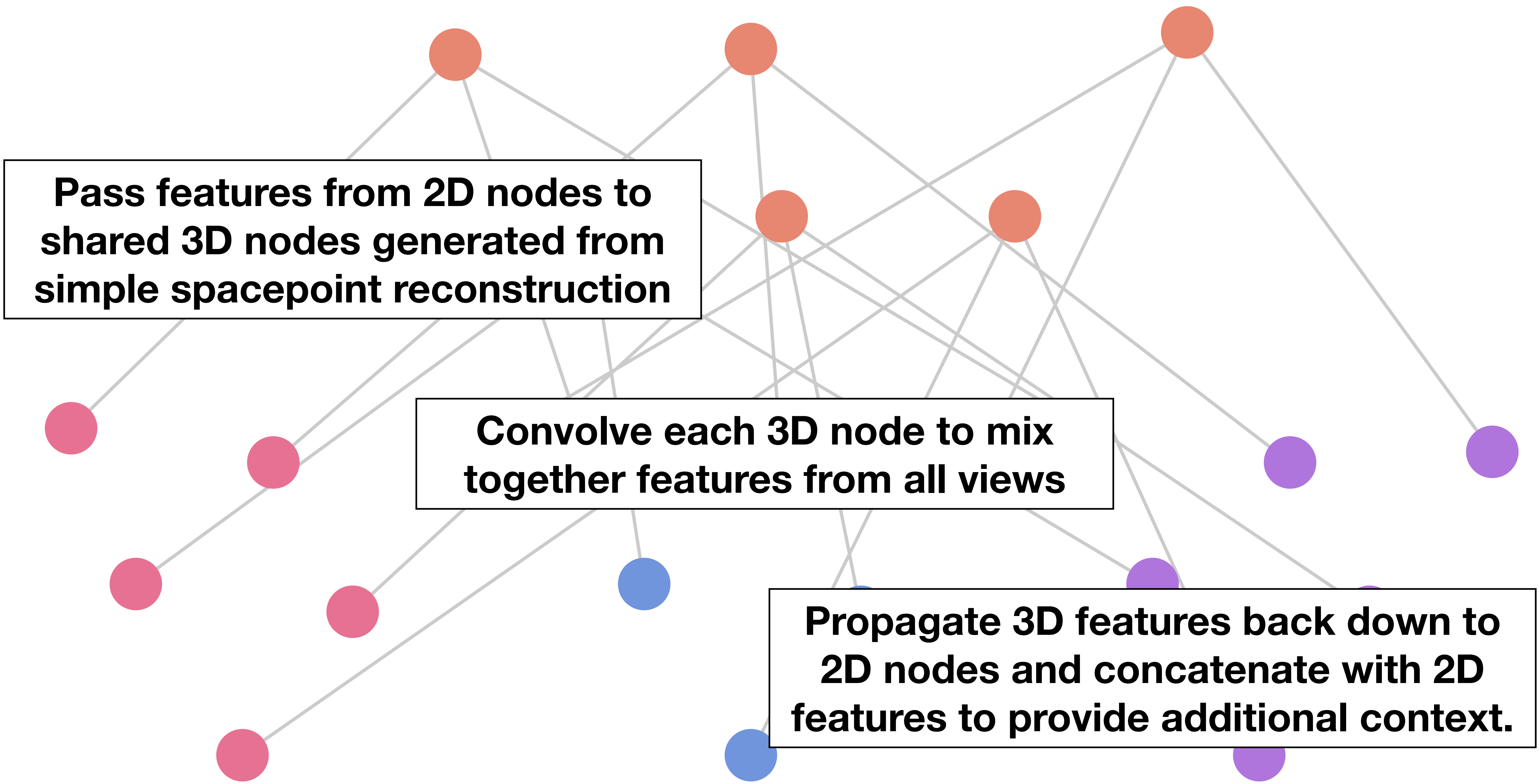
3D Nexus convolutions

- Add additional 3D step to the standard message-passing loop.



3D Nexus convolutions

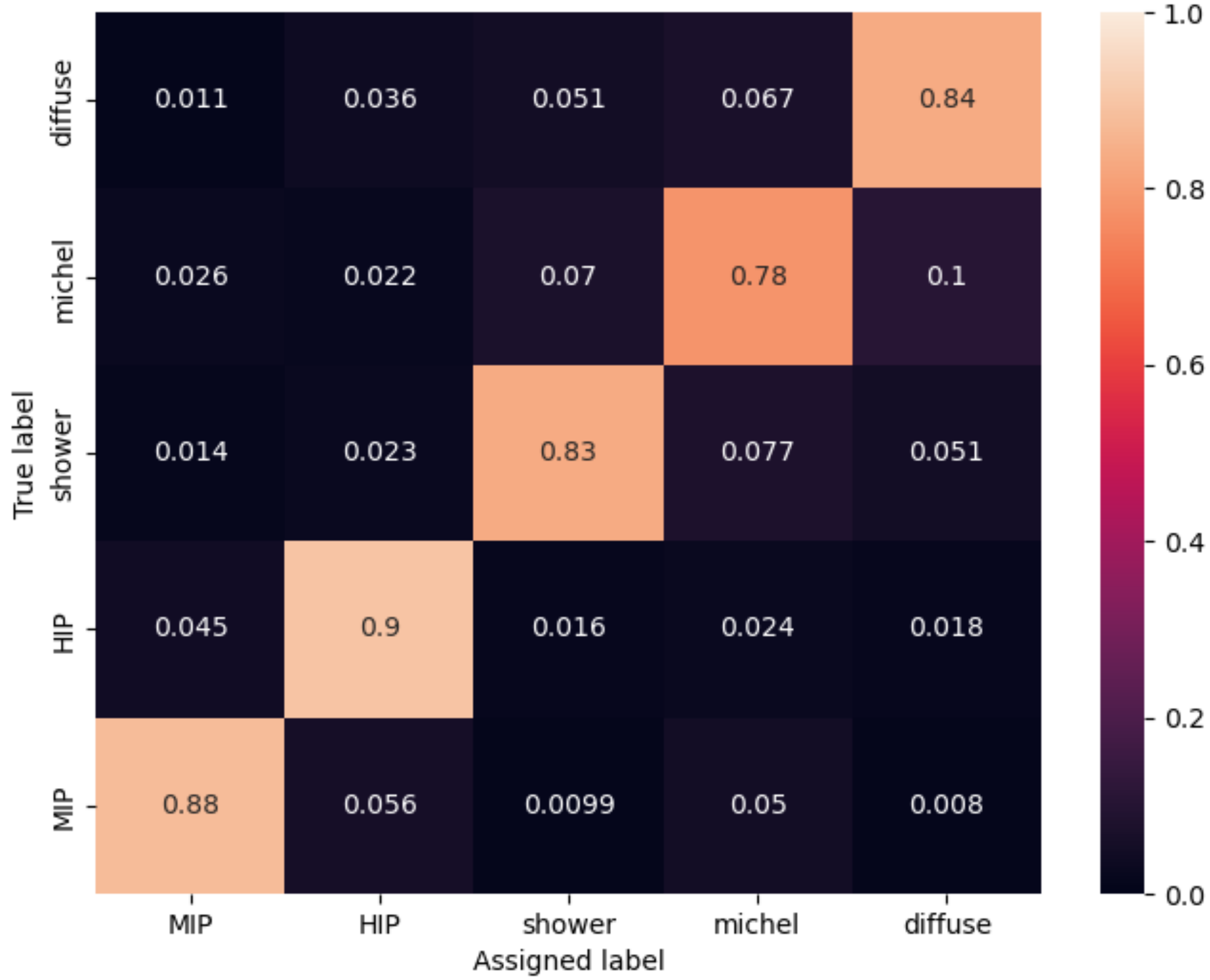
- Add additional 3D step to the standard message-passing loop.



NuGraph2

- Network achieves **~86%** overall hit classification accuracy.
- With 3D connections, consistency of representations between views is now around **98%**, compared to ~70% without.

Confusion matrix weighted by
true semantic label.
to show **efficiency**.

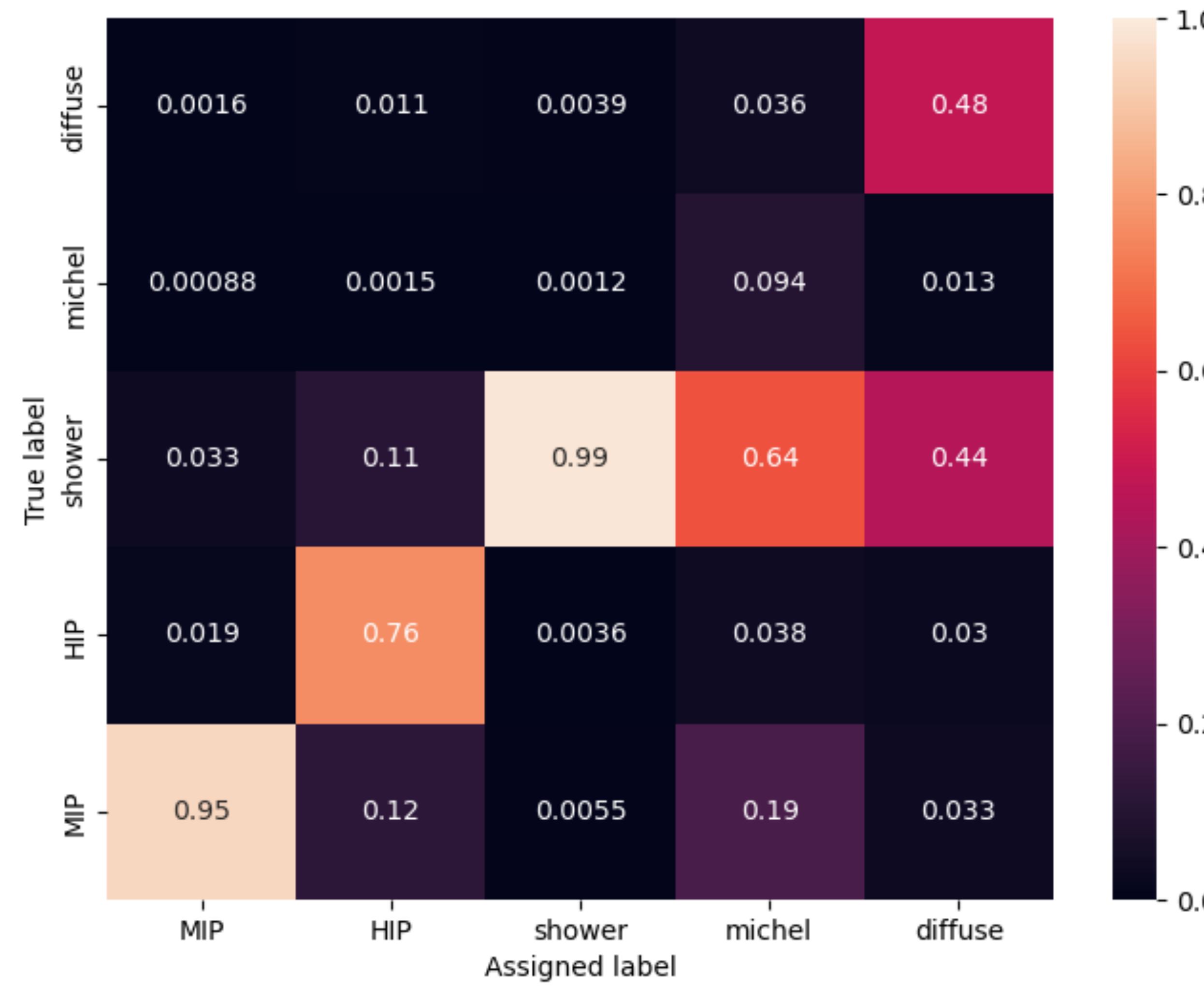


University of CINCINNATI

NuGraph2

- Network achieves **~86%** overall hit classification accuracy.
- With 3D connections, consistency of representations between views is now around **98%**, compared to ~70% without.

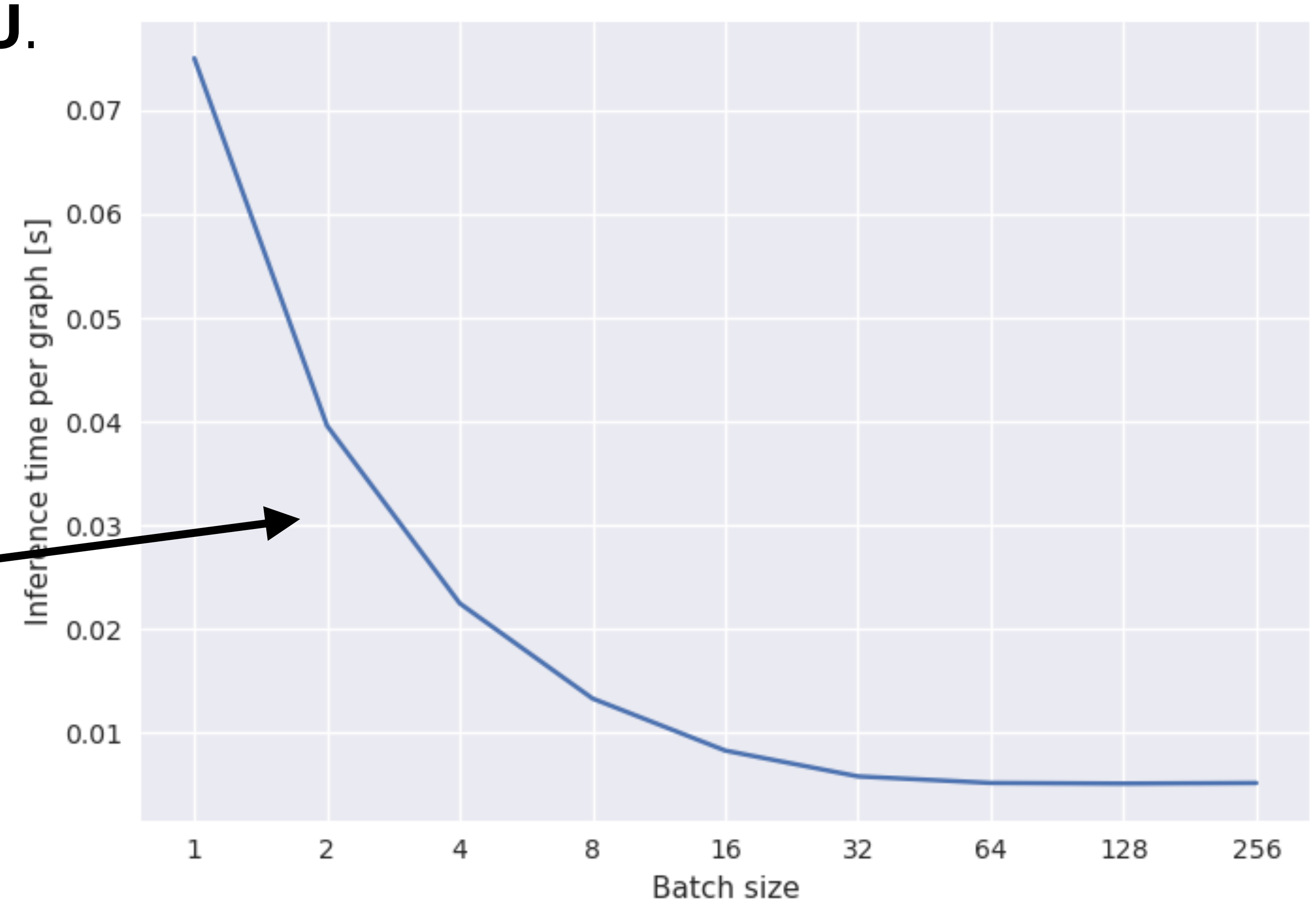
Confusion matrix weighted by **predicted semantic label**. to show **purity**.



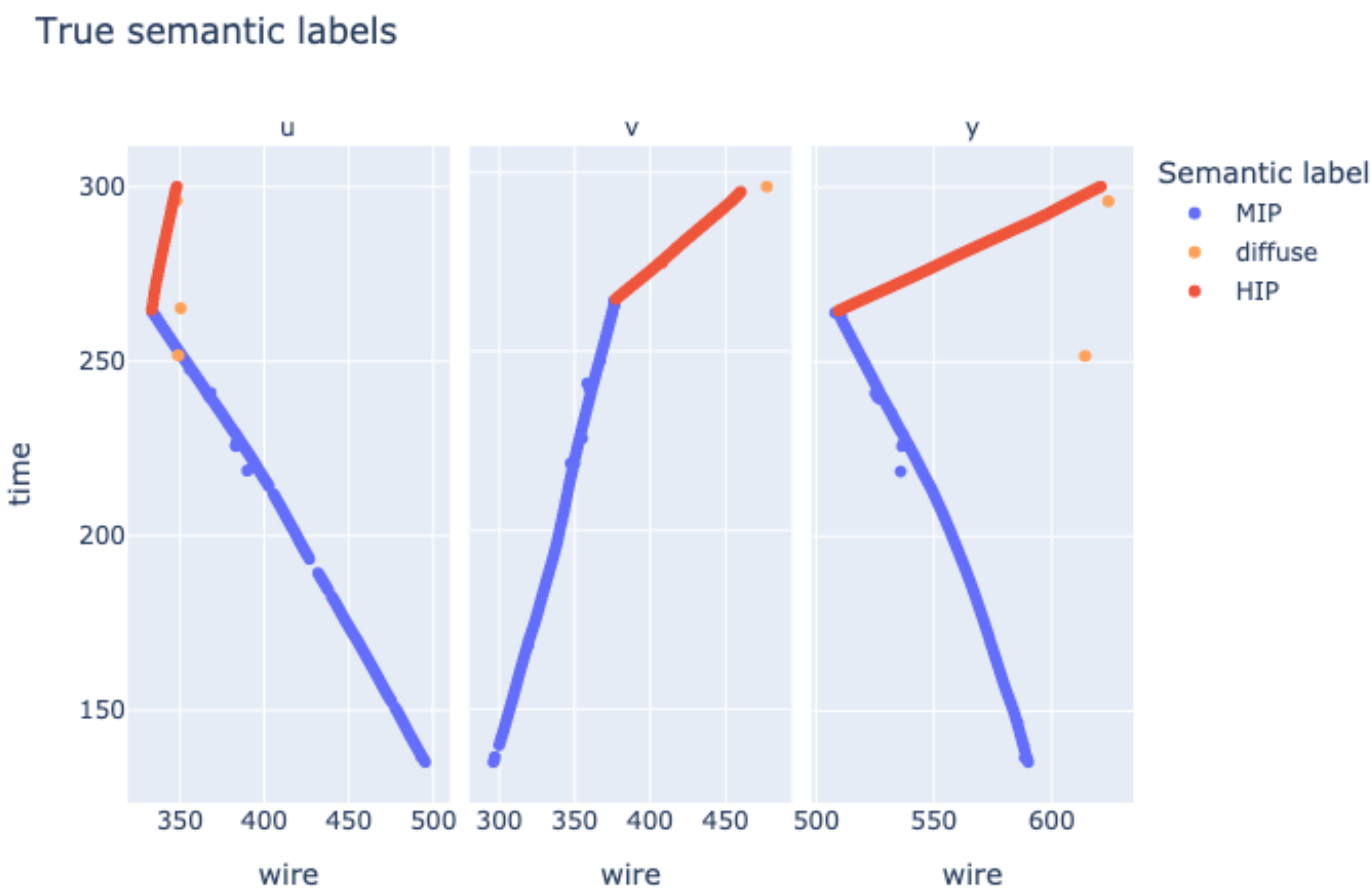
NuGraph2

- Network achieves **~86%** overall hit classification accuracy.
- With 3D connections, consistency of representations between views is now around **98%**, compared to ~70% without.
- Inference takes **0.12 s / event on CPU**.

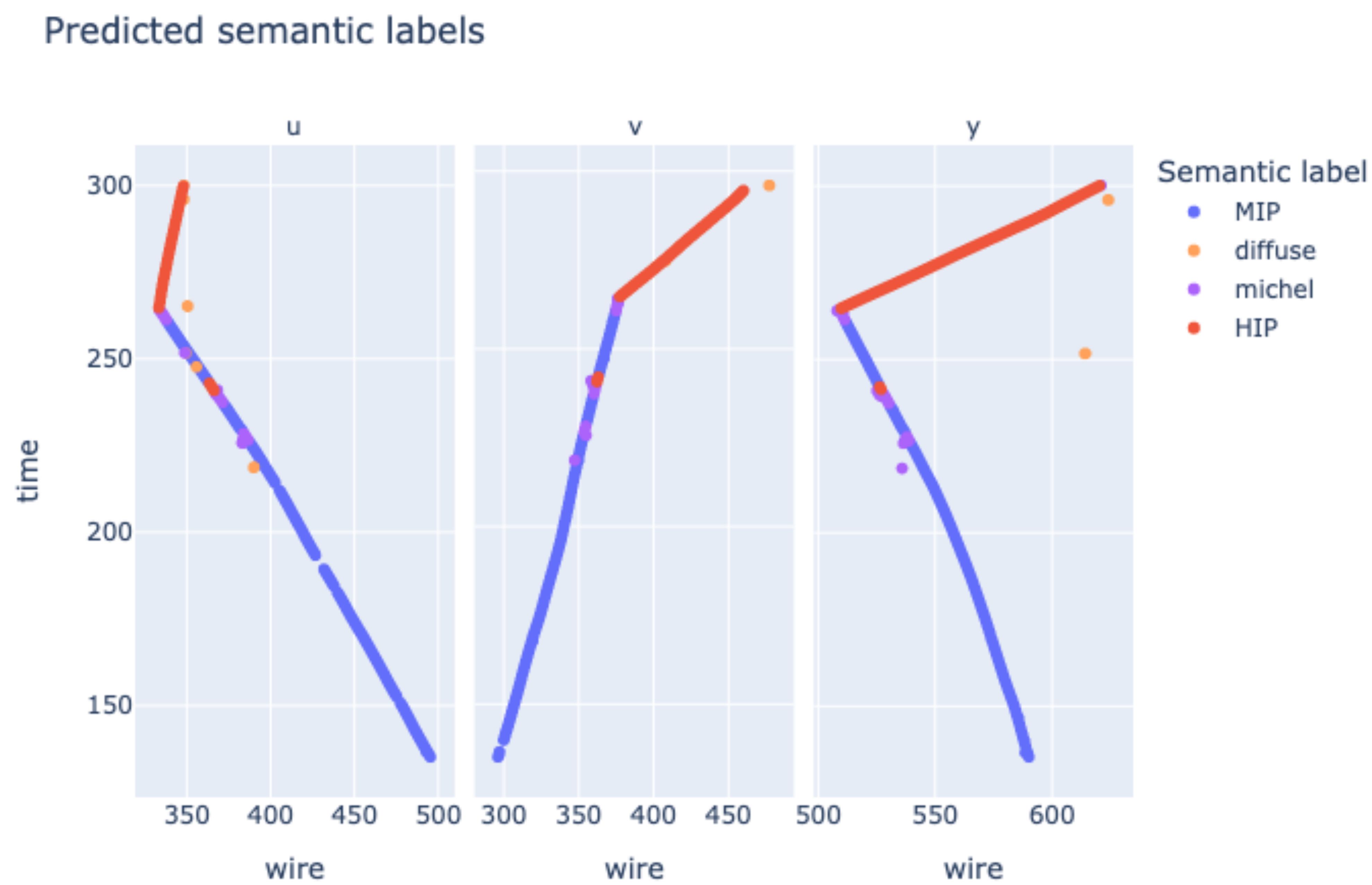
GPU inference time as a function of batch size



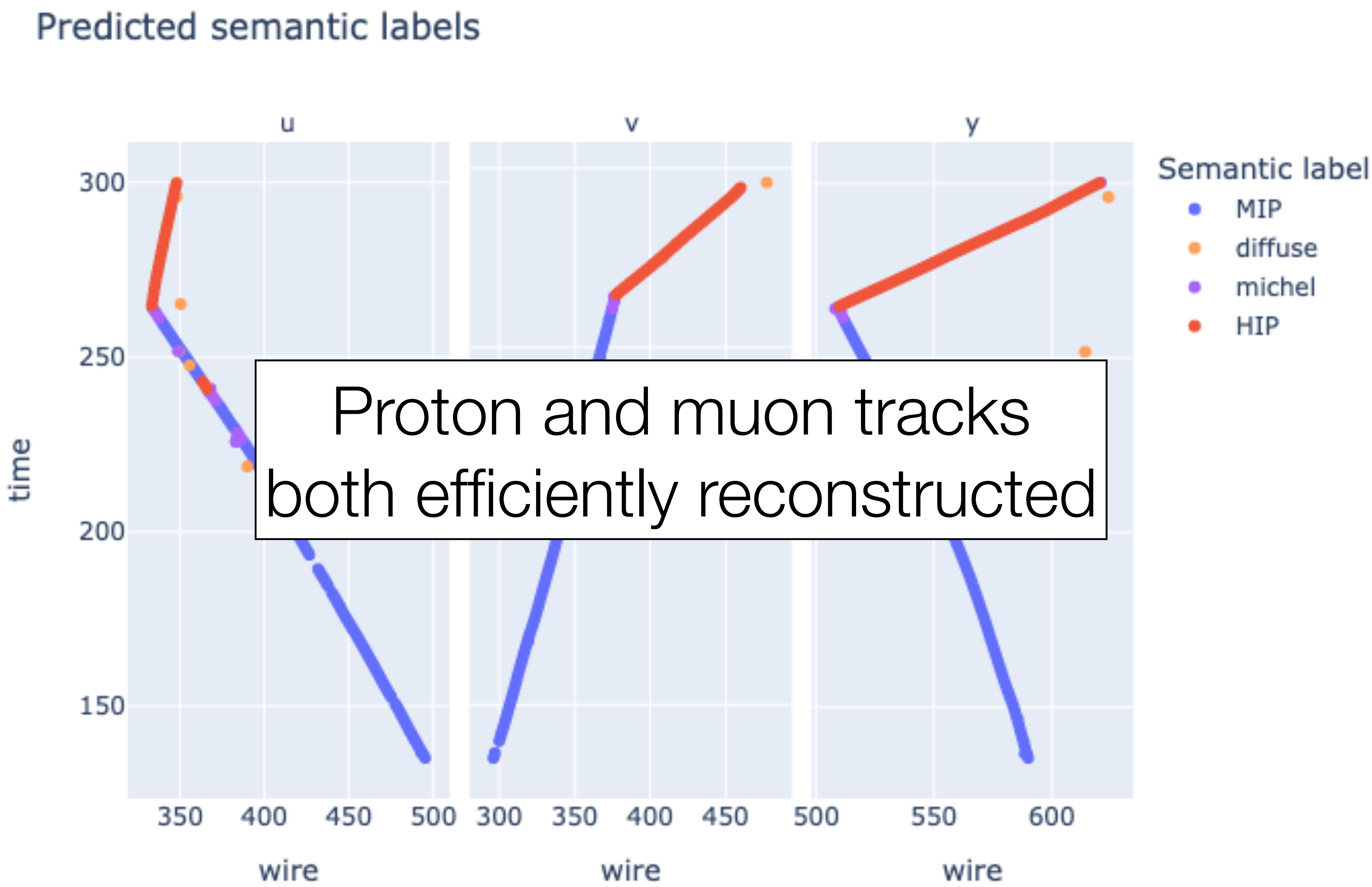
Example ν_μ interaction



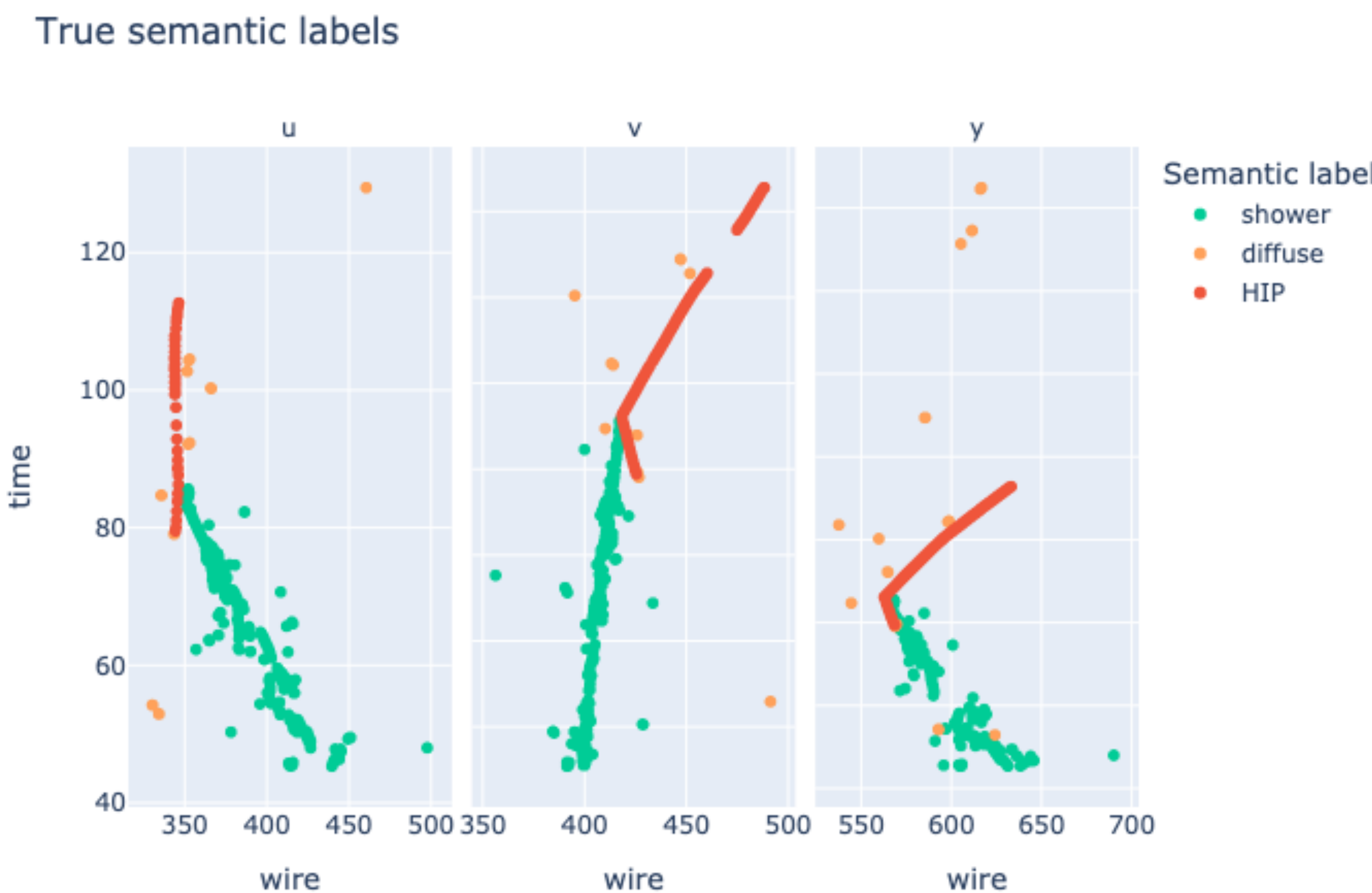
Example ν_μ interaction



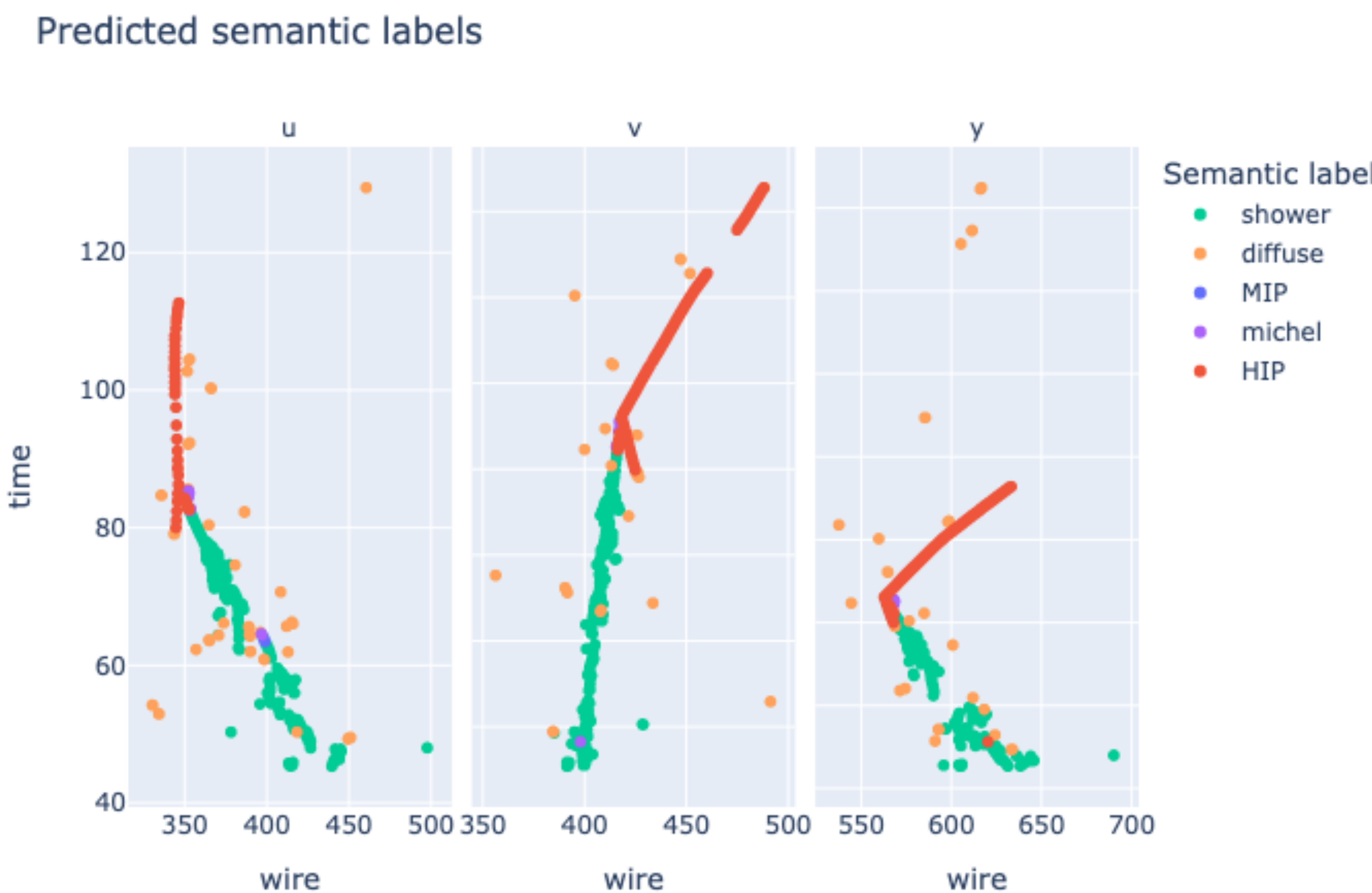
Example ν_μ interaction



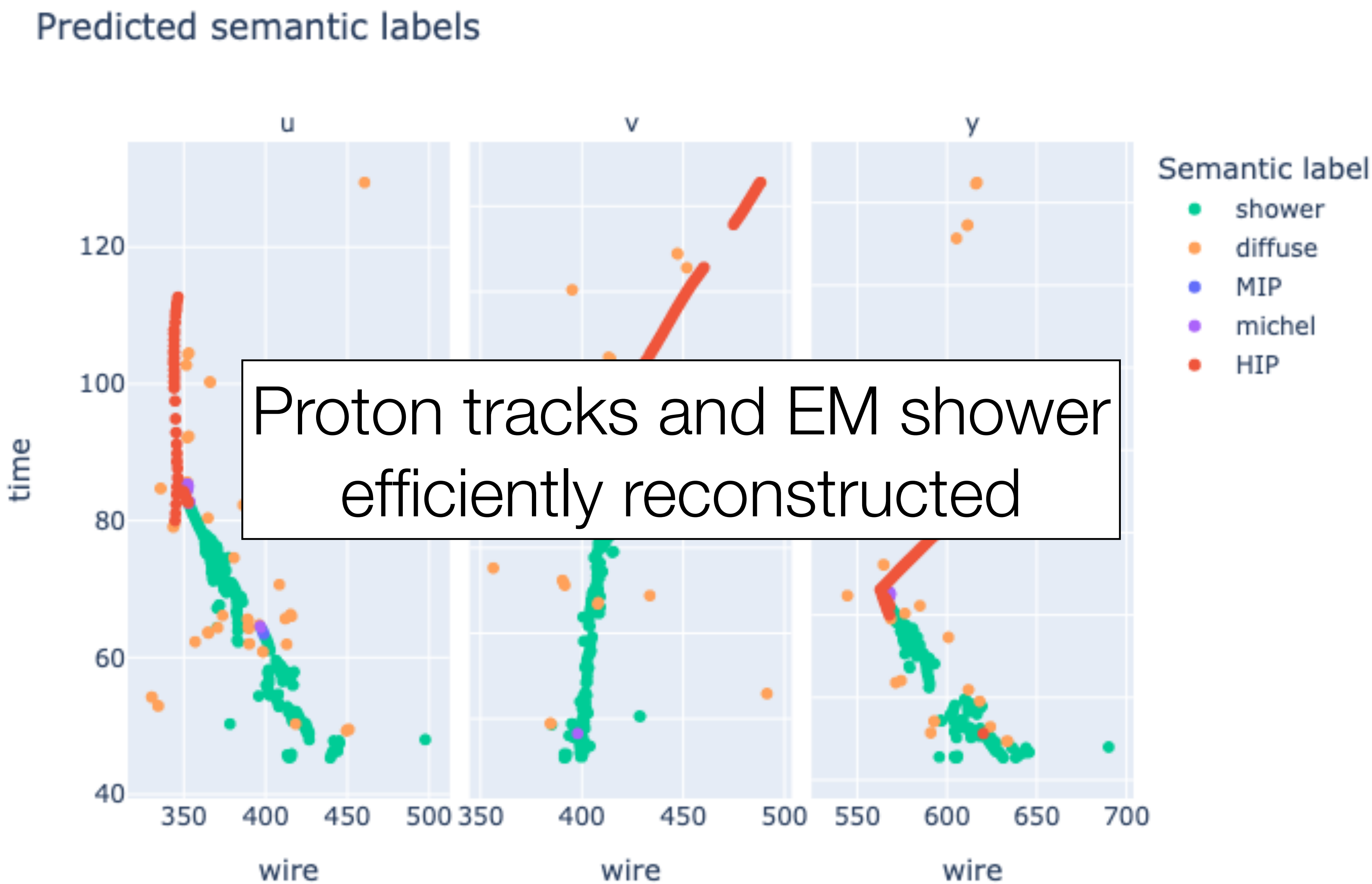
Example ν_e interaction



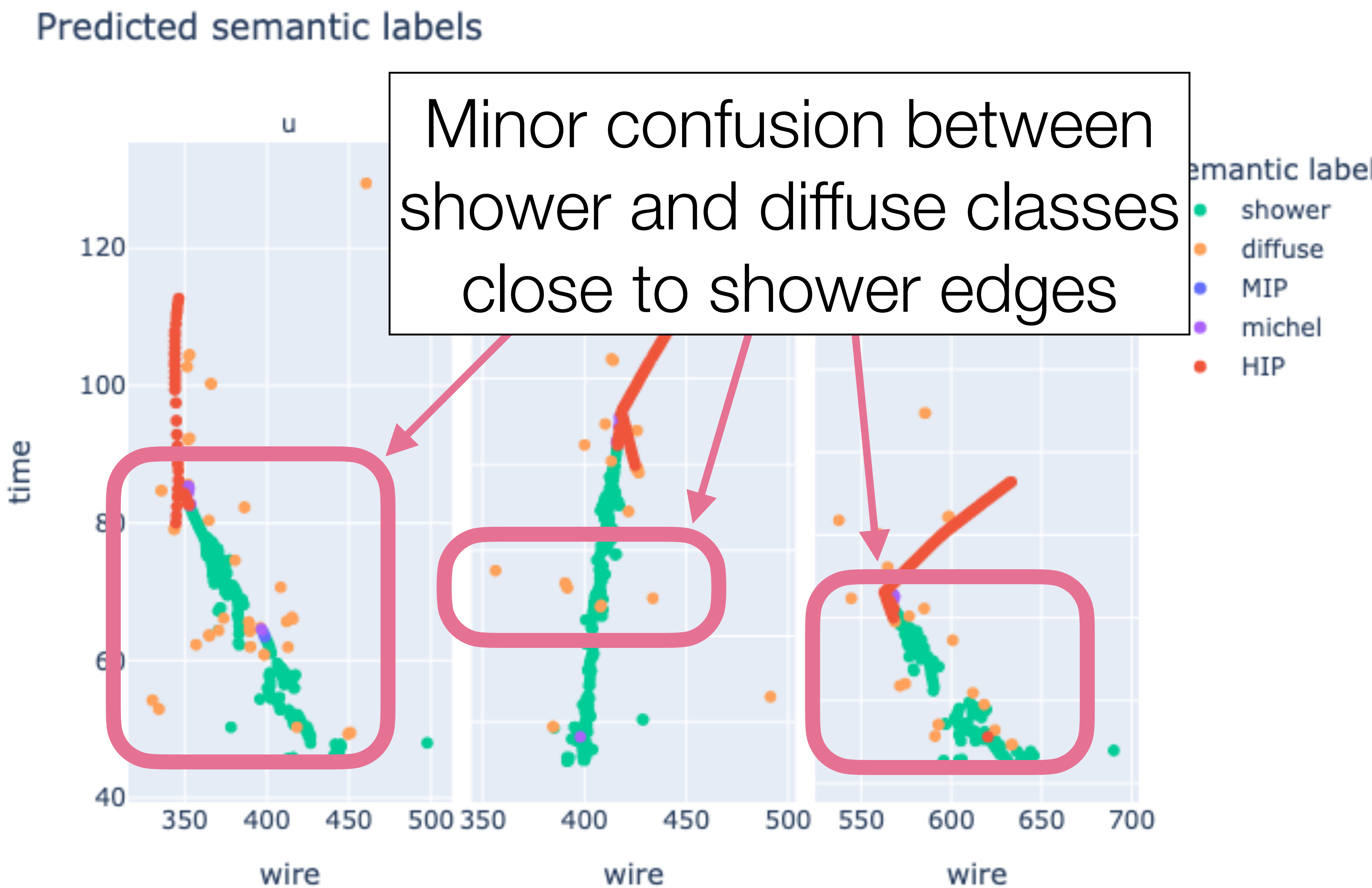
Example ν_e interaction



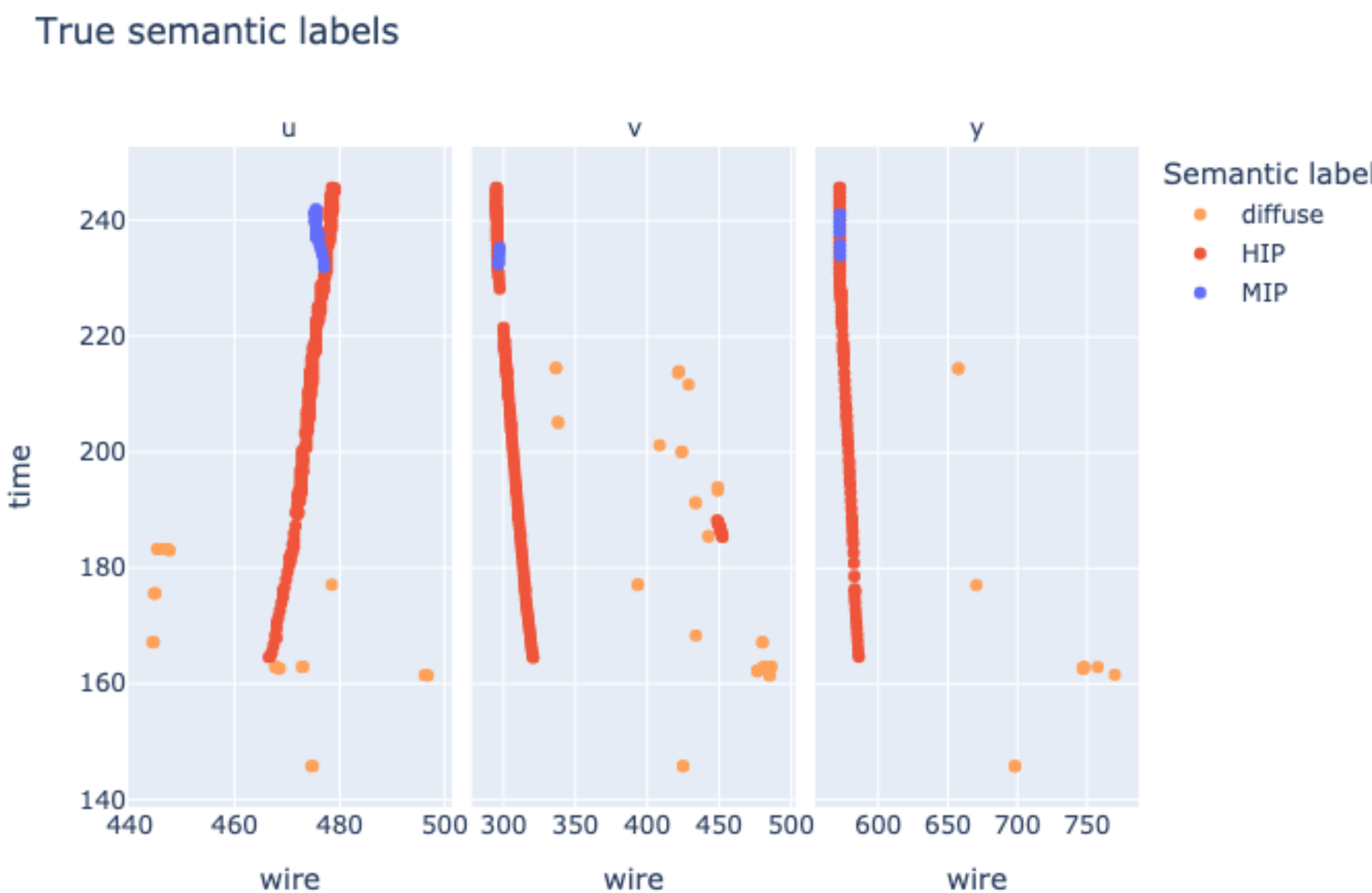
Example ν_e interaction



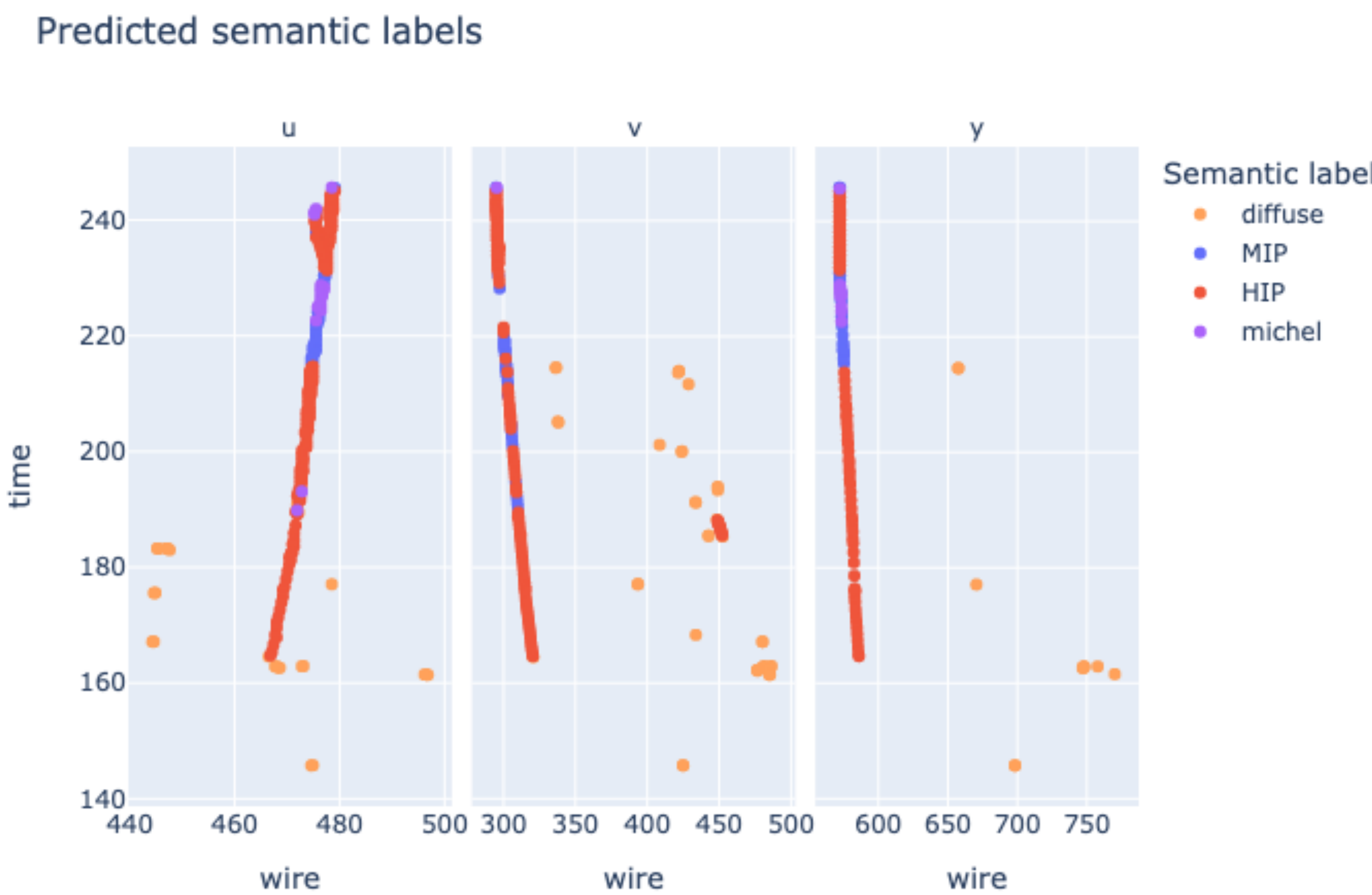
Example ν_e interaction



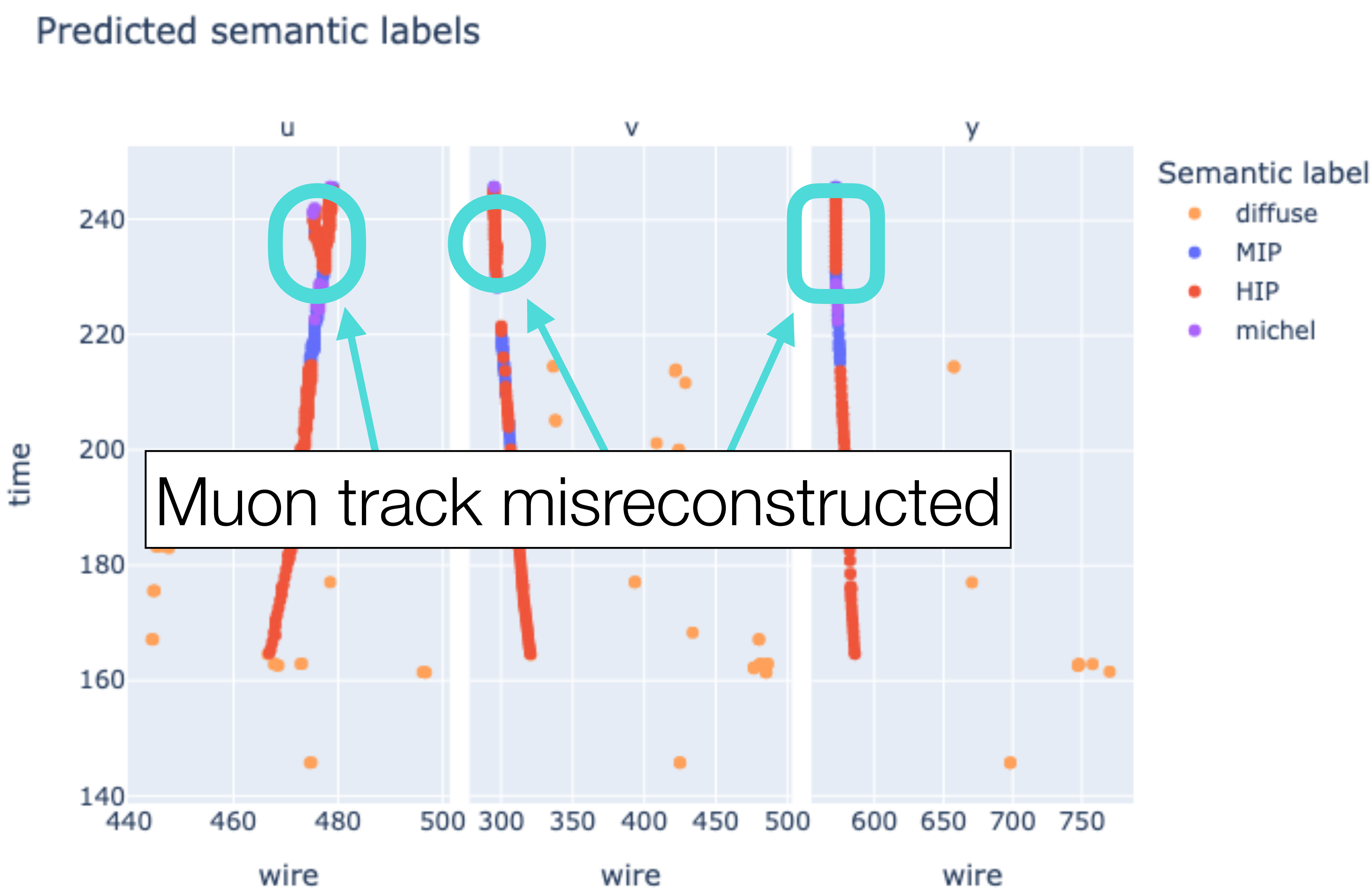
Example ν_μ interaction



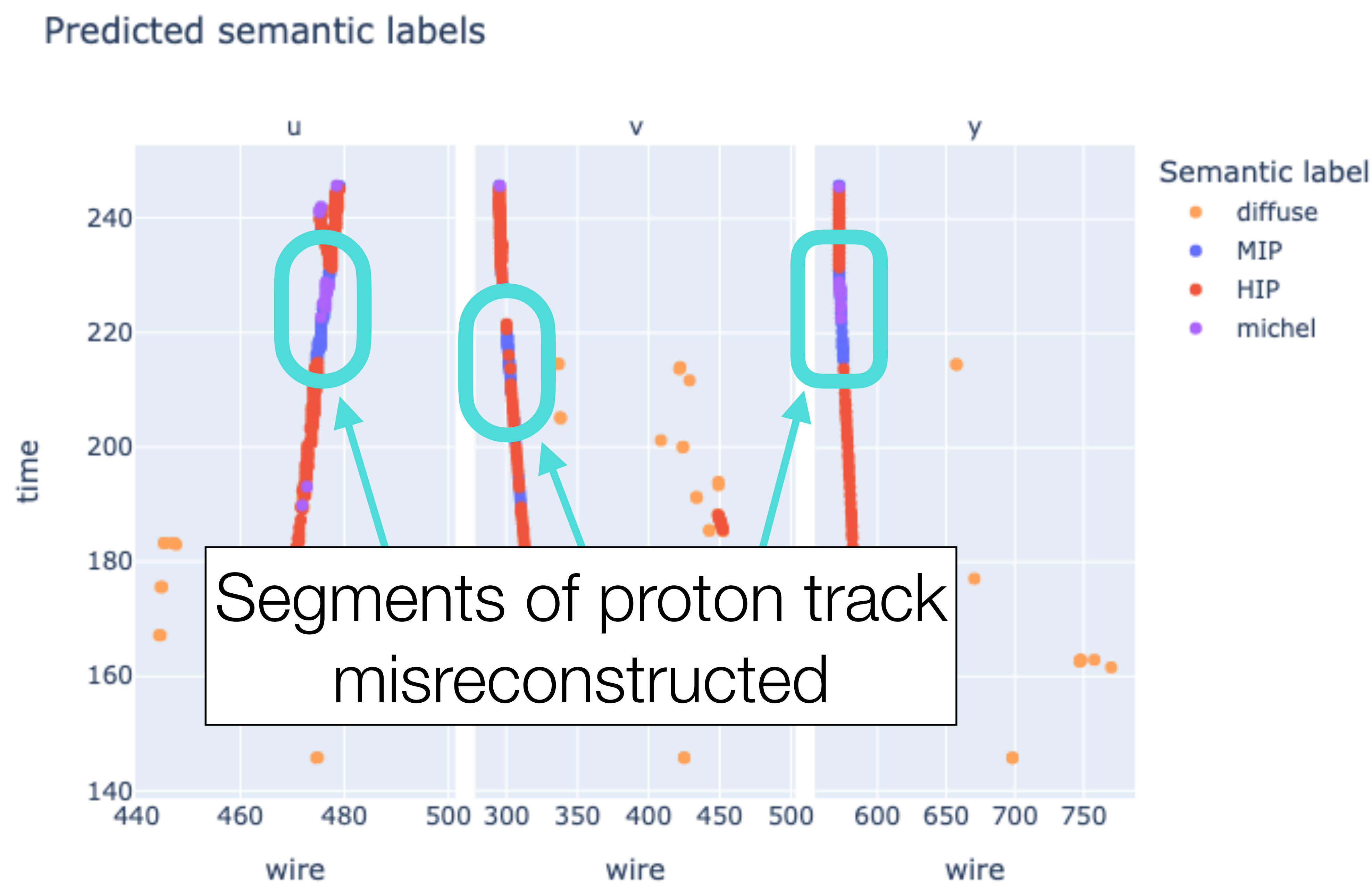
Example ν_μ interaction



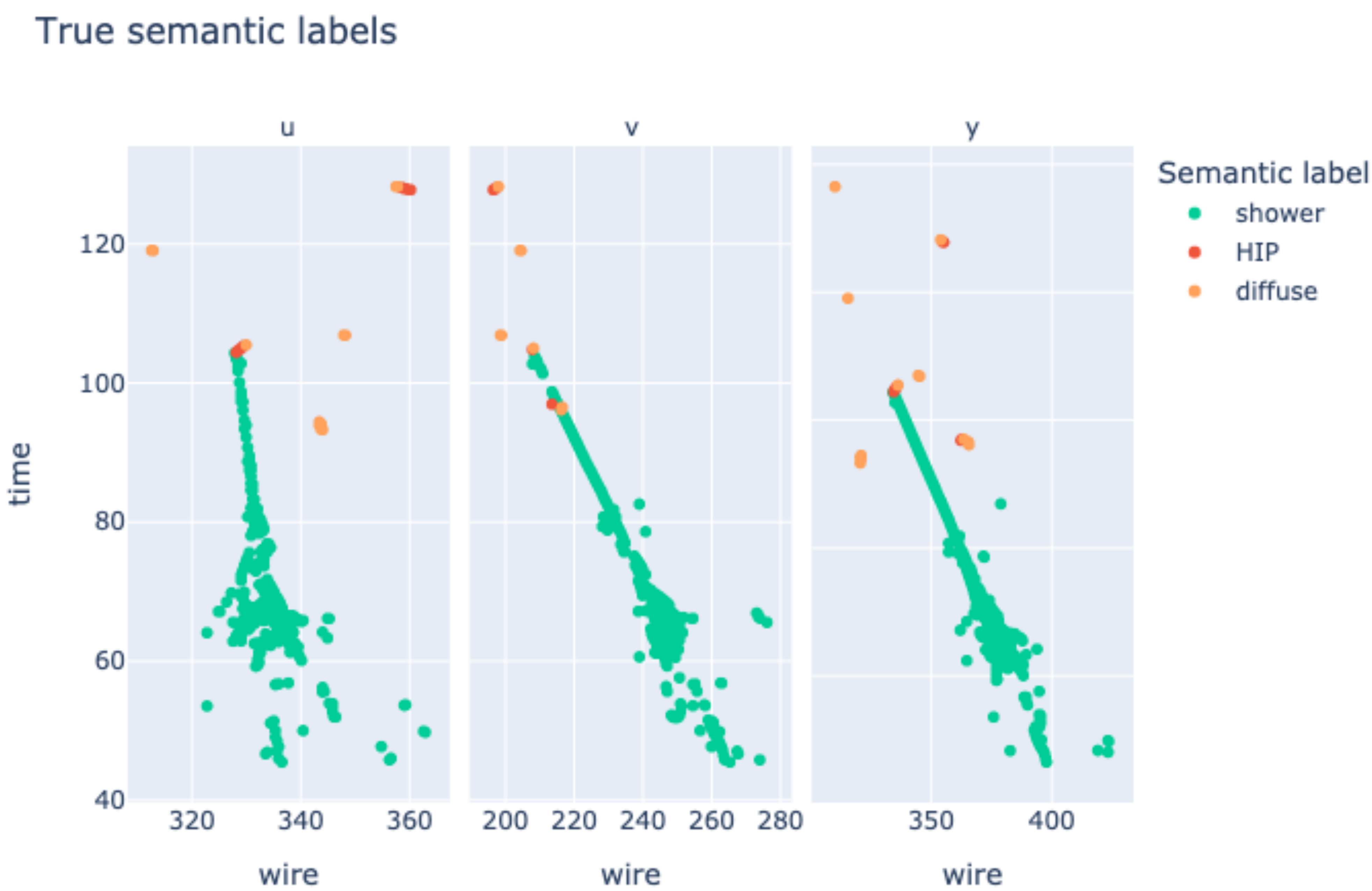
Example ν_μ interaction



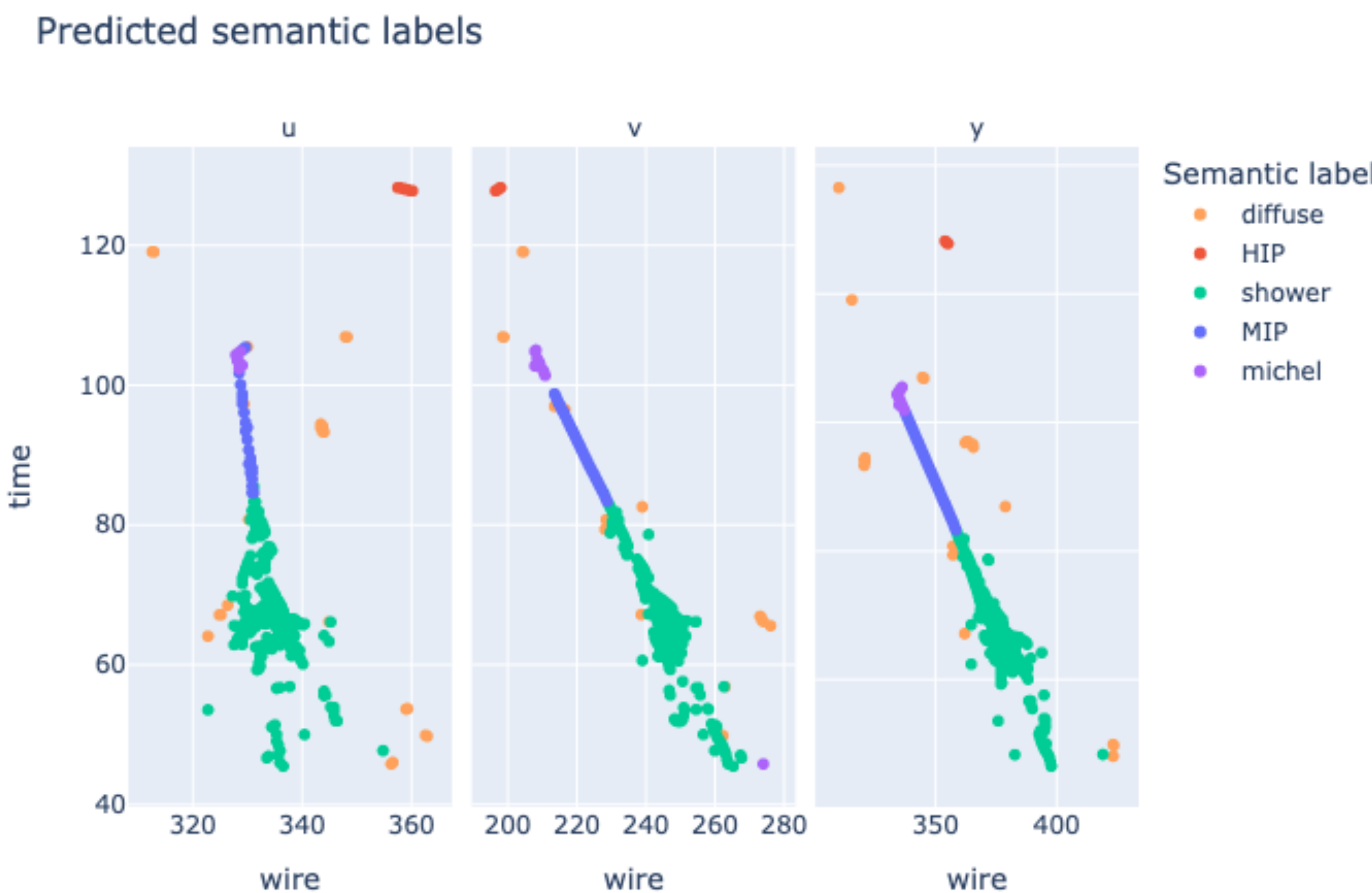
Example ν_μ interaction



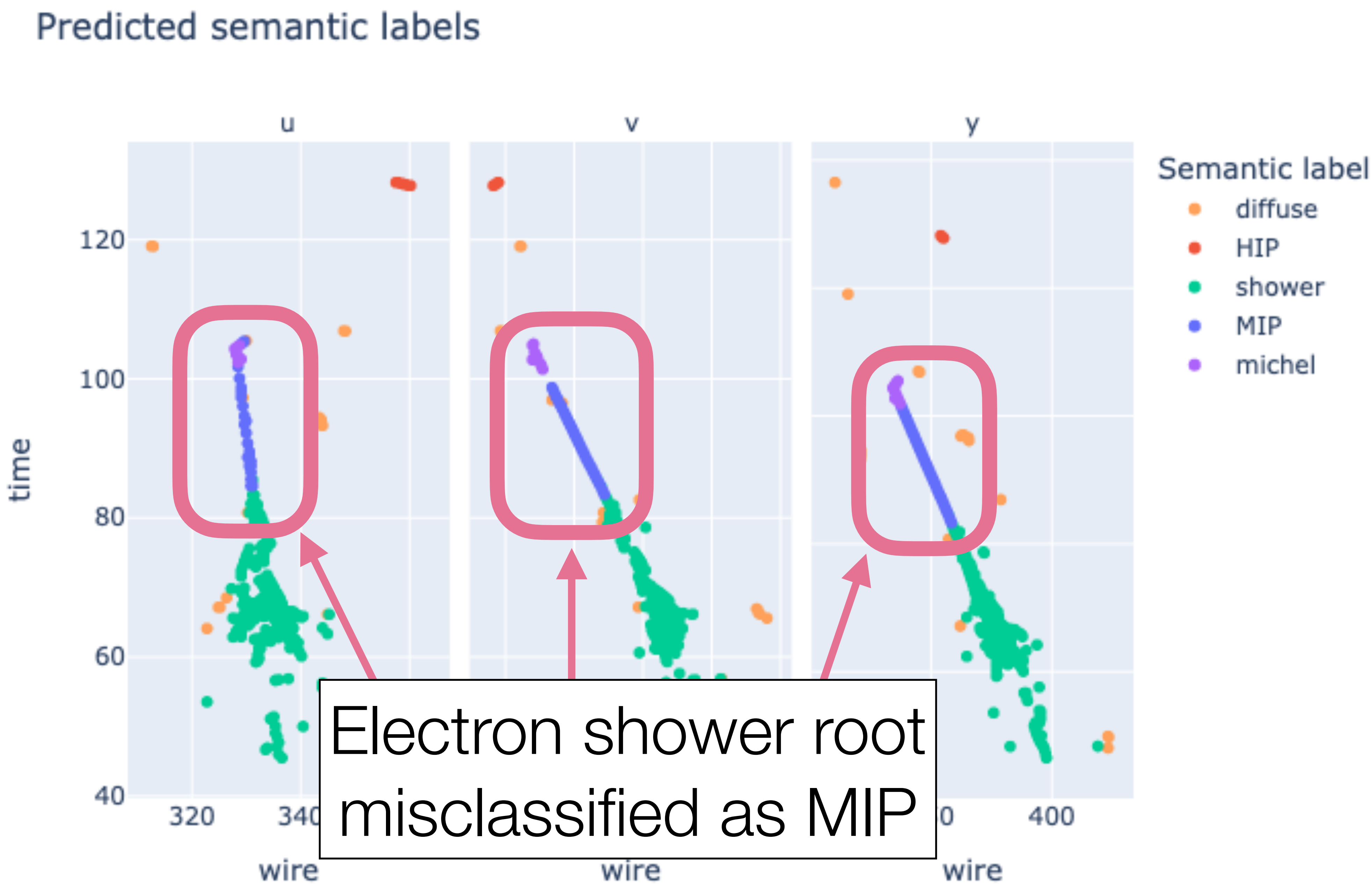
Example ν_e interaction



Example ν_e interaction

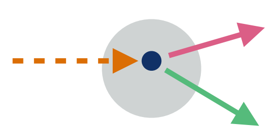
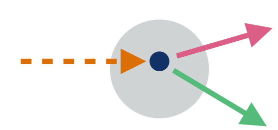
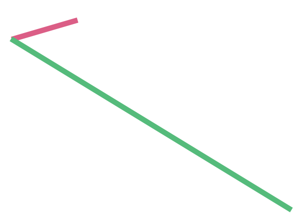
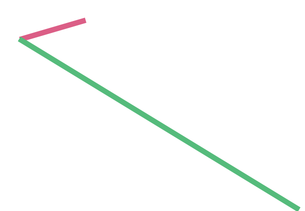
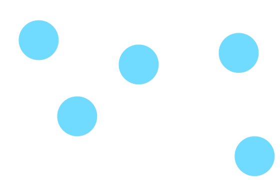
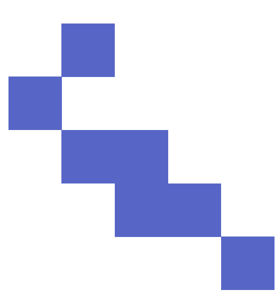
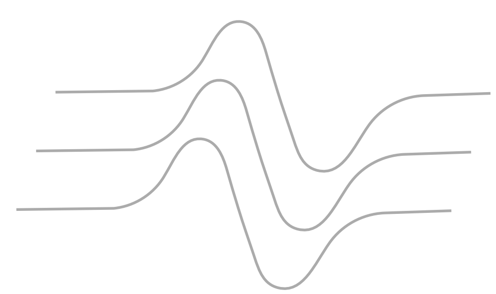


Example ν_e interaction



Common abstraction for neutrino experiments

- Although the details of many neutrino physics experiments vary, the majority of them share a common paradigm at a high level.

NOvA		MicroBooNE		Shared structure
	Neutrino generator (GENIE)		Neutrino generator (GENIE)	**Event information**
	Particle simulation (Geant4)		Particle simulation (Geant4)	**True particles**
	True light depositions		True ionization electrons	**True energy deposits**
	Photoelectrons on APDs		Pulses on TPC wires	**Detector hits**

NuML & PyNuML

- The **NuML** package is a toolkit for writing **physics event records** to an **HDF5 file format**.
  - Hold low-level information such as **simulated particles**, **hits**, **true energy depositions** etc.
  - Generic data structure can be **shared across experiments**.
  - Common interface with **PandAna** analysis toolkit (see CHEP 2021 talk).
  - Available as LArSoft package on GitHub.

NuML & PyNuML

- The **NuML** package is a toolkit for writing **physics event records** to an **HDF5 file format**.
  - Hold low-level information such as **simulated particles**, **hits**, **true energy depositions** etc.
  - Generic data structure can be **shared across experiments**.
  - Common interface with **PandAna** analysis toolkit (see CHEP 2021 talk).
  - Available as LArSoft package on GitHub.
- The **PyNuML** package is designed to provide a **generic**, **accessible**, **efficient** and **flexible** solution for many of the necessary tasks in leveraging ML for particle physics.
  - Define **particle ground truth labels** for Geant4-simulated particles.
  - **Arrange detector hits into ML objects**, ie. graphs, CNN pixel maps, etc.
  - Efficiently **preprocess ML inputs in parallel in HPC environments** using MPI.
  - Available as Python package on GitHub, or install with `pip install pynuml`!

Summary

- **NuGraph2** is a state-of-the-art graph neural network for semantically labelling detector hits in neutrino physics experiments.
  - Model developed and tested in MicroBooNE and DUNE, and designed to be utilised across many neutrino physics detectors.
  - Targeting full particle reconstruction for next generation architecture.
- Standardised process of producing ML inputs from HEP data for general use with **NeutrinoML** toolkit.
  - Toolkit utilised for MicroBooNE's public data release.
  - Open-source, easy-to-install code packages.
- Next step: train up-to-date architecture on DUNE simulation, and close the inference loop by incorporating output into Art record.

Backup

Producing graphs for model training

PyNuML

Art record
(hits, simulated particles, true energy depositions)

NuGraph2

Producing graphs for model training

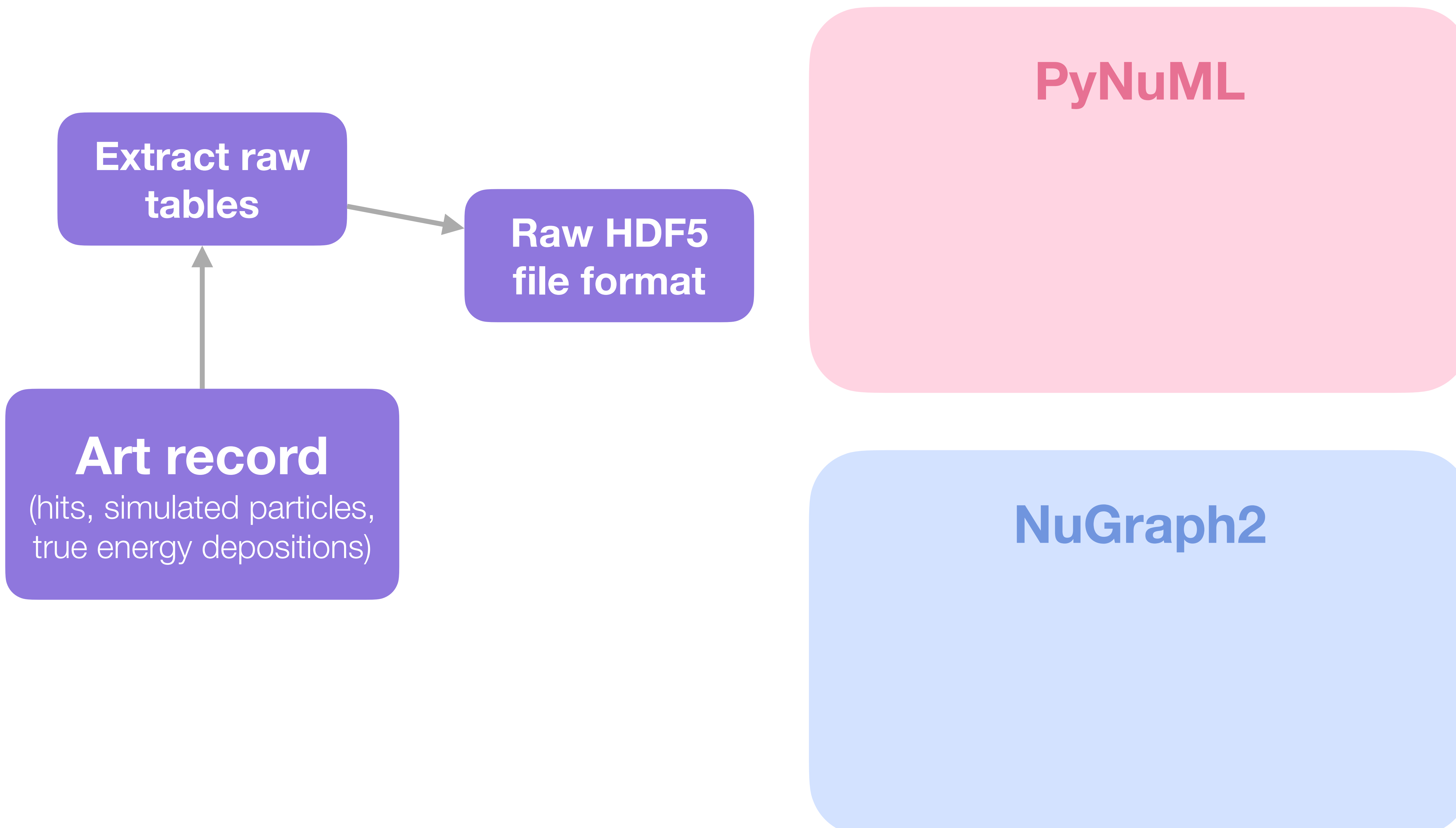
Extract raw tables

Art record
(hits, simulated particles, true energy depositions)

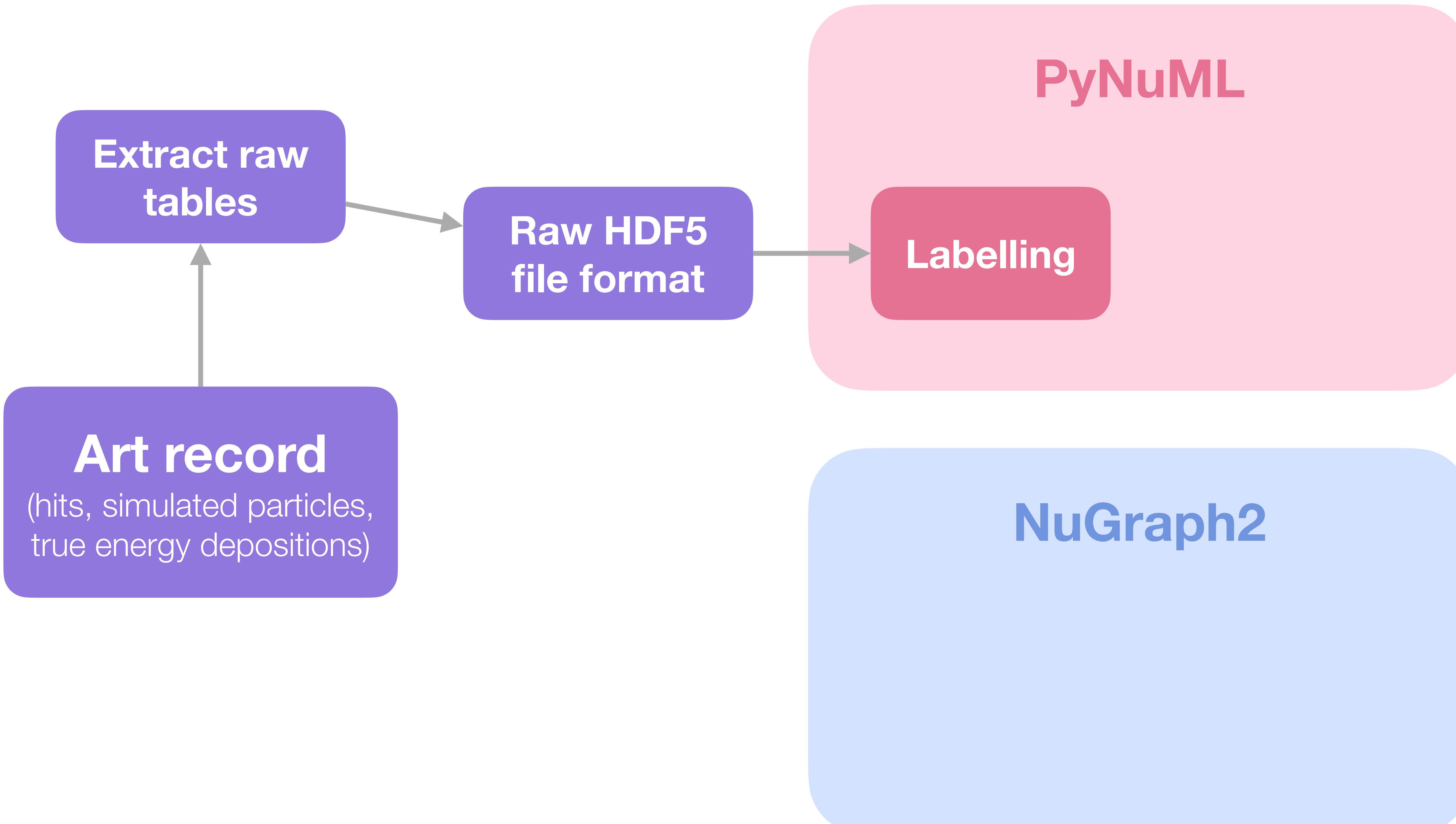
PyNuML

NuGraph2

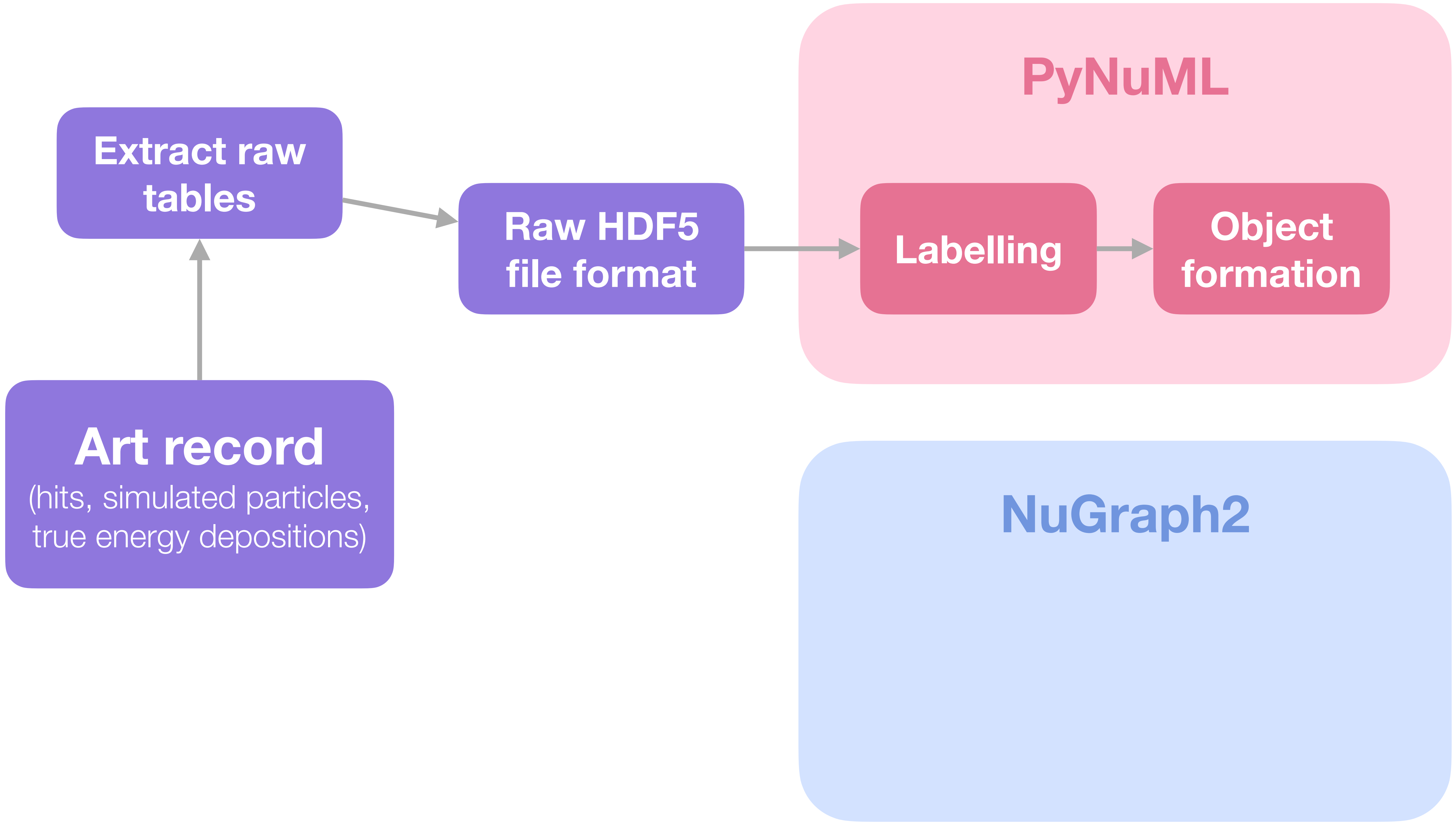
Producing graphs for model training



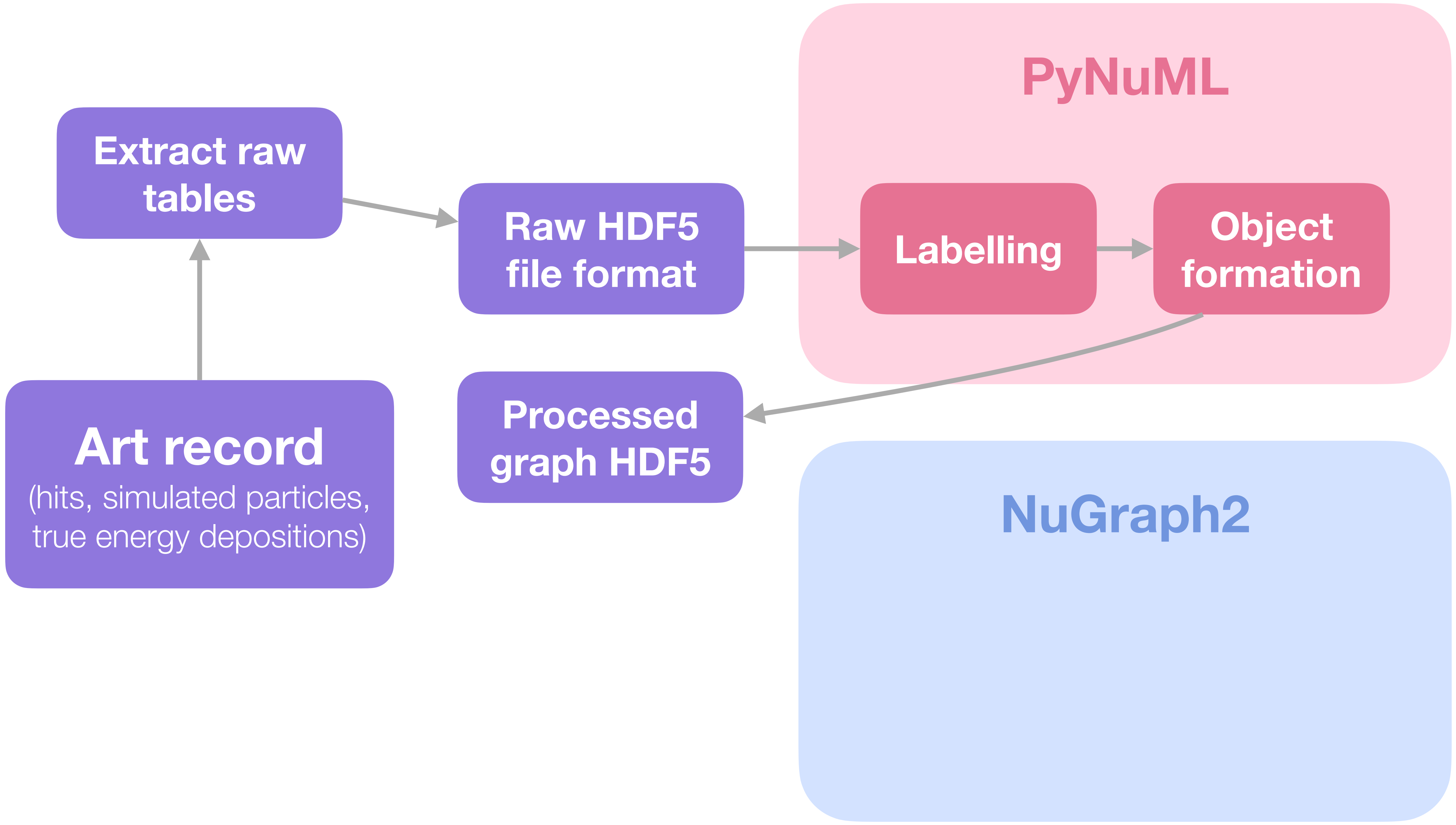
Producing graphs for model training



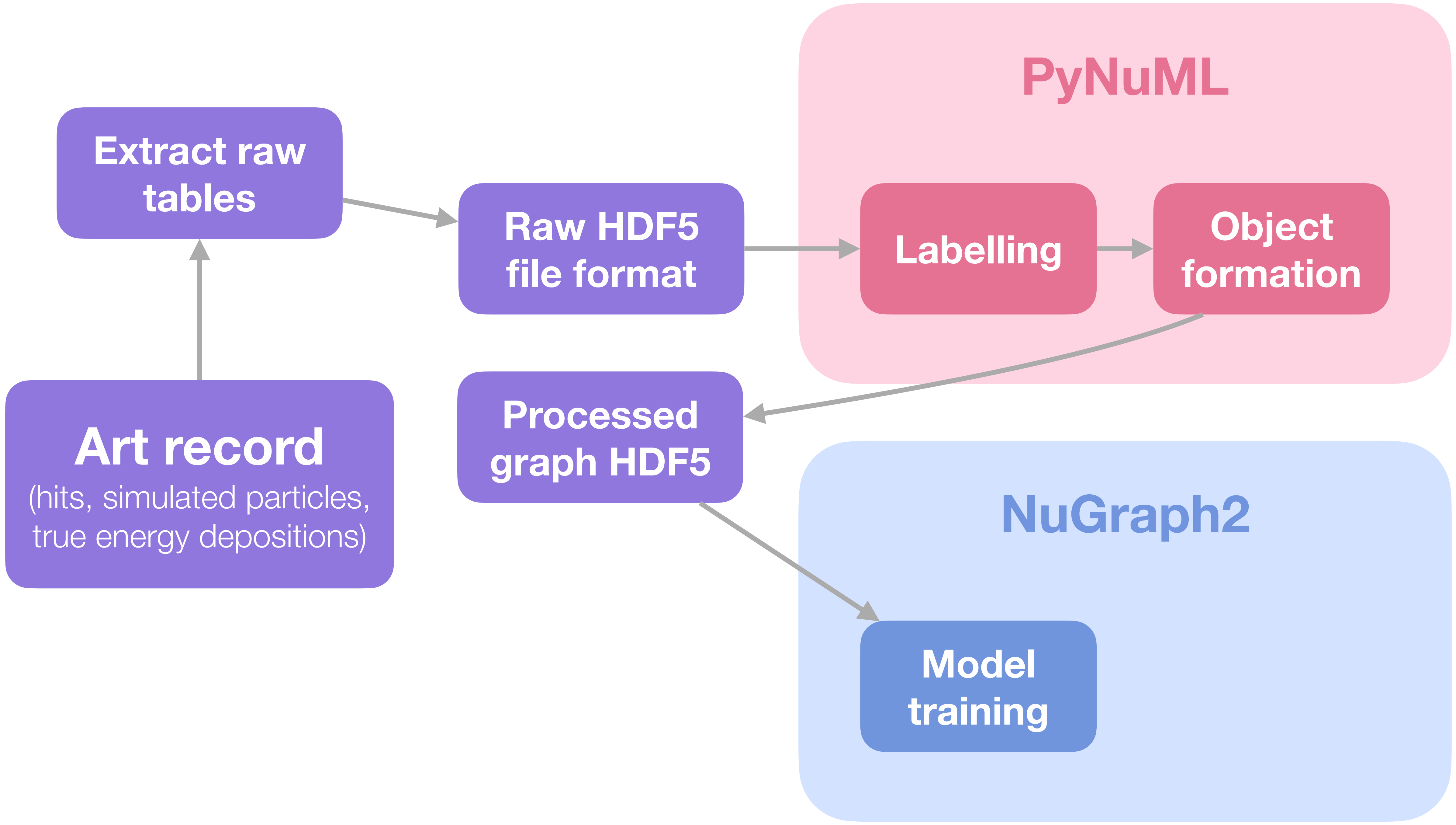
Producing graphs for model training



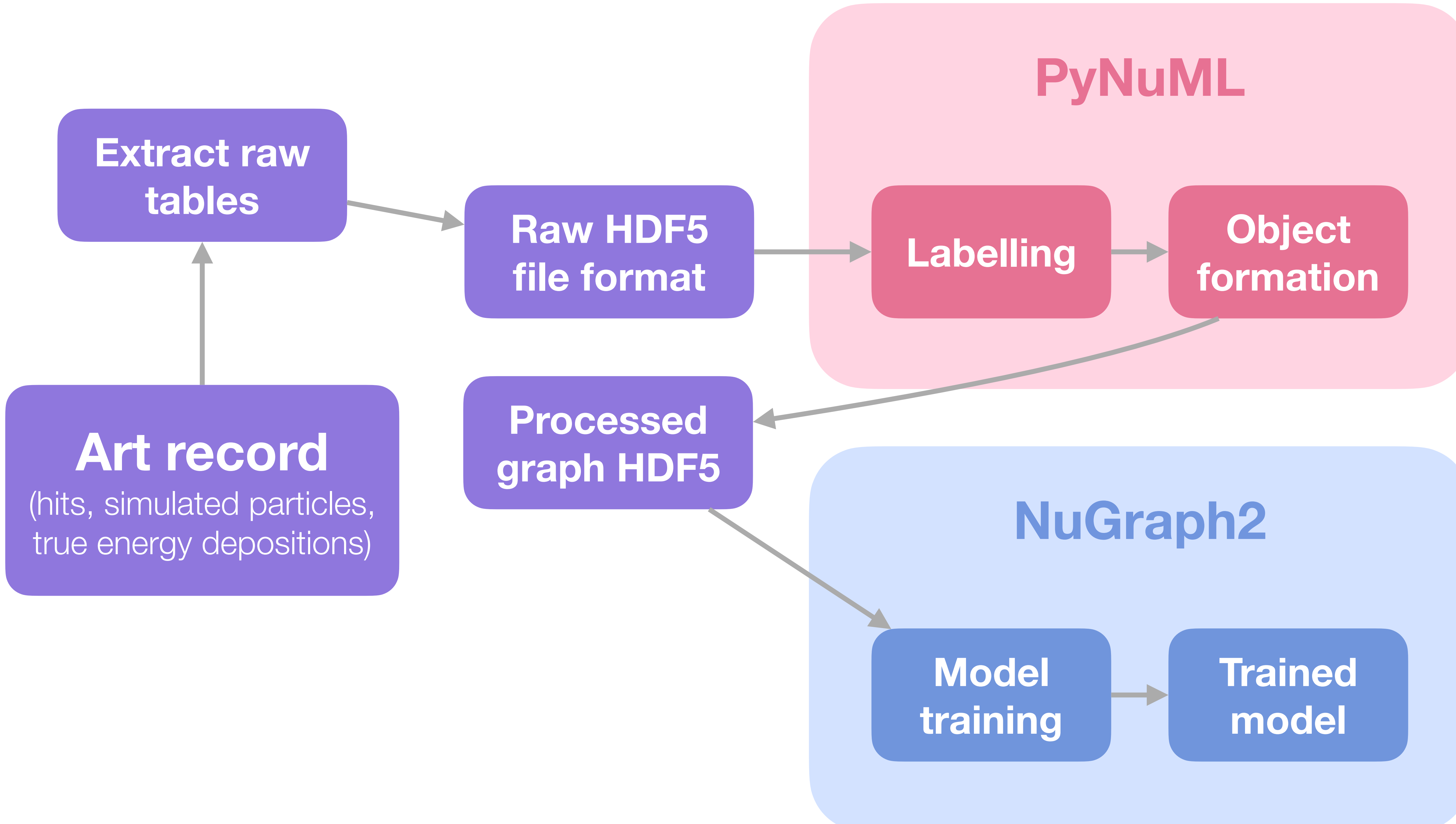
Producing graphs for model training



Producing graphs for model training



Producing graphs for model training



Inference in production

PyNuML

Object formation

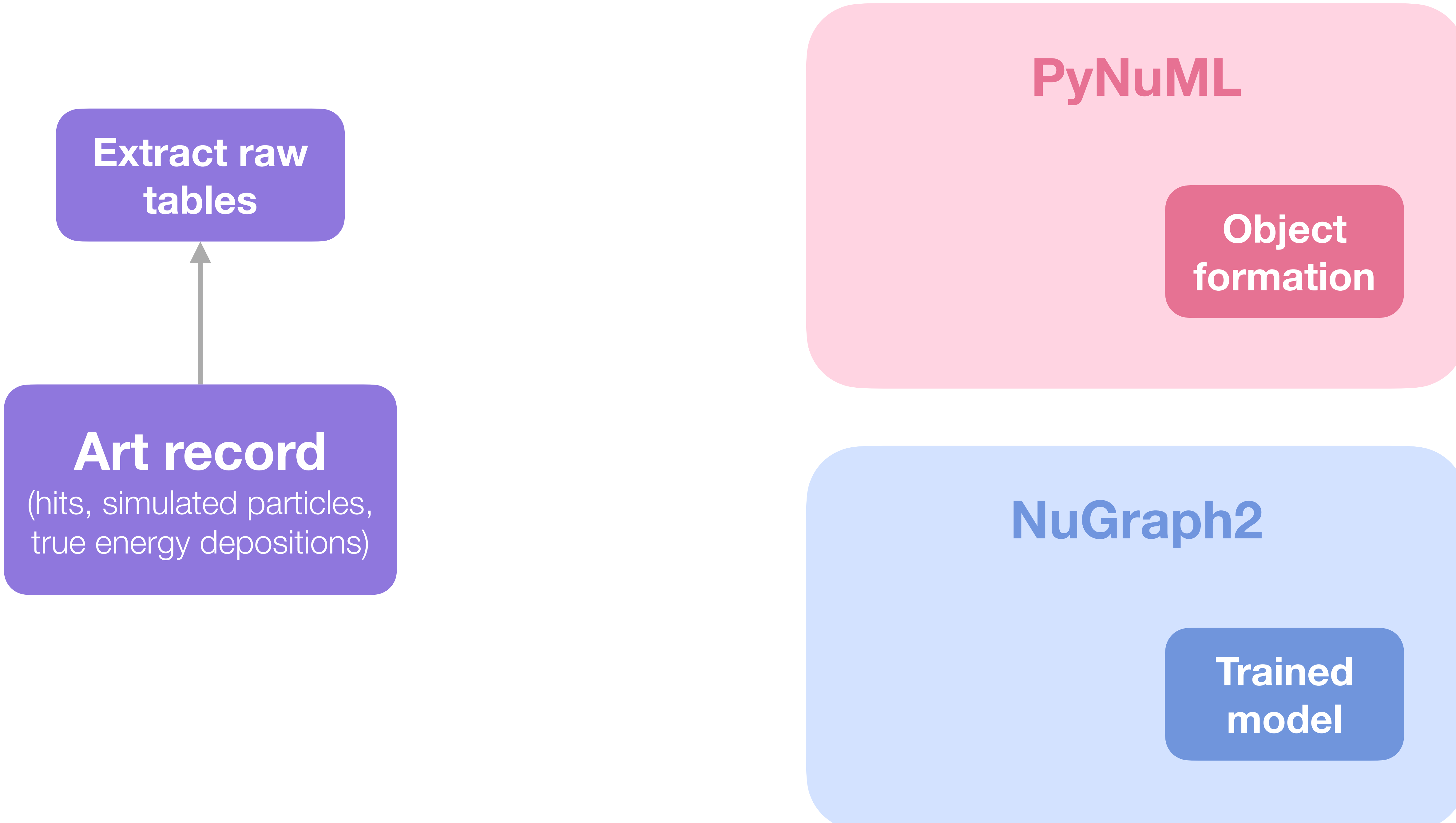
Art record

(hits, simulated particles, true energy depositions)

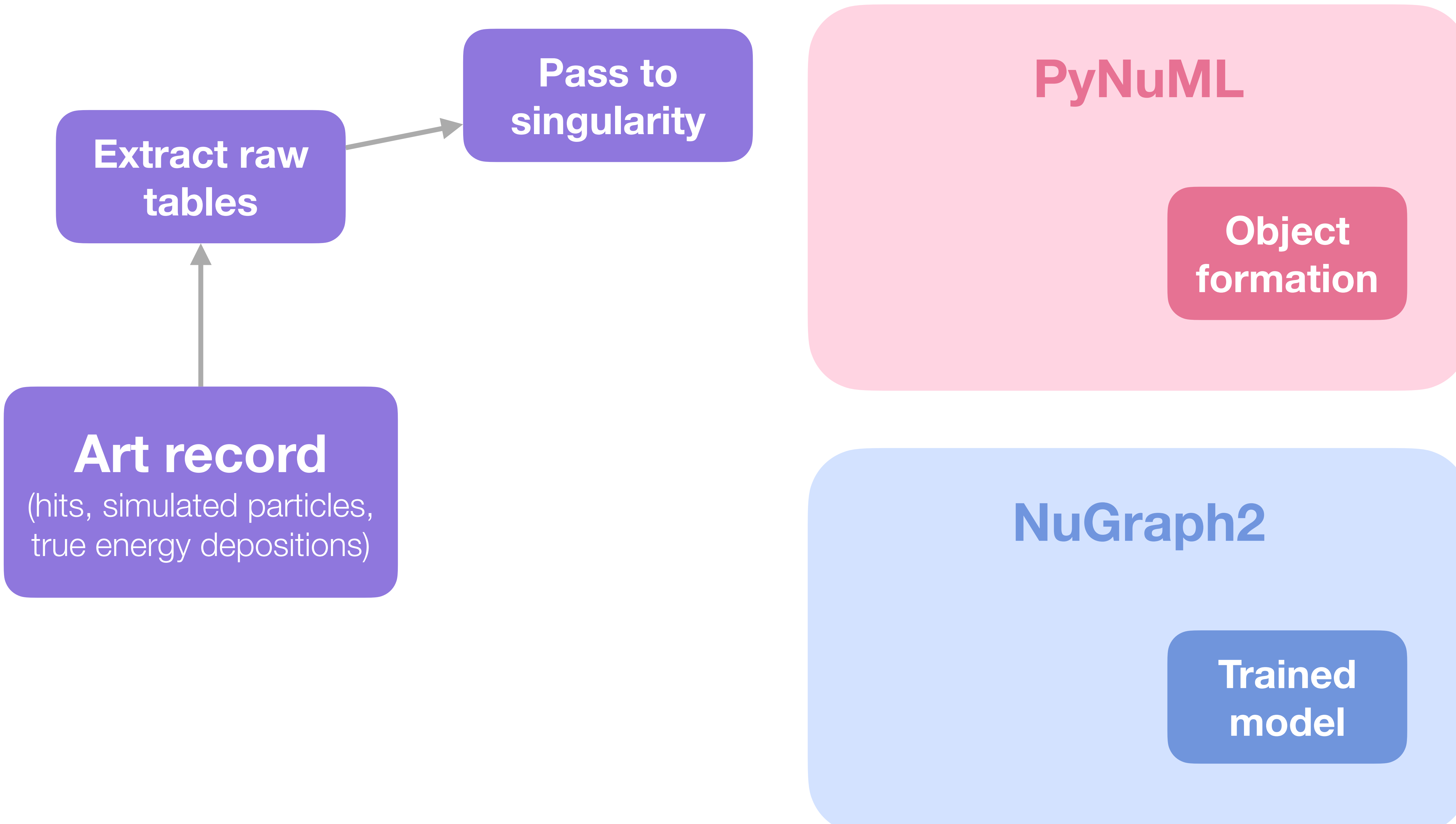
NuGraph2

Trained model

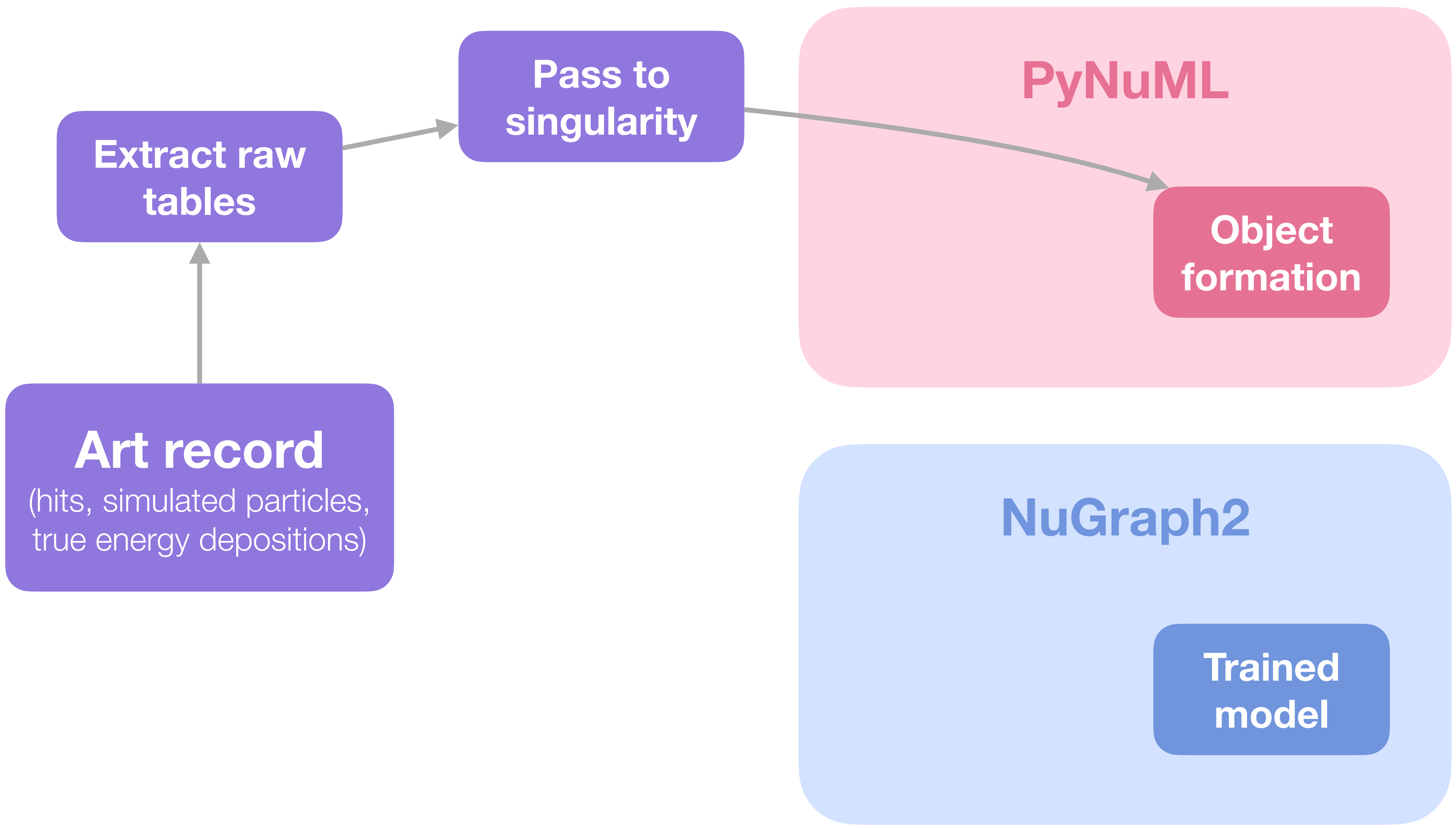
Inference in production



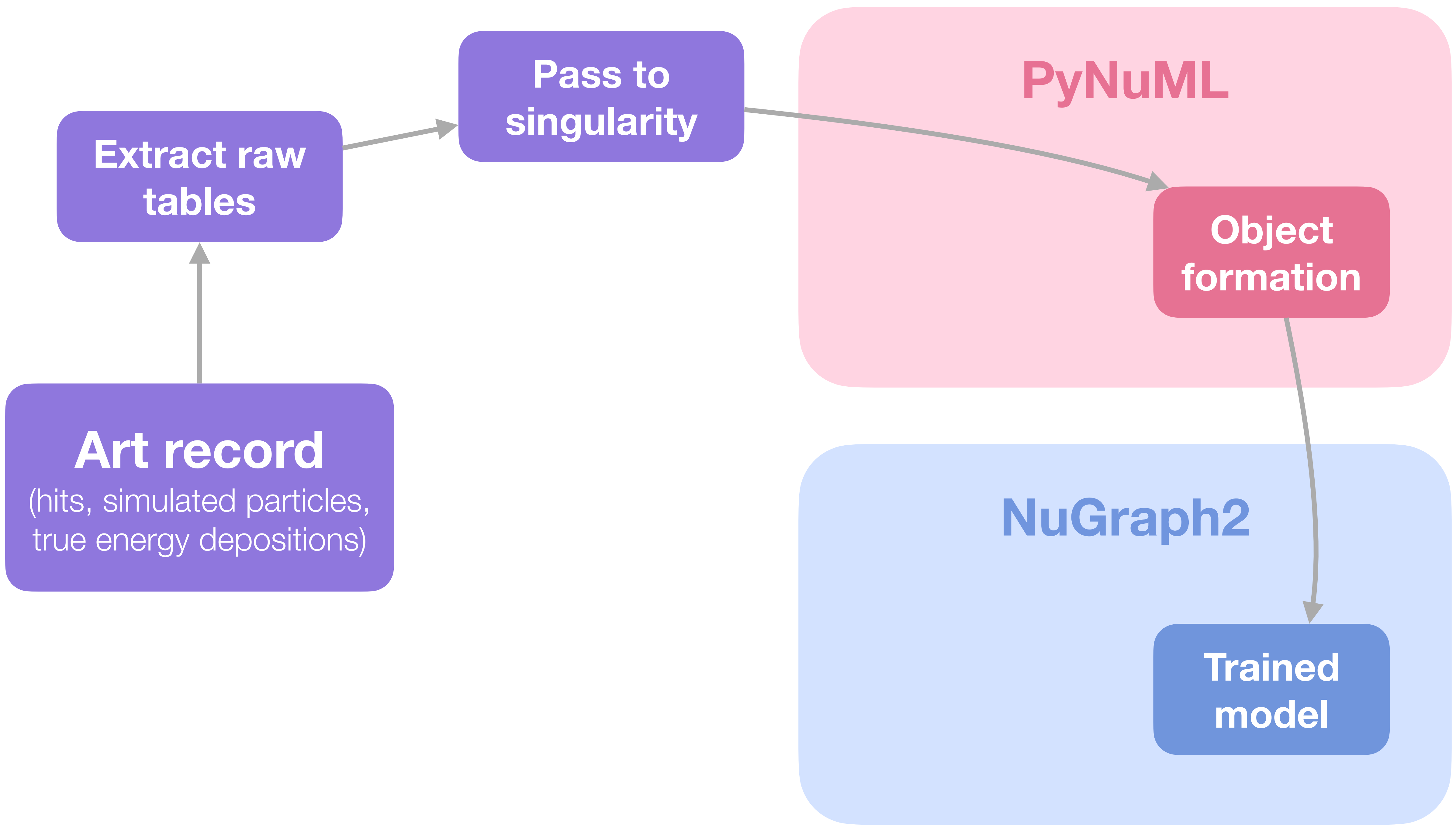
Inference in production



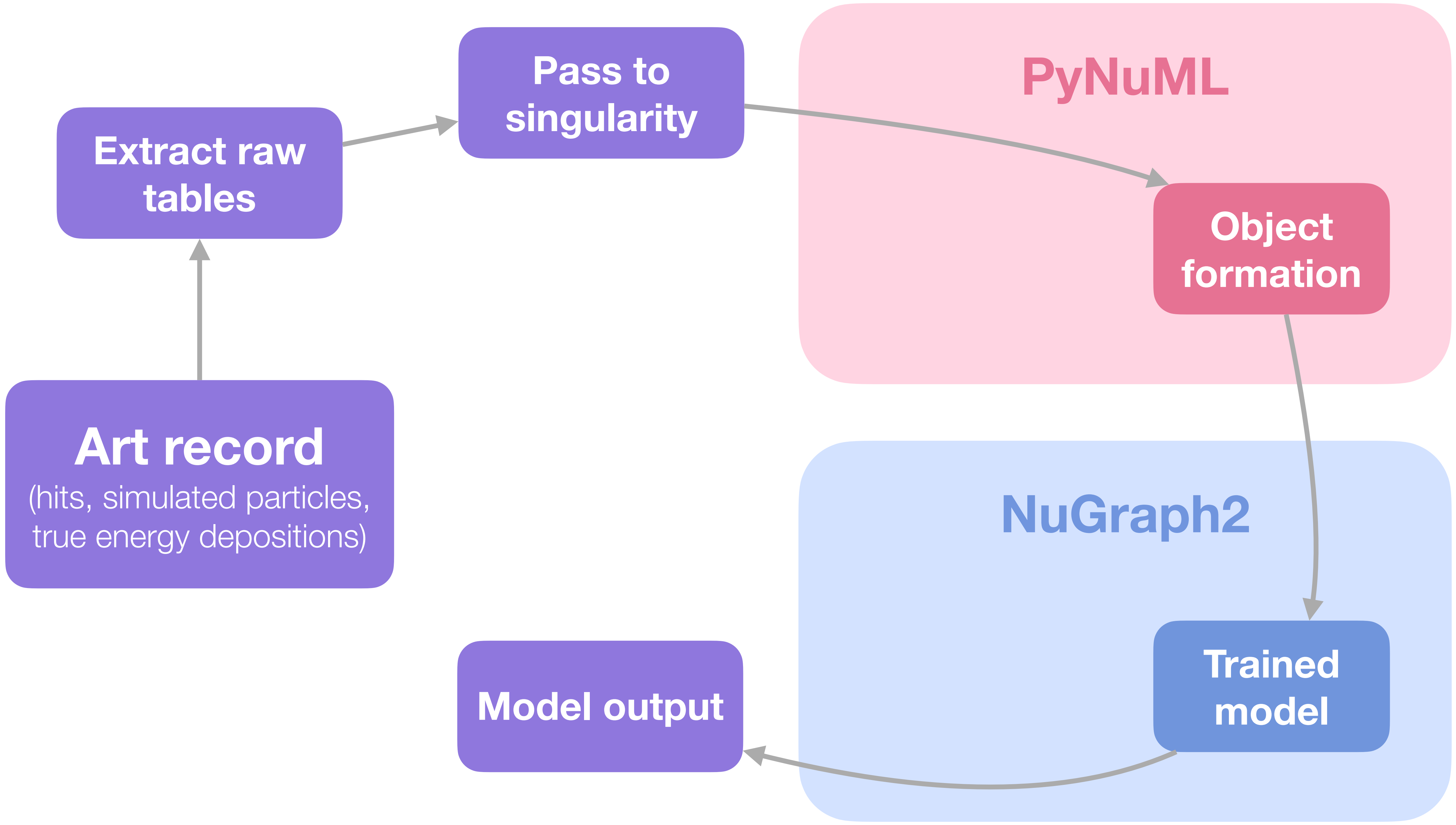
Inference in production



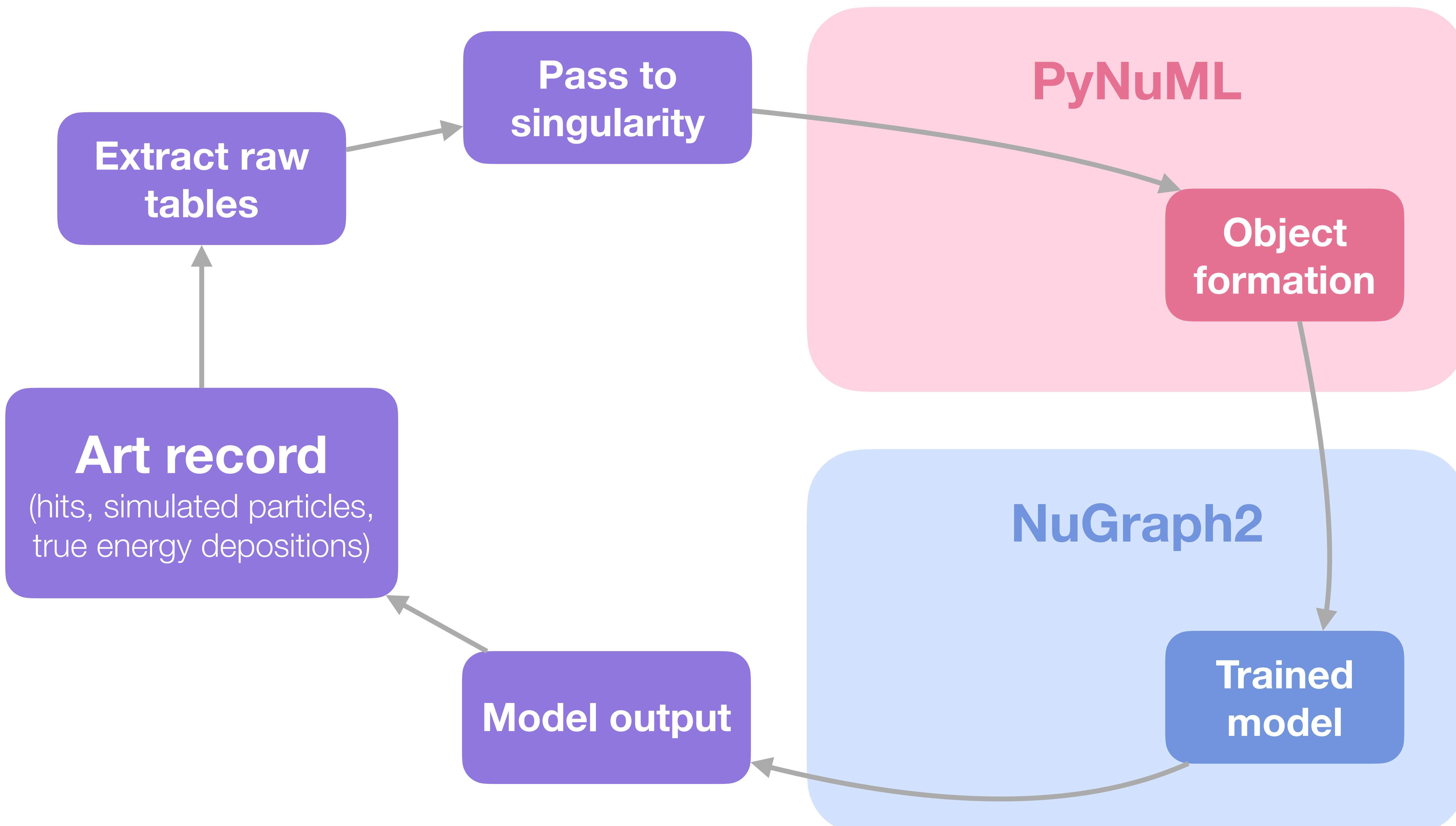
Inference in production



Inference in production



Inference in production



Inference in production