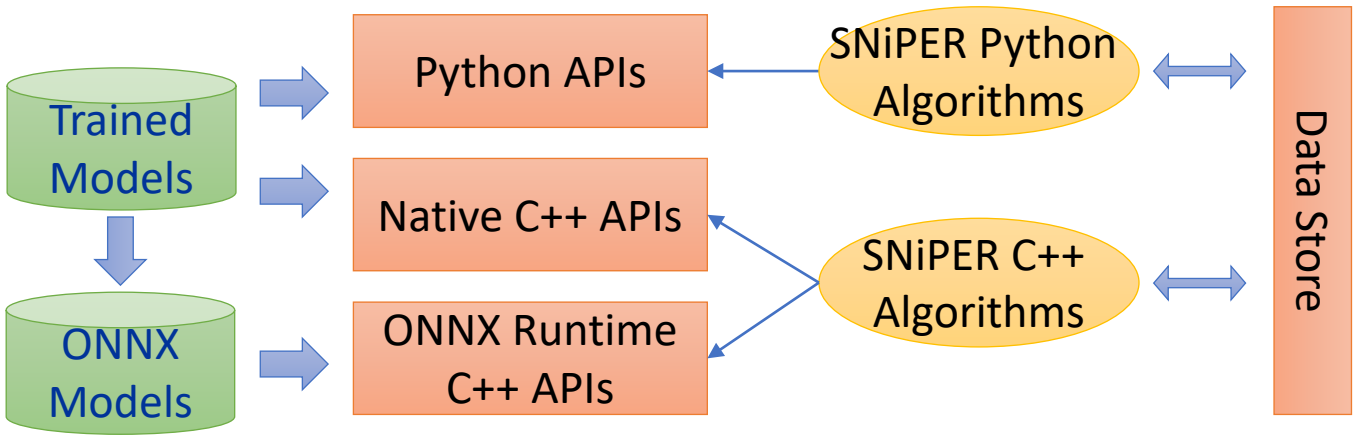
Integration of machine learning-trained models into JUNO's offline software

Tao Lin (on behalf of the JUNO Collaboration)

IHEP, CAS / lintao@ihep.ac.cn

Popular ML libraries are integrated into JUNOSW in C++ for **inference** with **three solutions**:

1. Python APIs and data sharing between C++ and Python
2. Native C/C++ APIs: TensorFlow, PyTorch
3. Open standard for ML: ONNX + ONNX Runtime (C++)



Keys to support ML integration (Link to Example)

numpy array: the data format used in C++ and Python

```
p::tuple shape = p::make_tuple(hit_col_size);
np::dtype dtype_int = np::dtype::get_builtin<int>();
np::ndarray arr_npe = np::zeros(shape, dtype_int);
```

PyDataStore: exchange data between C++ and Python algorithms

```
SniperDataPtr<PyDataStore> pystore(*getRoot(), "DataStore");
pystore->set("npe", arr_npe);
```

```
# In Python:
npe = self.datastore["npe"]
npe = torch.from_numpy(self.npe) # convert by torch
```

Authors: Tao Lin, Jiaheng Zou, Weidong Li, Xingtao Huang, Teng Li (on behalf of the JUNO Collaboration)

中国科学院高能物理研究所 Institute of High Energy Physics Chinese Academy of Sciences

SHANDONG UNIVERSITY