Hybrid Nuclear Interaction Models for Improved Fragmentation Modeling in Ion Therapy

European Nuclear Physics Conference 2025 - Caen, France

L. Arsini[1,2], S. Burrello[3], B. Caccia[4], A. Ciardiello[4], M. Colonna[3], S. Giagu[1,2], C. Mancini Terracciano[1,2]

[1]Department of Physics, Sapienza University of Rome, Rome, Italy. [2]INFN, Section of Rome, Rome, Italy.
[3]INFN, Section of LNS, Catania, Italy. [4]Istituto Superiore di Sanità, Rome, Italy.

22/09/2025

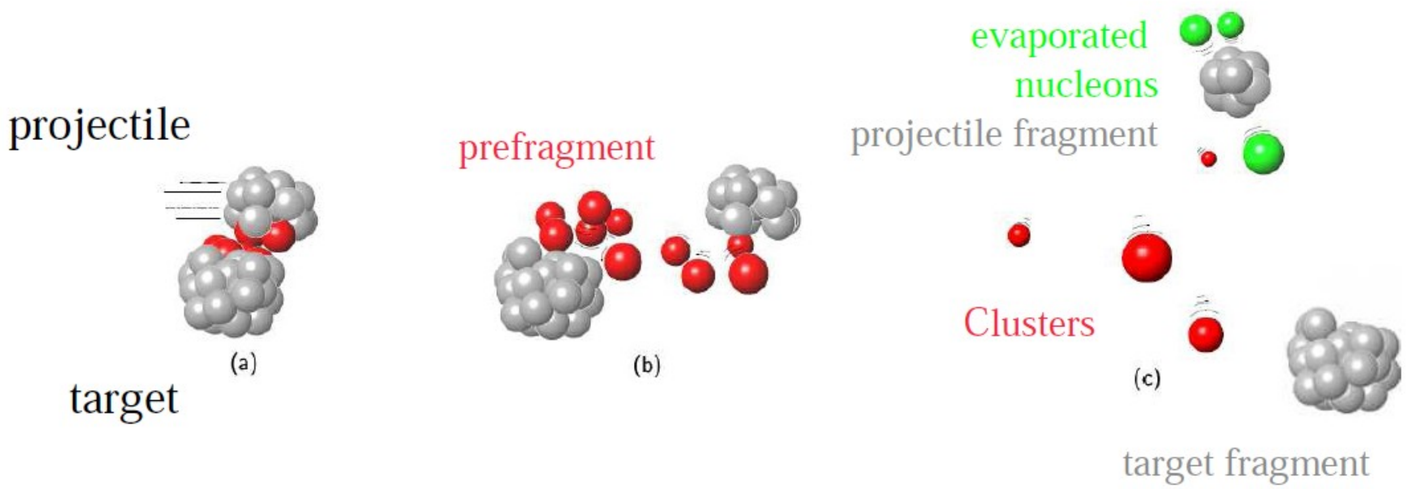
Nuclear interaction models

In Geant4

- Nuclear interaction models are **slow**
- In particular the most sophisticated ones e.g. **QMD**

Trade-off between computing **time** and **precision**

Use simpler models



Problems in Geant4 below 100 MeV/u

No dedicated model to nuclear interaction **below 100 MeV/u** in Geant4

Many papers showed discrepancies:

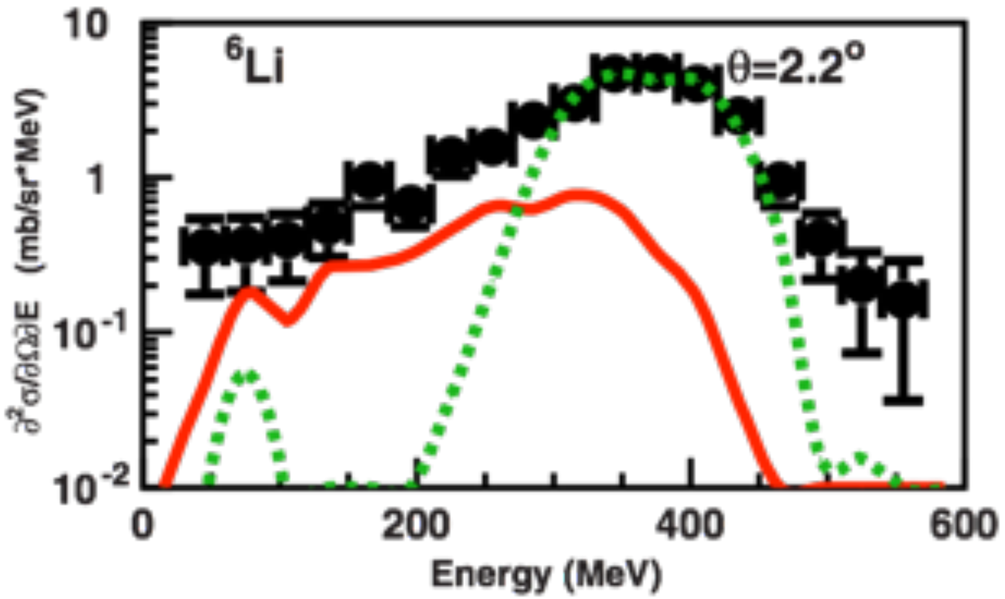
Braunn et al. : one order of magnitude in 12C fragmentation at 95 MeV/u on thick PMMA target

De Napoli et al. : angular distribution of the secondaries emitted in the interaction of 62 MeV/u 12C on thin carbon target

Dudouet et al. : similar results with a 95 MeV/u 12C beam on H, C, O, Al and Ti targets

- **Exp. data**
- **G4-BIC**
- **G4-QMD**

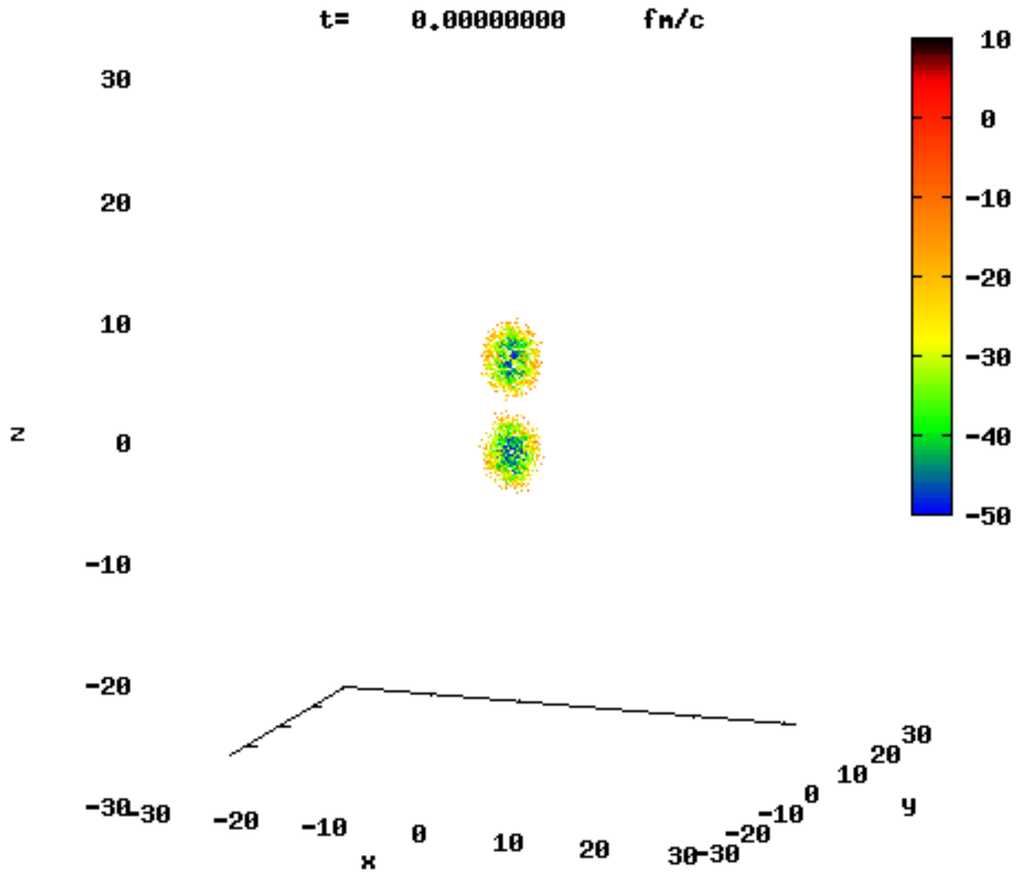
[Plot from De Napoli et al. Phys. Med. Biol., vol. 57, no. 22, pp. 7651–7671, Nov. 2012]



Cross section of the ^{6}Li production at 2.2 degree in a ^{12}C on ^{nat}C reaction at 62 MeV/u.

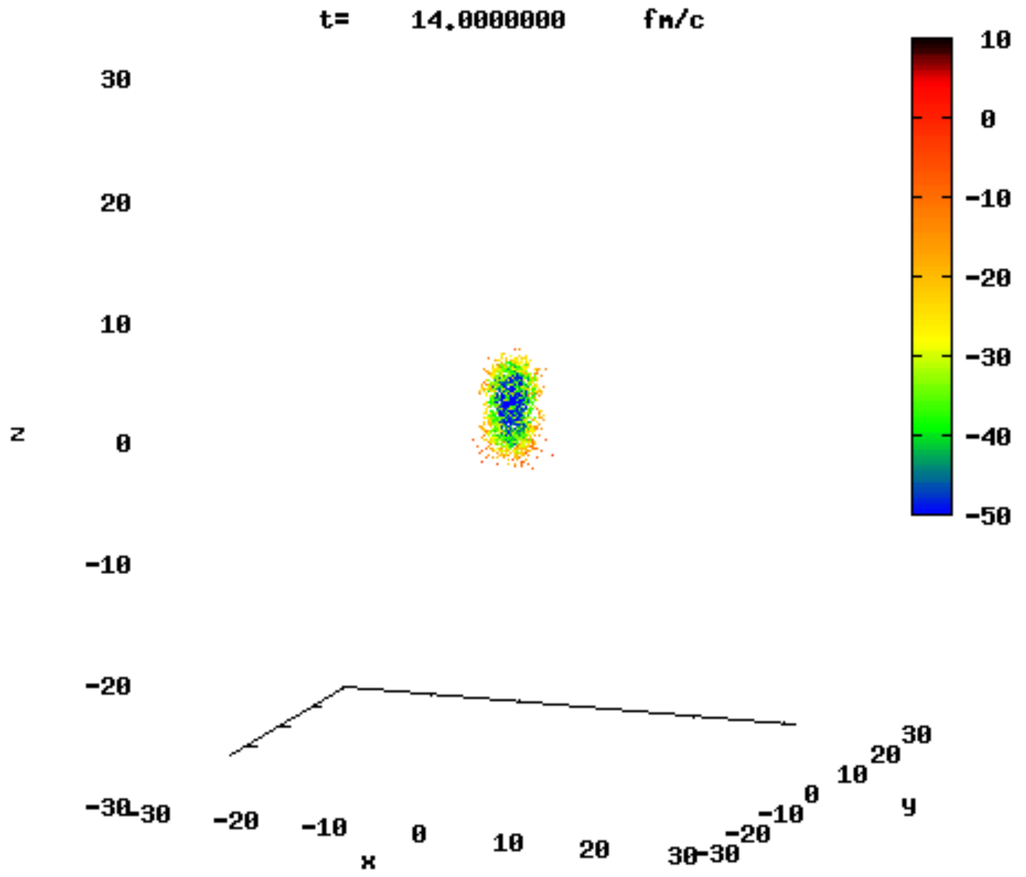
BLOB (Boltzmann-Lagevein One Body)

- Test-particle approach
- Self-consistent **mean field** + collisions
- Probability to find a nucleon in the phase space



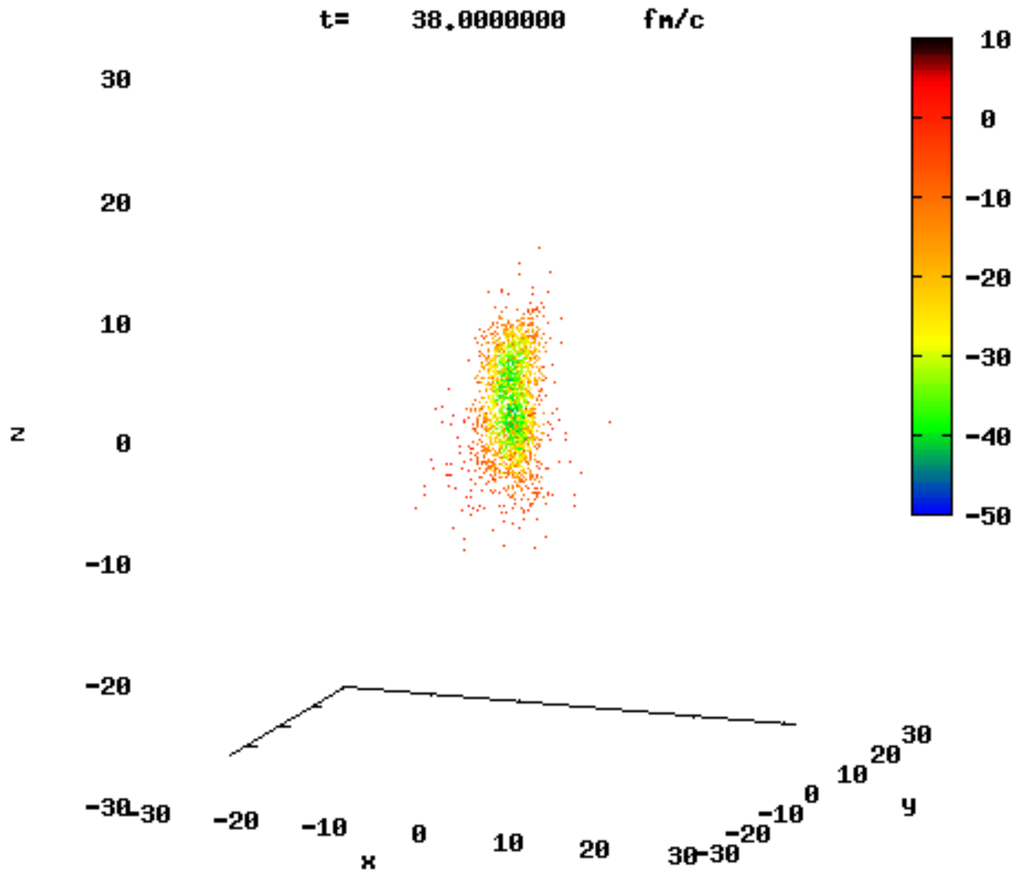
BLOB (Boltzmann-Lagevein One Body)

- Test-particle approach
- Self-consistent **mean field** + collisions
- Probability to find a nucleon in the phase space



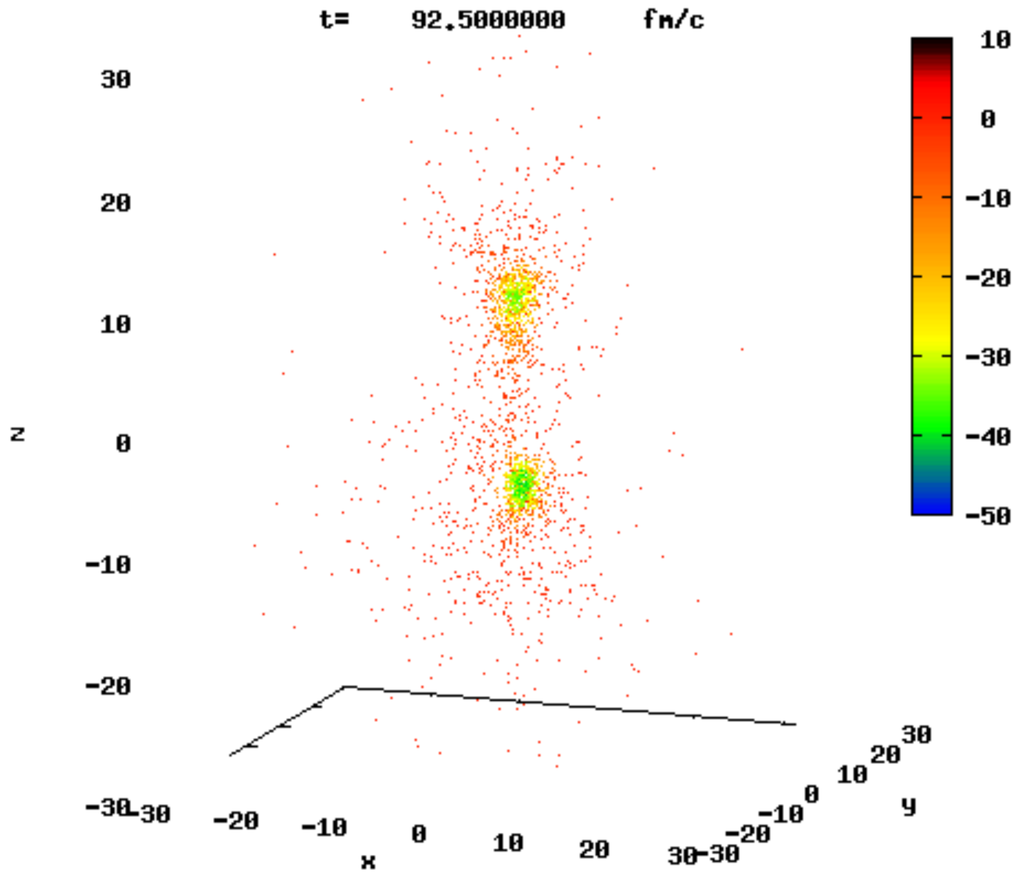
BLOB (Boltzmann-Lagevein One Body)

- Test-particle approach
- Self-consistent **mean field** + collisions
- Probability to find a nucleon in the phase space



BLOB (Boltzmann-Lagevein One Body)

- Test-particle approach
- Self-consistent **mean field** + collisions
- Probability to find a nucleon in the phase space



BLOB (Boltzmann-Lagevein One Body)

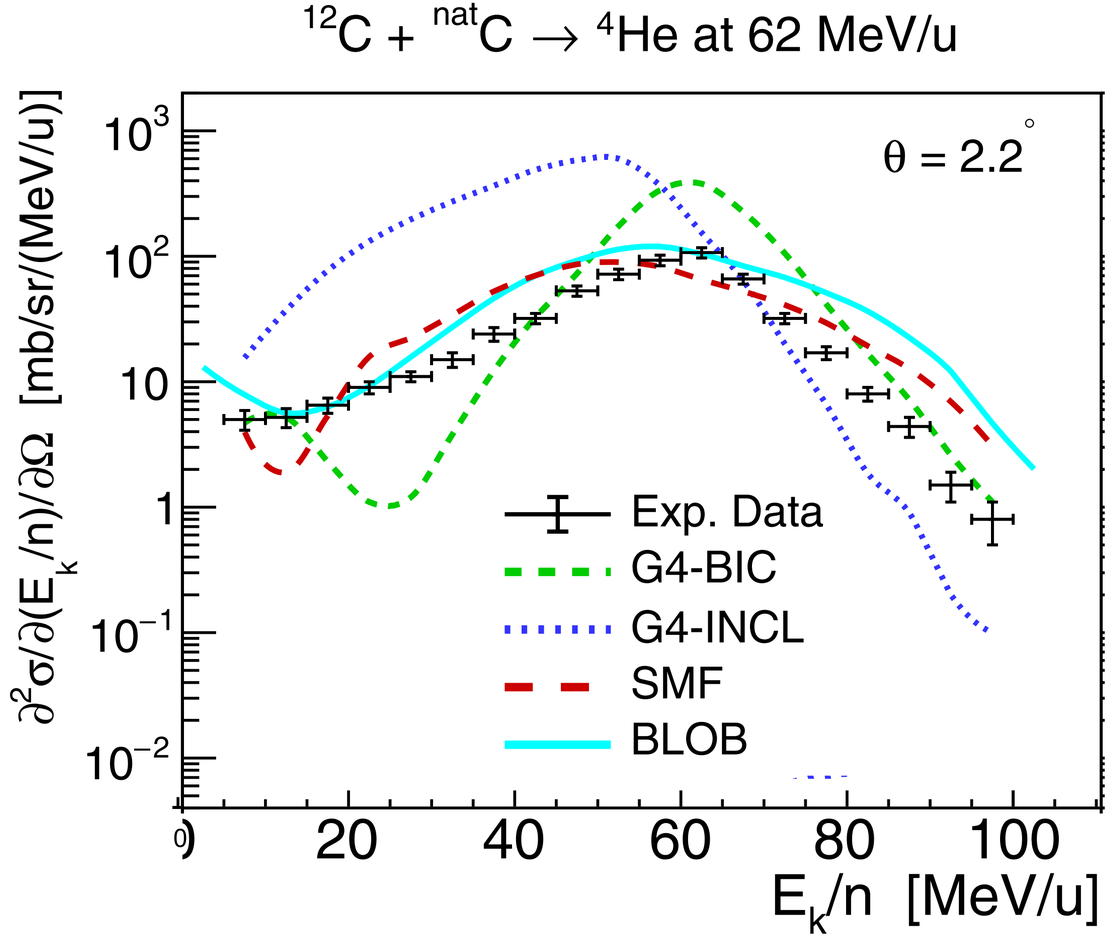
We interfaced BLOB with Geant4 and its de-excitation model

[C. Mancini-Terracciano et al. Preliminary results coupling “Stochastic Mean Field” and “Boltzmann-Langevin One Body” models with Geant4. In: Physica Medica 67 (2019), pp. 116–122. doi: 10.1016/j.ejmp.2019.10.026.]

Accurate

Slow

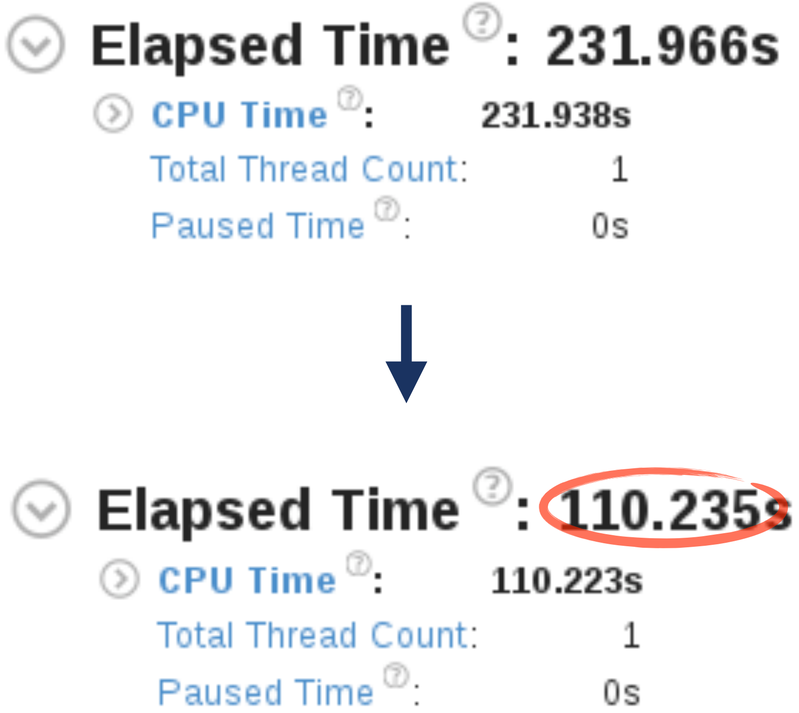
Order of minutes per interaction!



Blob code optimization

- We optimised BLOB without changing the code structure (52% speed-up overall)

Not enough for practical applications

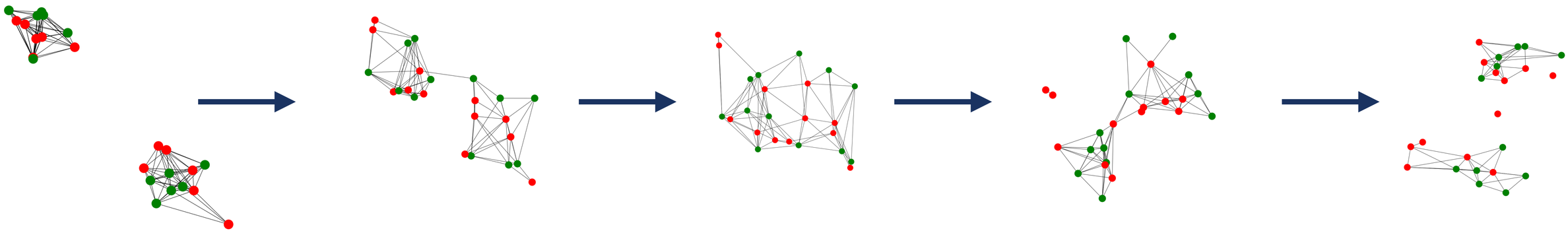


Deep Learning to accelerate NIMs

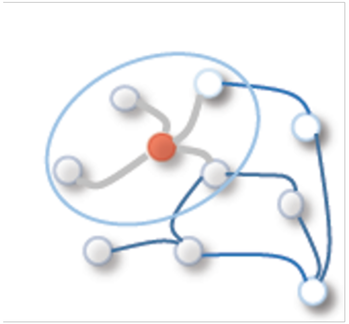
Starting from a proof-of-concept study on **G4-QMD**

Graph Networks Simulator approach

Train a **Graph Neural Network** to emulate the **interaction dynamics**

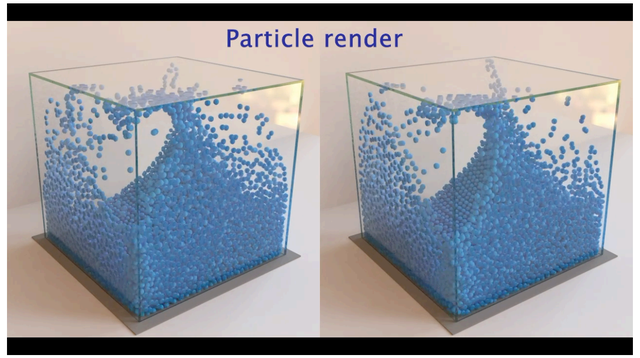


Each nucleon is a node of the graph



Convolve the central node's representation with its neighbours' representations

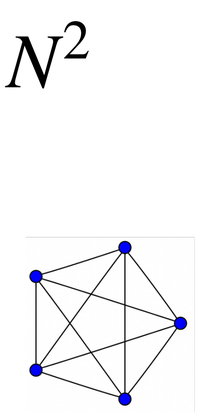
[A. Sanchez-Gonzalez et al. "Learning to simulate complex physics with graph networks." PMLR, 2020.]

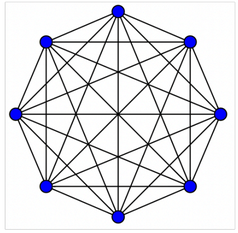


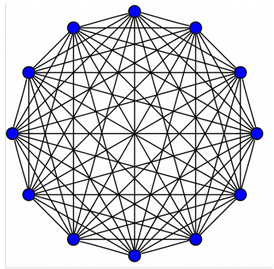
Drawbacks

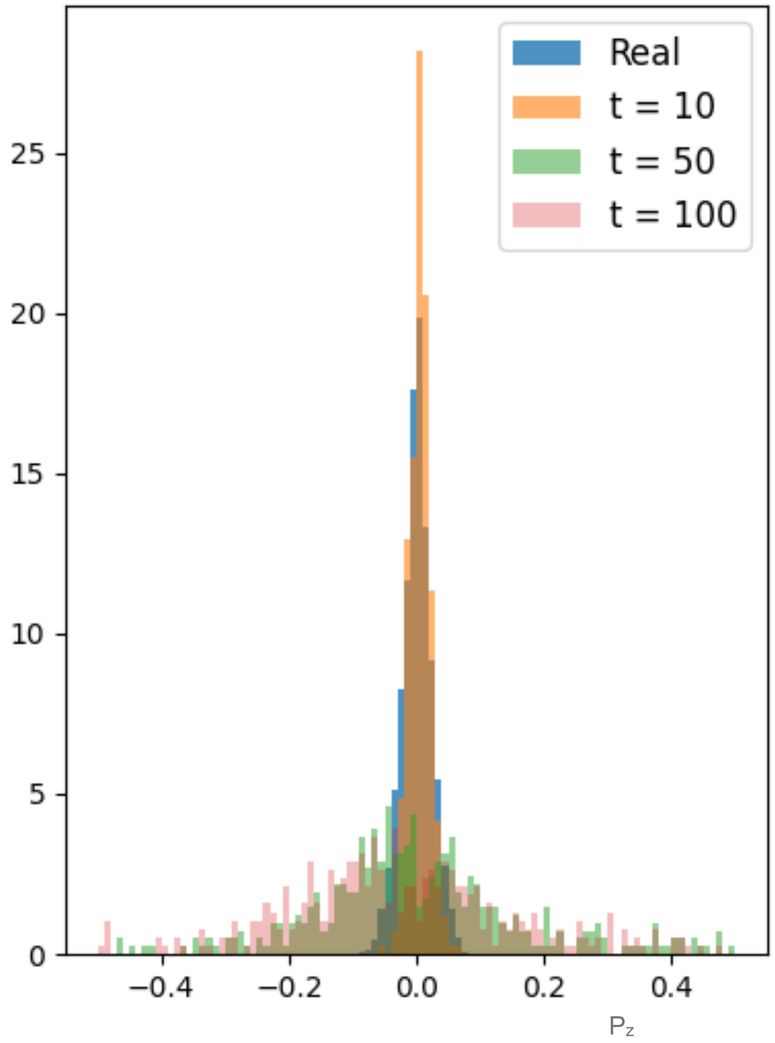
- Quantities are conserved on average
- Comparable performance to Euler method

- Results with fully connected graph
- $N_{edges} \propto N^2$









Going Hybrid

Why?

- Approximating complex functions with **Neural Networks**
- Leveraging **GPU acceleration** for ultra-fast execution

How?

- Building **Physics-inspired** architectures → Physics under control; **Explainability**

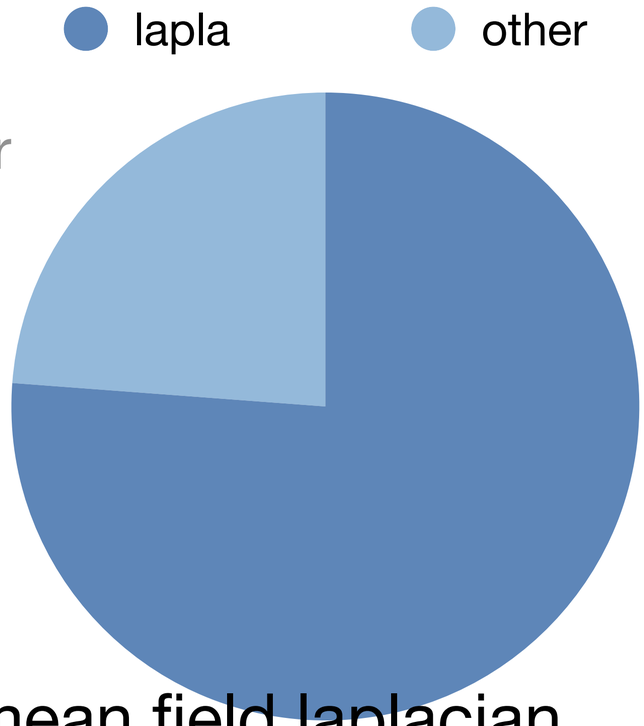
What to emulate?

The Potential

It is the Bottleneck

Profiling BLOB

with Intel VTune Amplifier



~ 4 mins per interaction

3 mins: computing mean field laplacian

Elapsed Time: 110.235s

CPU Time:	110.223s
Total Thread Count:	1
Paused Time:	0s

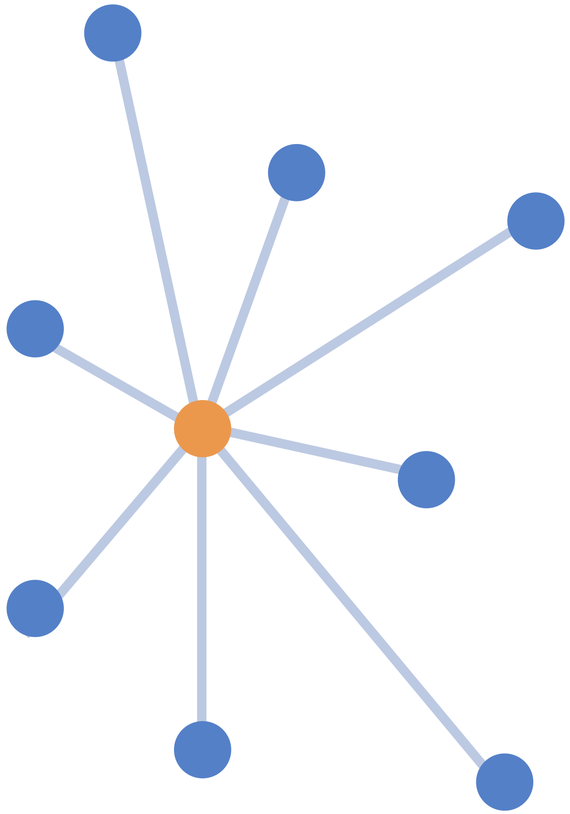
Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time
lapla	run	56.086s
erff	libm.so.6	17.038s
define_two_clouds_rp	run	9.051s
sortrx	run	7.450s
powf	libm.so.6	5.184s
[Others]		15.414s

Learning the Potential: DL model

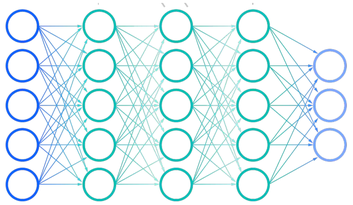
Particle-wise MLP for Potential Prediction



In QMD

$$V_i = \sum A_{ij} + \left(\sum B_{ij} \right)^{\gamma}$$

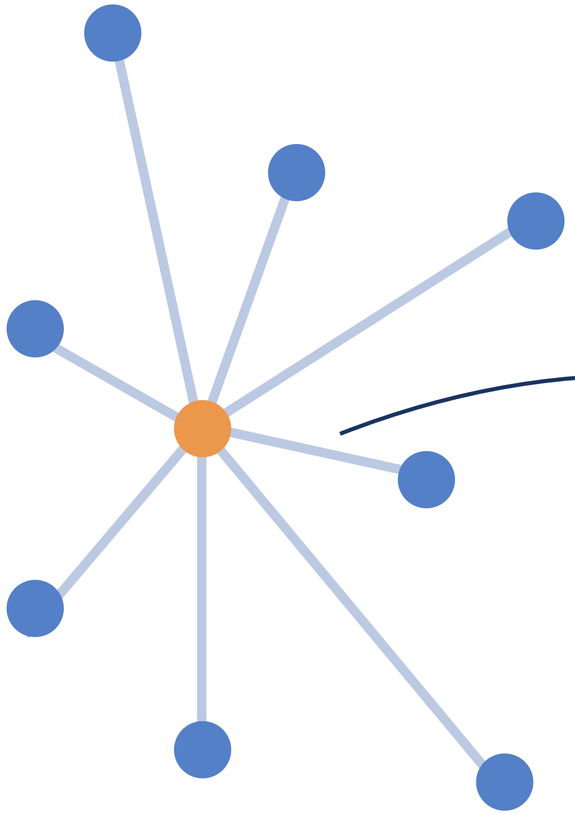
$$f\left(q_i, q_j, p_i, p_j, c_i, c_j\right) =$$



MLP

Learning the Potential: DL model

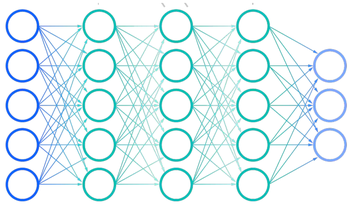
Particle-wise MLP for Potential Prediction



In QMD

$$V_i = \sum A_{ij} + \left(\sum B_{ij} \right)^{\gamma}$$

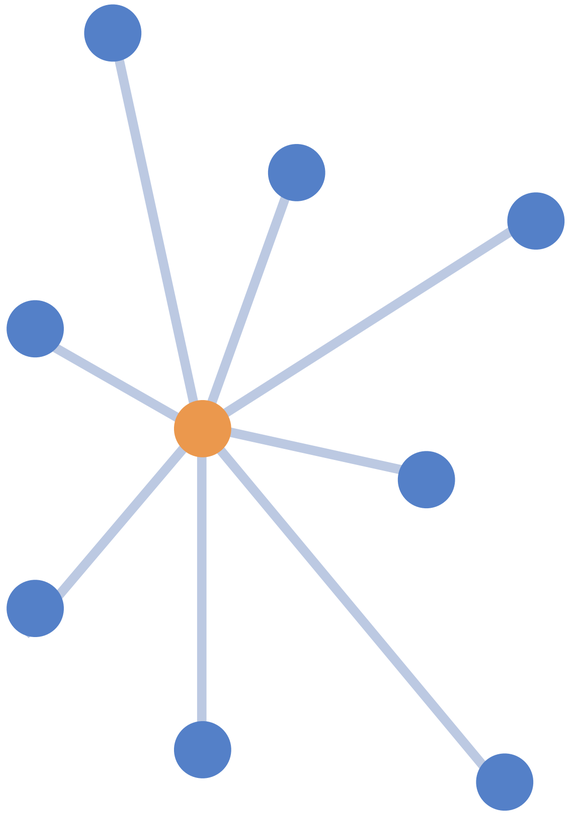
$$f\left(q_i, q_j, p_i, p_j, c_i, c_j\right) =$$



MLP

Learning the Potential: DL model

Particle-wise MLP for Potential Prediction



Building a DL model which:

- is **coherent** with the Physics

  Two body interactions embedded in the architecture

- works with **any** number of particles

  Particles are treated in **batch**

Potential Predictions

C12 on C12 at 62 MeV/u

Model:

5 layers MLP + ReLu + LayerNorm

Data:

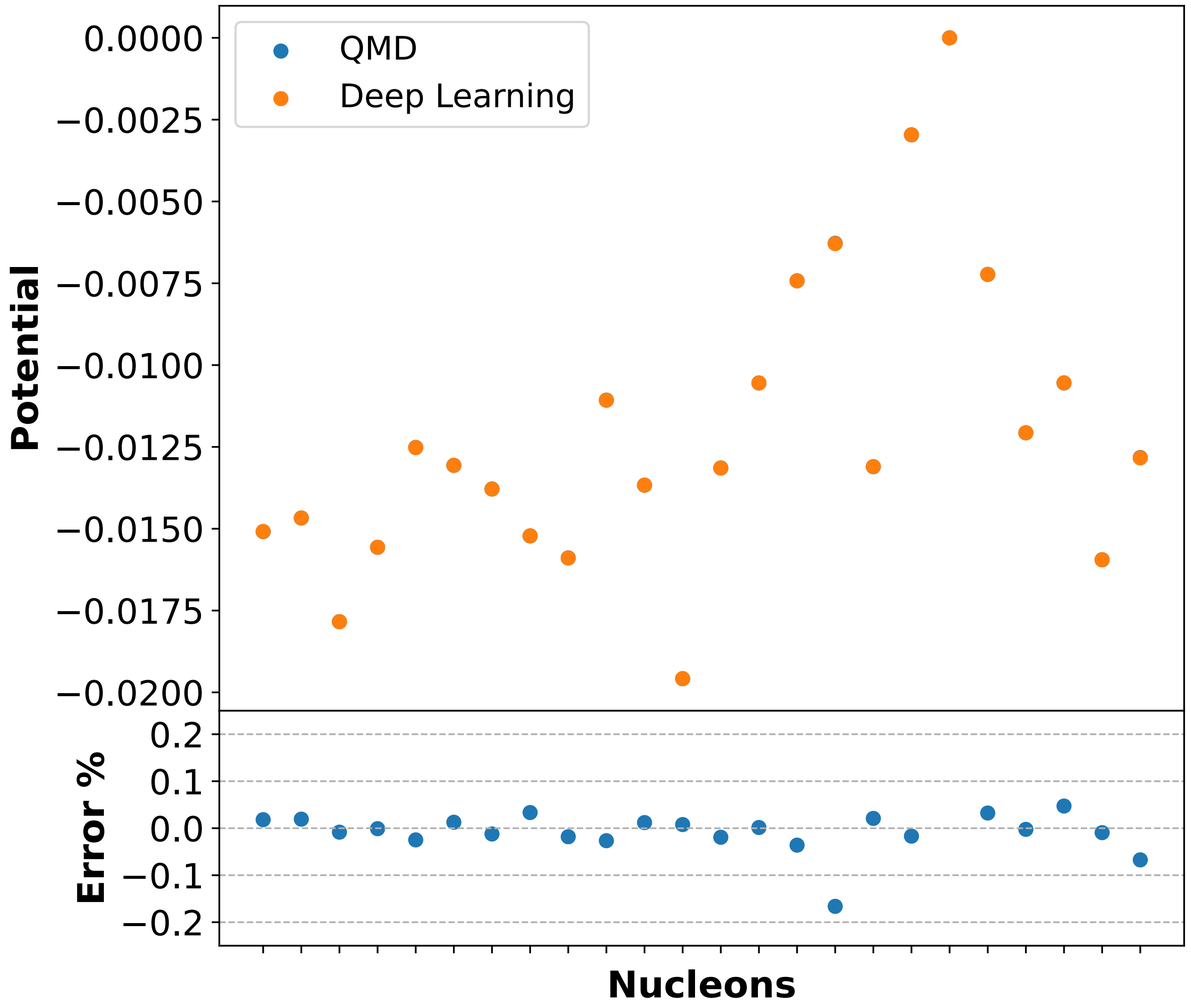
23k stories

10 events

24 particles : ~5 M examples

Training: ~3d training on Nvidia V100

Results: Median Relative Error 0,05 %



Implementation in Geant4

Exporting the DL models from pytorch to **ONNX**

Using ONNX C++ API ⟶ substituting GetPotential() Method in QMD

```
G4double MyQMDMeanField::GetPotential_dl( G4int i )
{
   // ------------------------PREDICT WITH DEEP LEARNING ---------------------------
  return static_cast<G4double>( ONNXInterface::GetInstance()->Generate(i, system)[0]);
   // ------------------------------------------------------------------------------
}
```

Thread-safe implementation

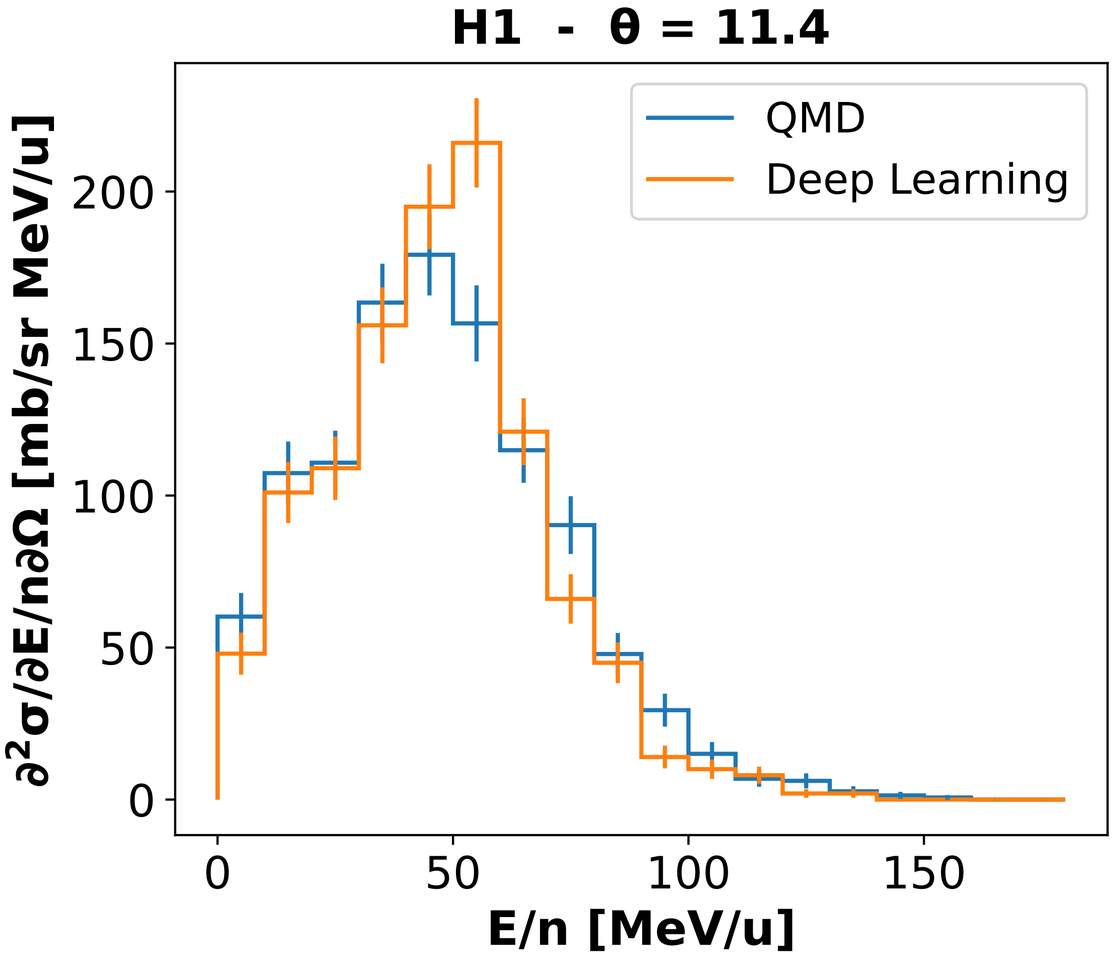
Test on the Potential

Simulating the reaction:

C12 on C_nat at 62 MeV/u

Interfacing DL model with Geant4

- Reasonable accuracy on double differential cross section of **lighter fragments**



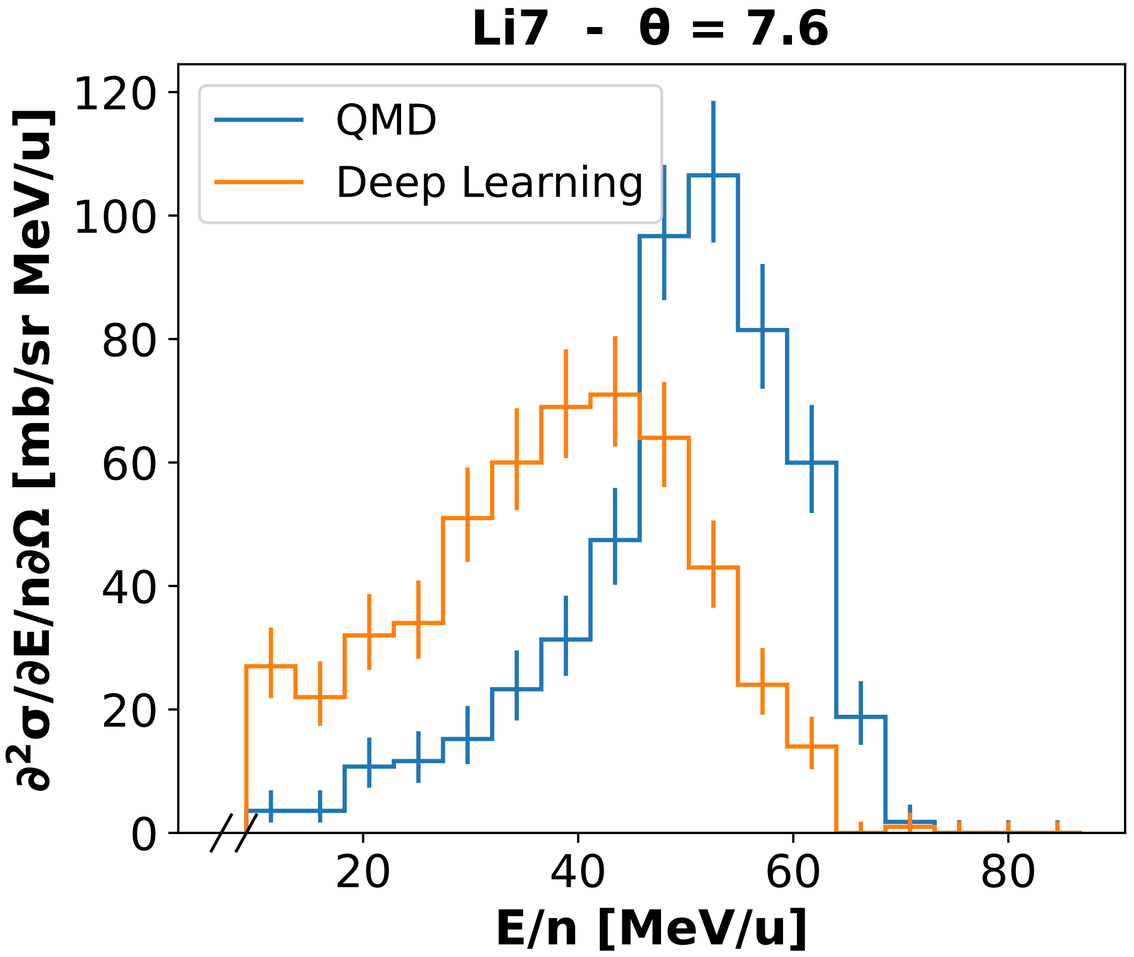
Test on the Potential

However: for heavier fragments

- Even **small errors** on the potential **propagate** badly to the double differential cross sections

- It is not the bottleneck!

Only 4% of QMD execution time



Another possibility

Derivatives of the Hamiltonian

Emulating $\frac{\partial H}{\partial q}, \frac{\partial H}{\partial p}$

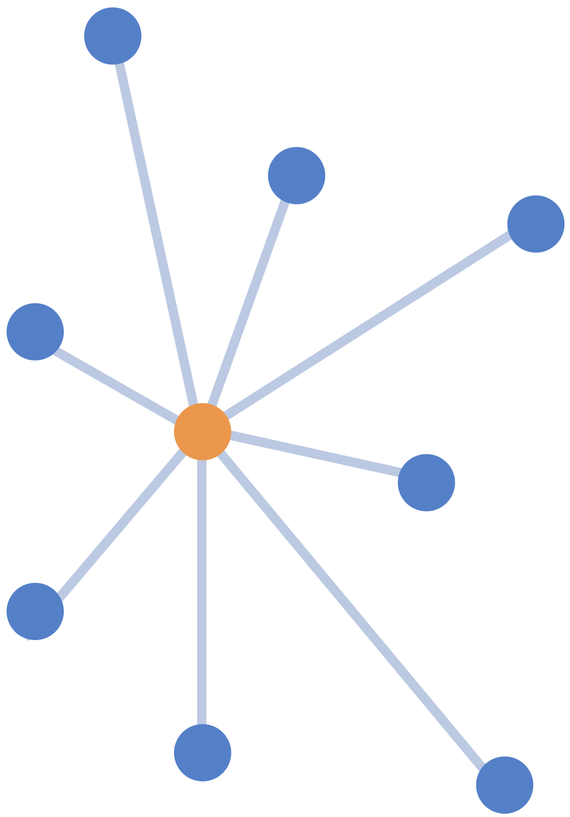
1) Cross sections are **resilient** to 1-2% errors

Callees	CPU Time: Total ▼
MyQMDReaction::ApplyYourself	100.0%
G4QMDMeanField::DoPropagation	88.7%
G4QMDMeanField::CalGraduate	47.5%
G4QMDMeanField::Cal2BodyQuantities	40.5%

2) This is the **bottleneck**!

CalGraduate() is **50%** of QMD

Emulating the derivatives



Same architectural design of the Potential model

$$\frac{\partial H}{\partial q, p} \approx \sum A_{ij} + \sum_{\alpha^{(k)}} \left(\sum B_{ij}^{(k)} \right)^{\alpha^{(k)}}$$

Approximating the derivatives

Hyper-parameter optimization on the number of **terms K**

Derivatives prediction

Model: 2 $\alpha^{(k)}$ terms +
5 layers MLP + ReLu + LayerNorm

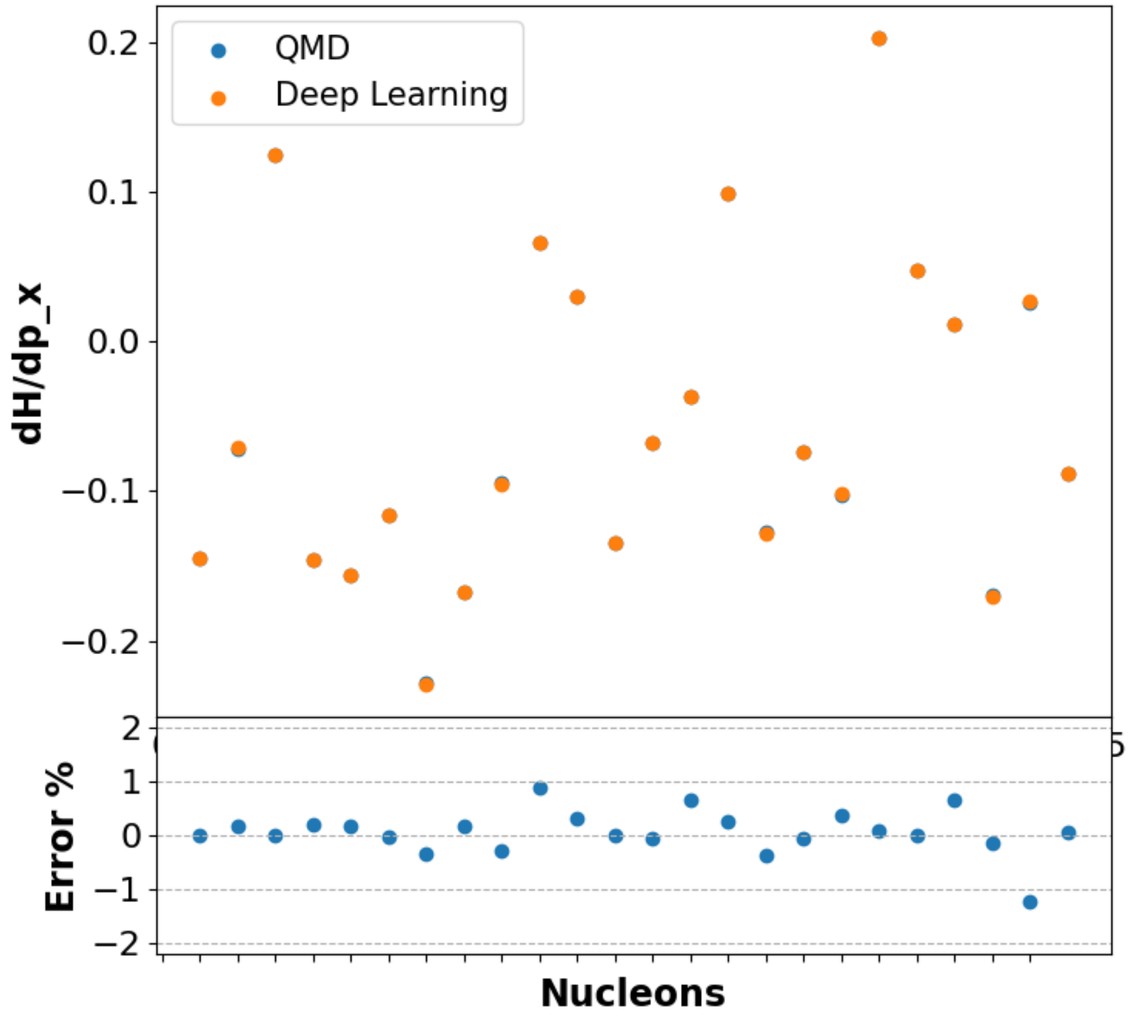
Data:

12k stories
1 events
24 particles : ~300k examples

Training: ~3h training on Nvidia V100

Results: Median Relative Error 0,6 %

C12 on C12 at 62 MeV/u



Implementation in Geant4

Exporting the DL models from pytorch to **ONNX**

Using ONNX C++ API → substituting CalGraduate() Method in QMD

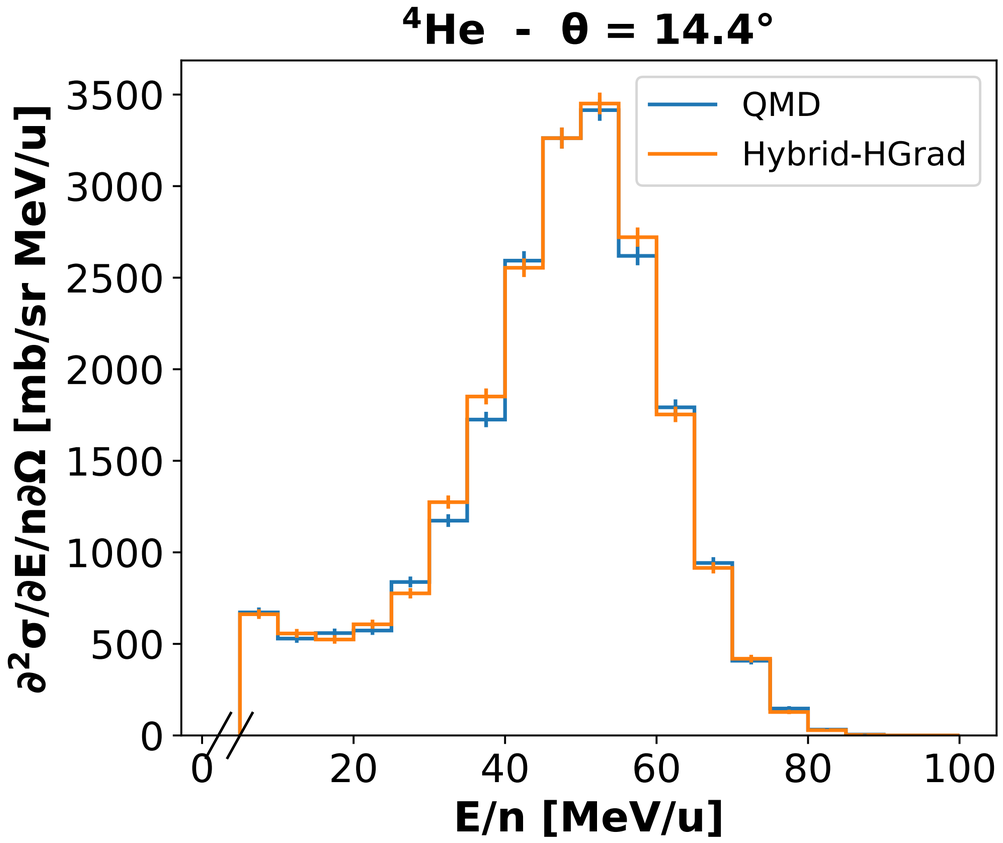
```
void MyQMDMeanField::CalGraduate_dl()
{
   ffr.resize( system->GetTotalNumberOfParticipant() );
   ffp.resize( system->GetTotalNumberOfParticipant() );

  // ---------------- PREDICT WITH DEEP LEARNING ----------------
  auto gradients = ( ONNXInterface::GetInstance()->Generate(system));
  ffr = gradients[0];
  ffp = gradients[1];

}
```

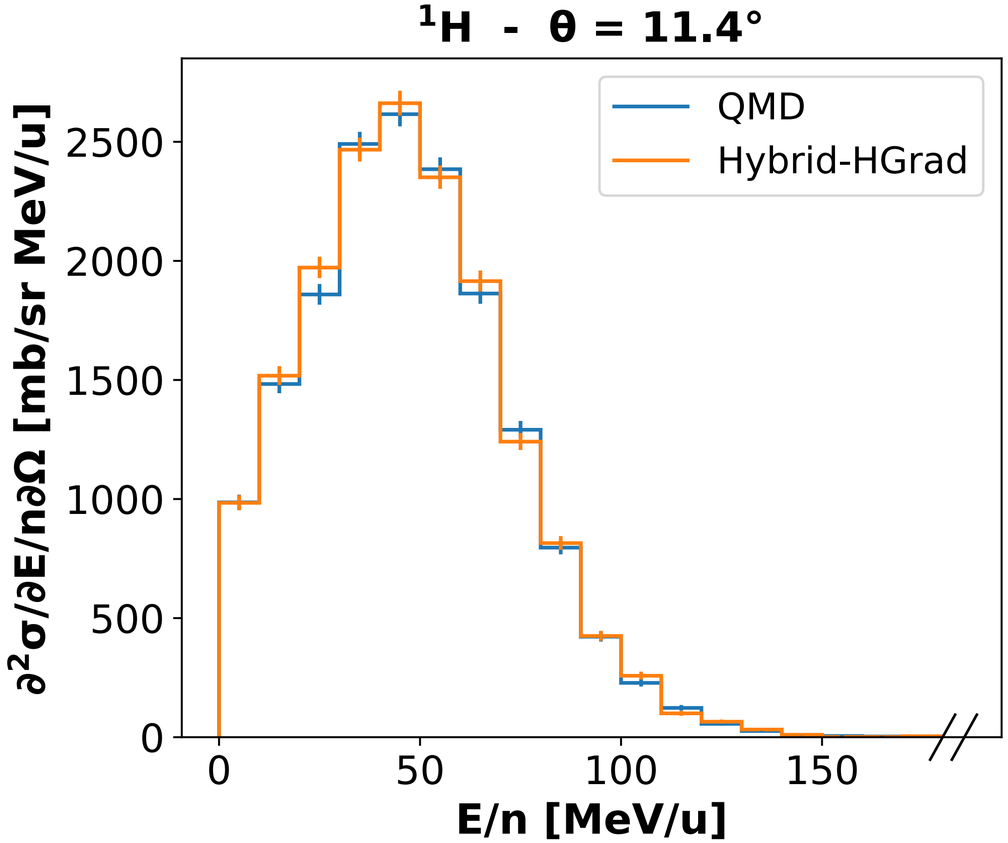
Thread-safe
implementation

Double differential cross sections

Emulating H Grad

^{12}C on ^{nat}C at 62 MeV/u



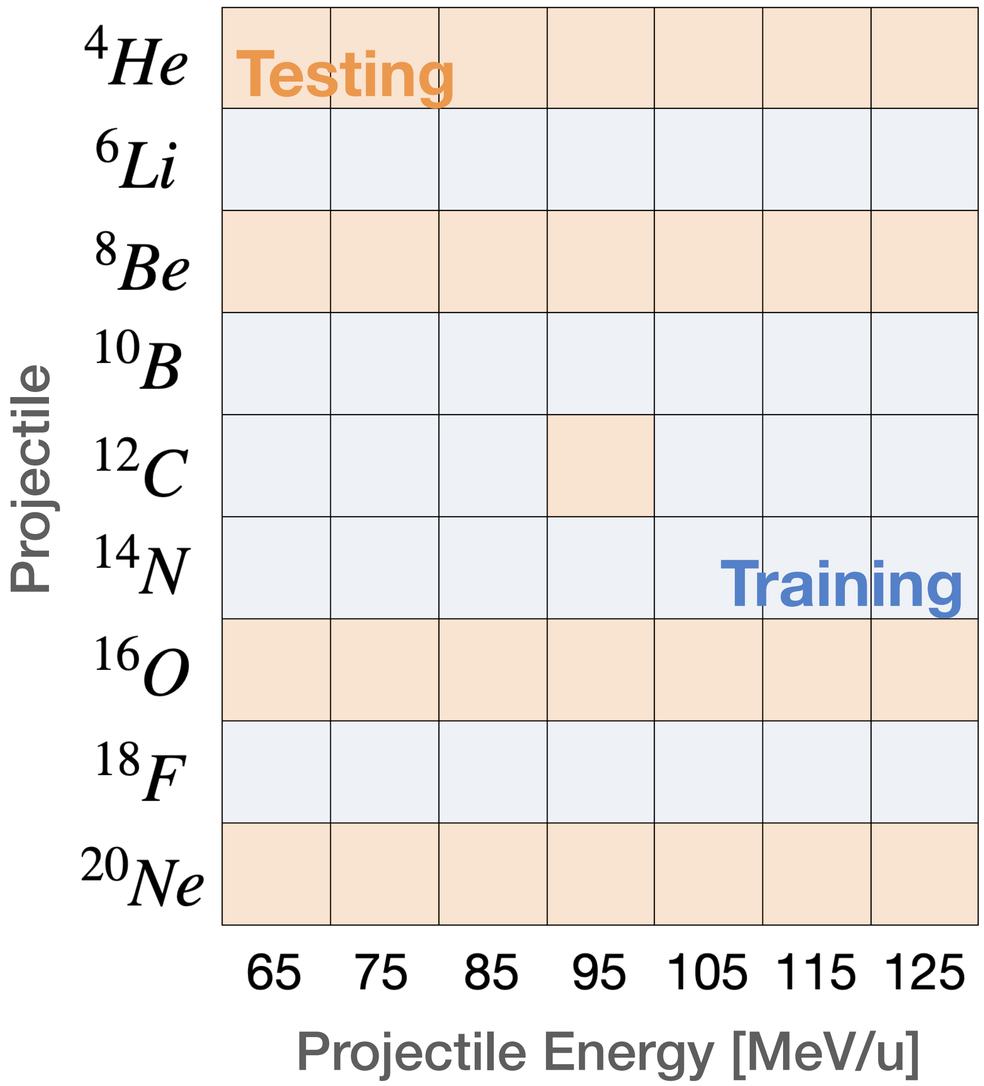


Multi Ion Training

- Extension to multi ions and energies
- Extension to **Light Ion QMD**

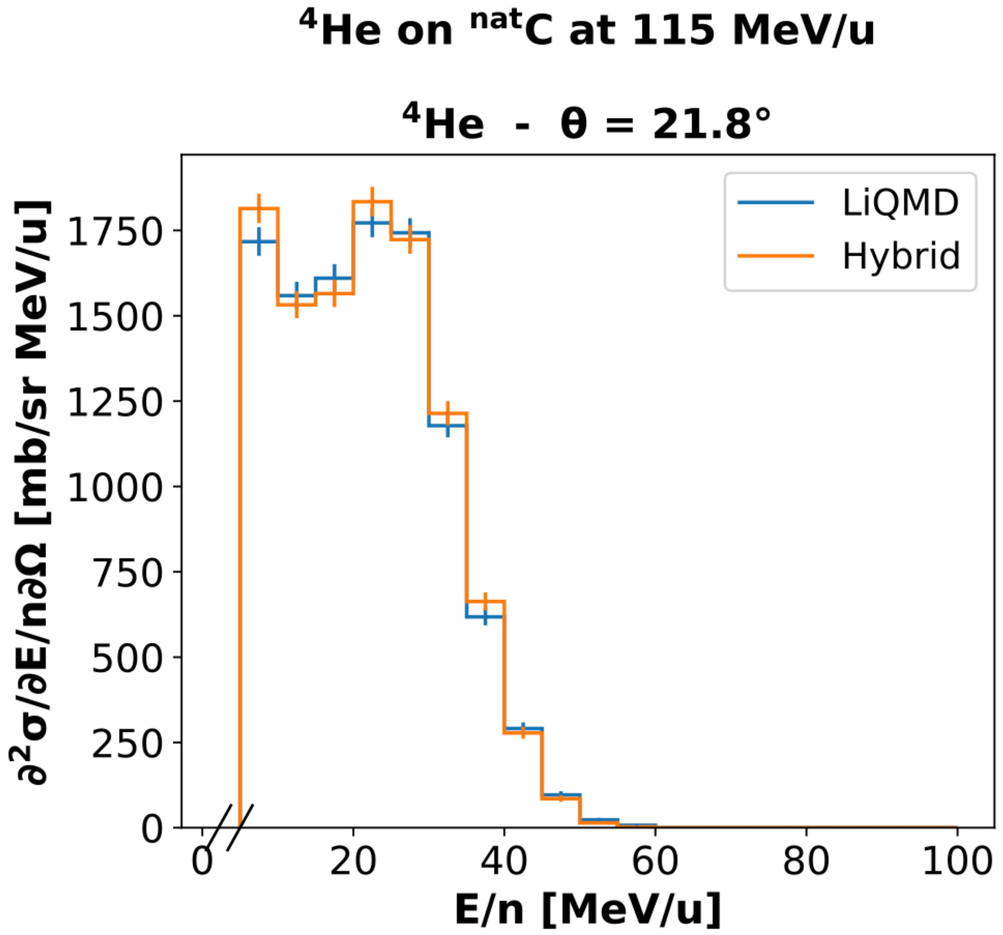
Recent QMD optimization

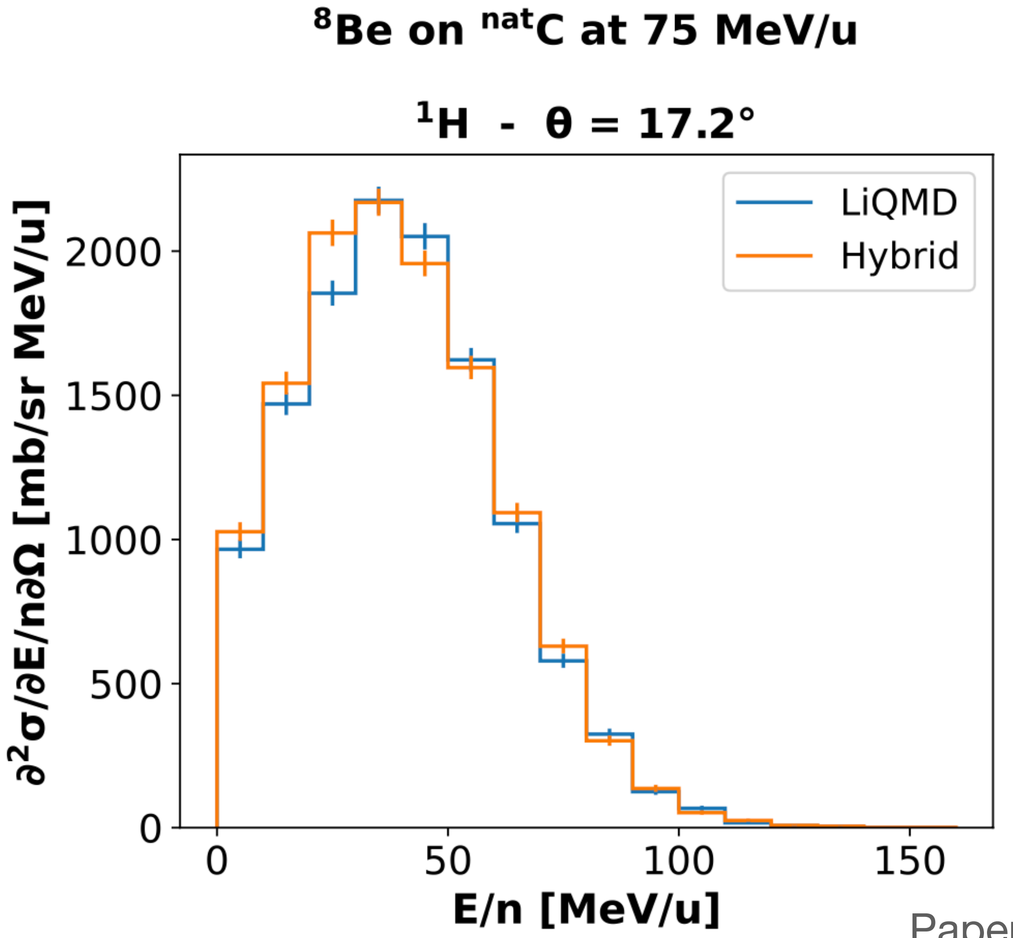
$$\frac{\partial H}{\partial q, p} \approx \sum A_{ij} + \sum_{\alpha^{(k)}} \left(\sum B_{ij}^{(k)} \right)^{\alpha^{(k)}}$$



Double differential cross sections

Test set

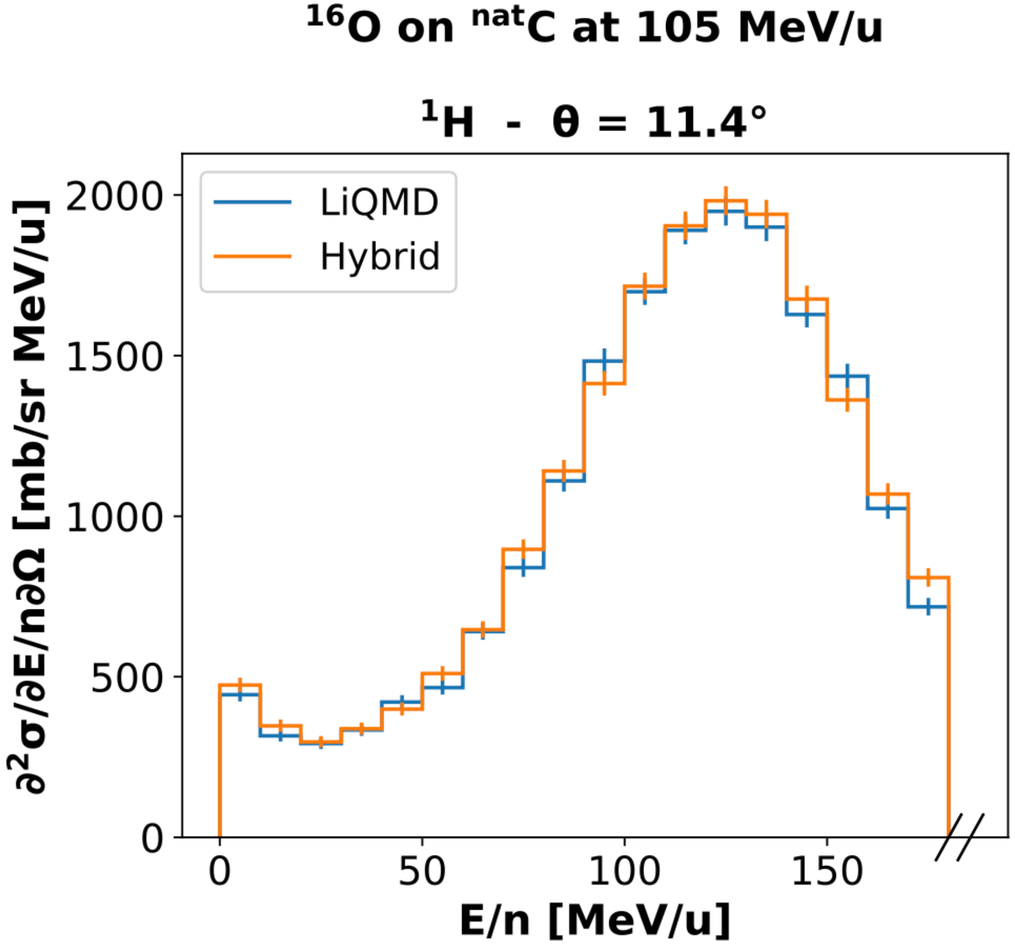


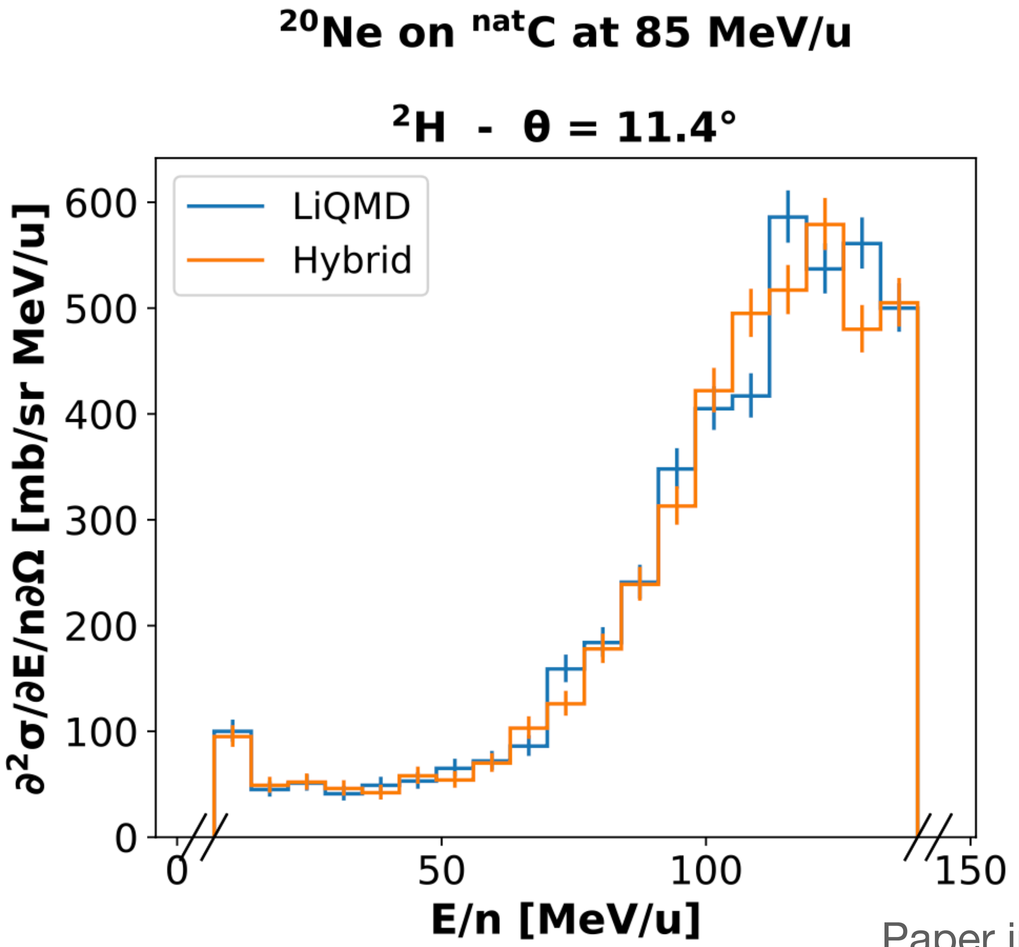


Paper in preparation

Double differential cross sections

Test set





Paper in preparation

Next steps

Extension to BLOB

Next steps

Extension to BLOB

Full DL modeling

Inputing the potential removes the need for fully connected geometry

Emulating tracking and collisions with GNN

Next steps

Extension to BLOB

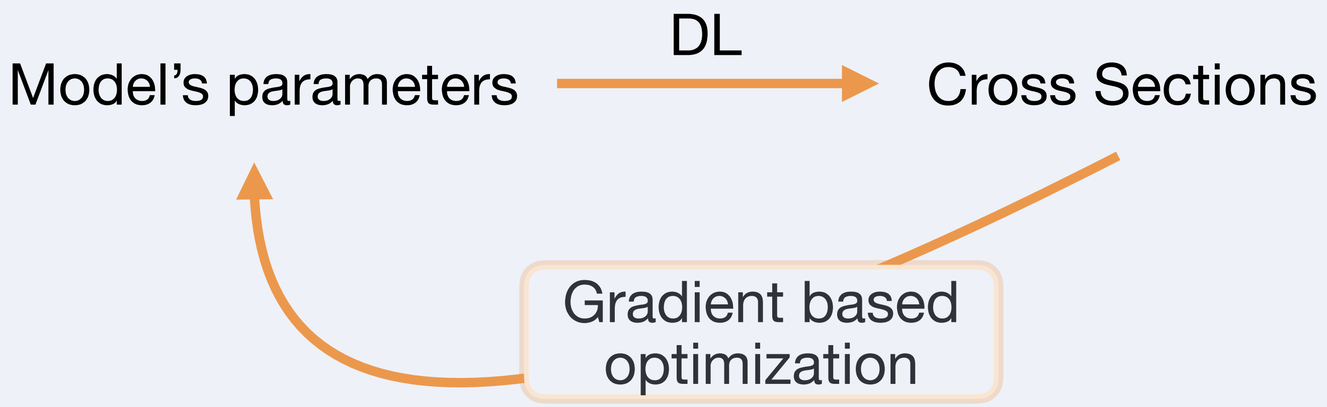
Full DL modeling

Inputing the potential removes the need for fully connected geometry

Emulating tracking and collisions with GNN

QMD and LiQMD Optimisation

Fully differentiable pipeline:



Emulating de-excitation model

Thank you for your attention!

- Nuclear interaction models in Geant4:
  - Sophisticated models are slow
  - No dedicated model under 100 MeV/u
- Deep Learning approach for model emulation
  - Emulation of Hamiltonian derivatives with DL for QMD
  - Multi ion training to achieve generalization
  - Possible model optimization or speed-up

Lorenzo Arsini **22-09-2025**
lorenzo.arsini@uniroma1.it

SAPIENZA UNIVERSITÀ DI ROMA
ISTITVTO SVPERIORE DI SANITÀ
INFN

Backup

SAPIENZA
UNIVERSITÀ DI ROMA

INFN

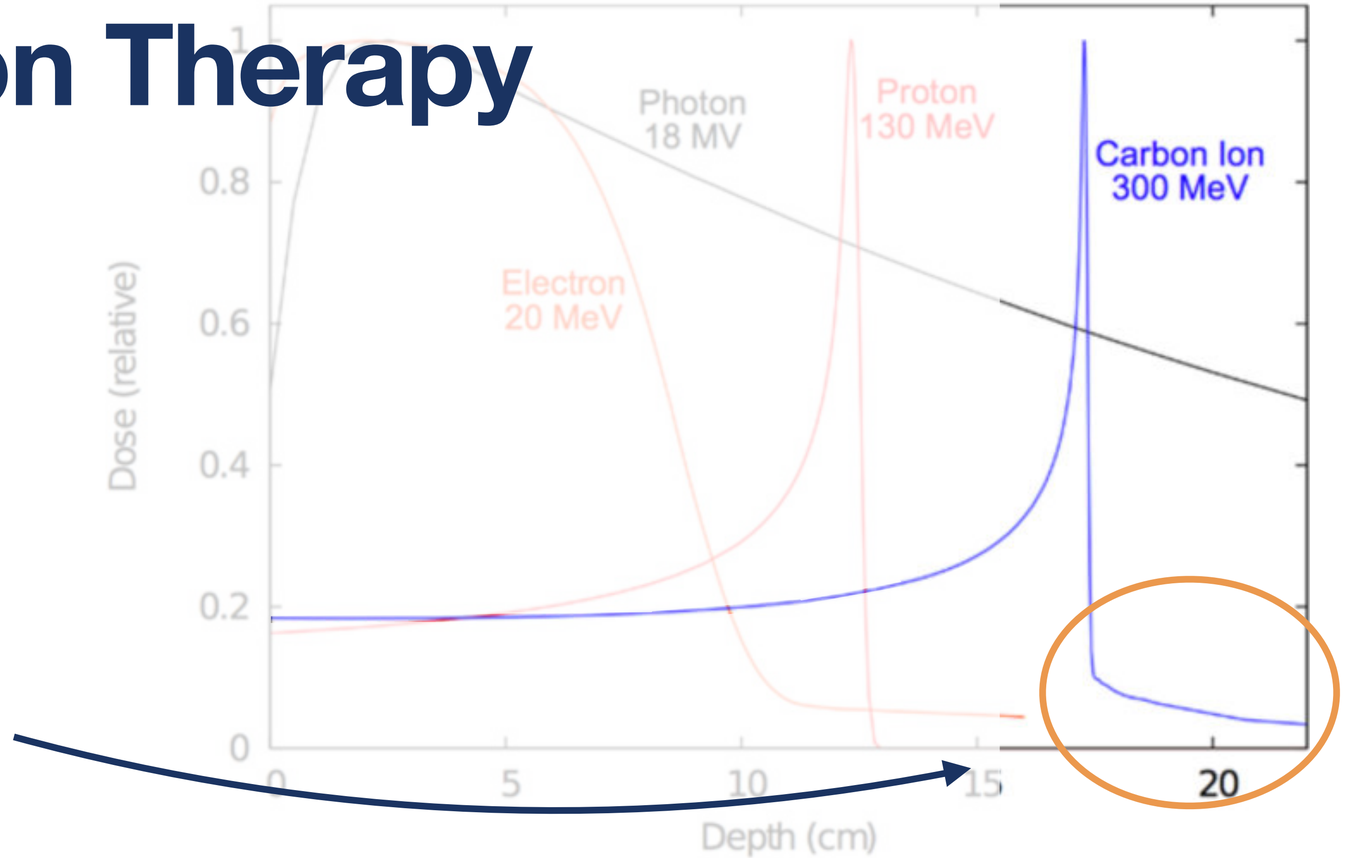
Carbon Ion Radiation Therapy

High biological effectiveness

Subject to fragmentation

↓

Responsible for the tail in the dose distribution

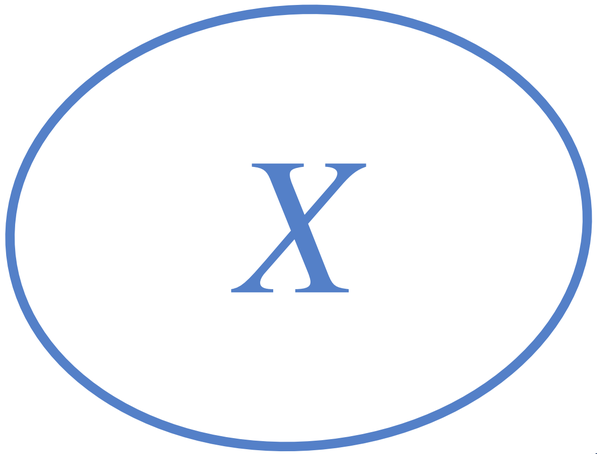


Accurate modeling of fragmentation is crucial

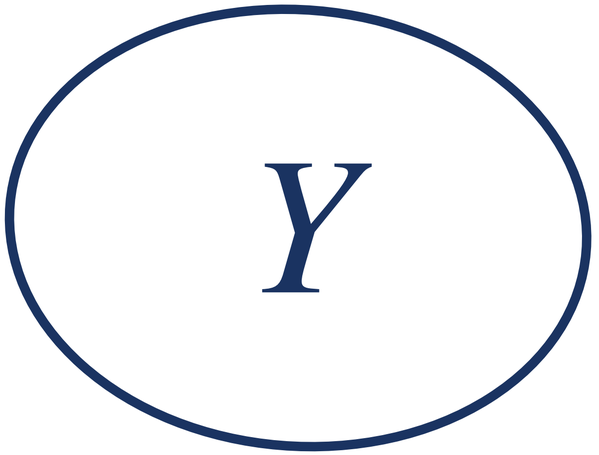
Lack of dedicated nuclear interaction models under 100 MeV/u

Deep Learning

$$f(X) = Y$$

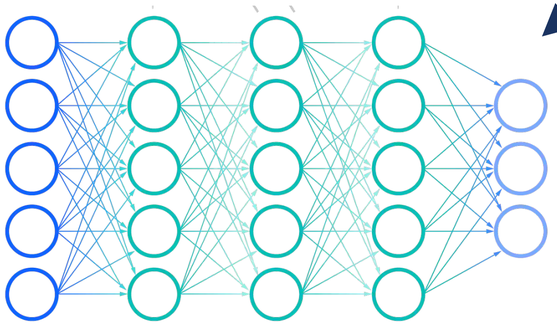


X



Y

$$\tilde{f}_w(X) = Y$$



Neural Network

Analytical gradients

$$\nabla_w \tilde{f}_w(X)$$

wrt **parameters**

$$\nabla_X \tilde{f}_w(X)$$

wrt **input**

Differentiability

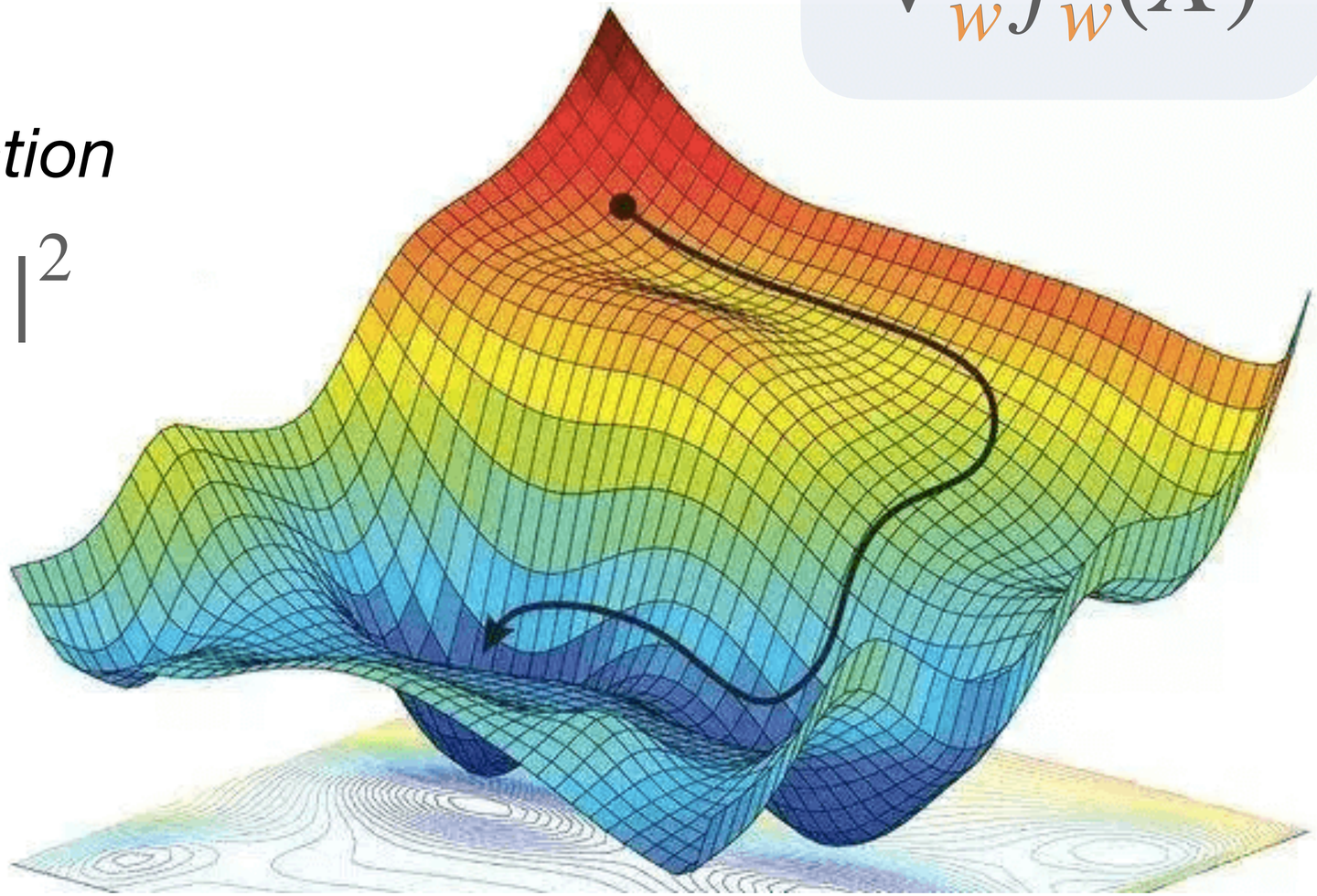
Training a Neural Network

$$\nabla_w \tilde{f}_w(X)$$

Minimization of a *Loss Function*

$$\mathcal{L}_w = |f(X) - \tilde{f}_w(X)|^2$$

Gradient descent



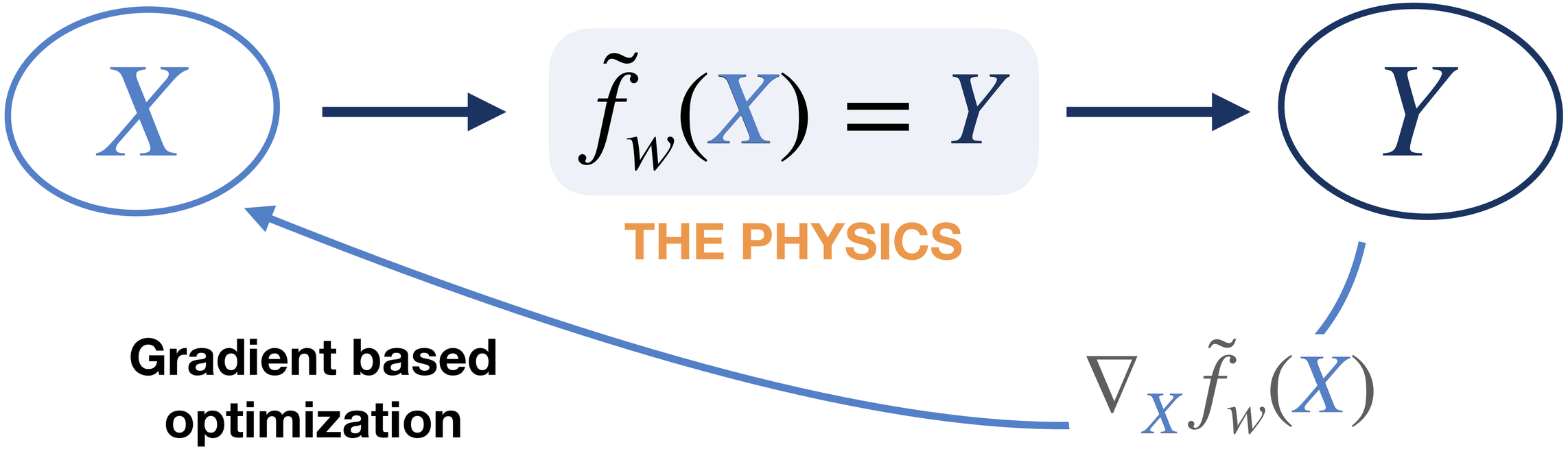
$$\min_w \mathcal{L}_w$$

Differentiability

After the training

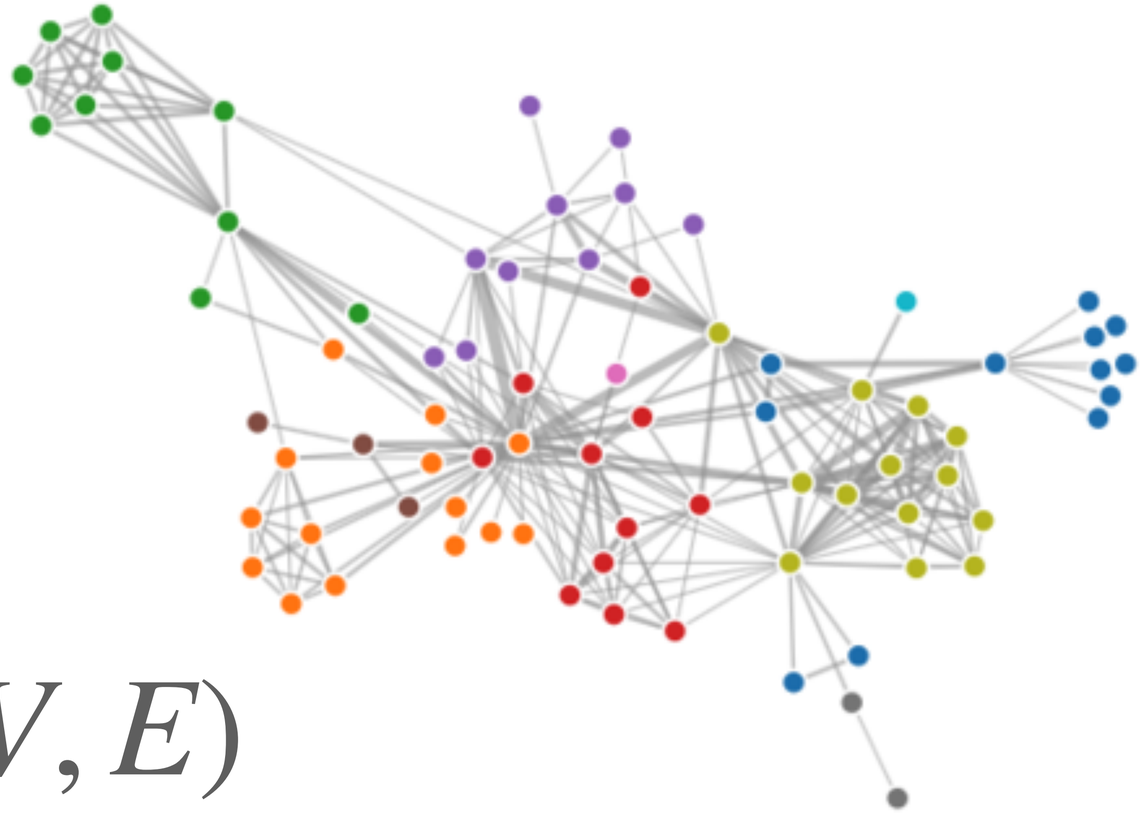
Fast and Accurate approximation of $f(X)$

Fix the weights w → $\tilde{f}_w(X)$ is **differentiable** with respect to X

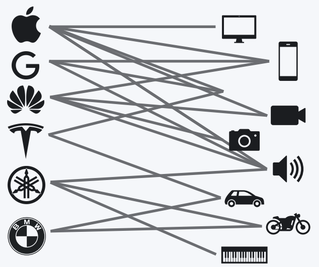


Graph Neural Networks

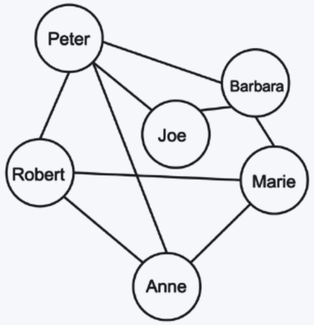
Learning on Graphs



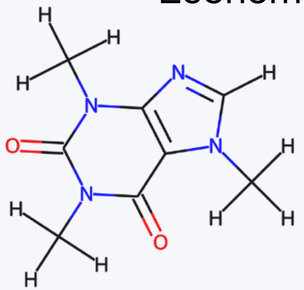
$$\mathcal{G}(V, E)$$



Economics

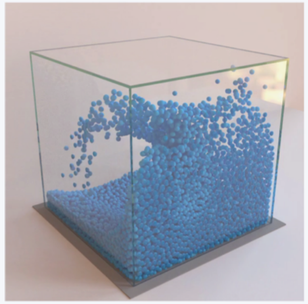


Social Networks



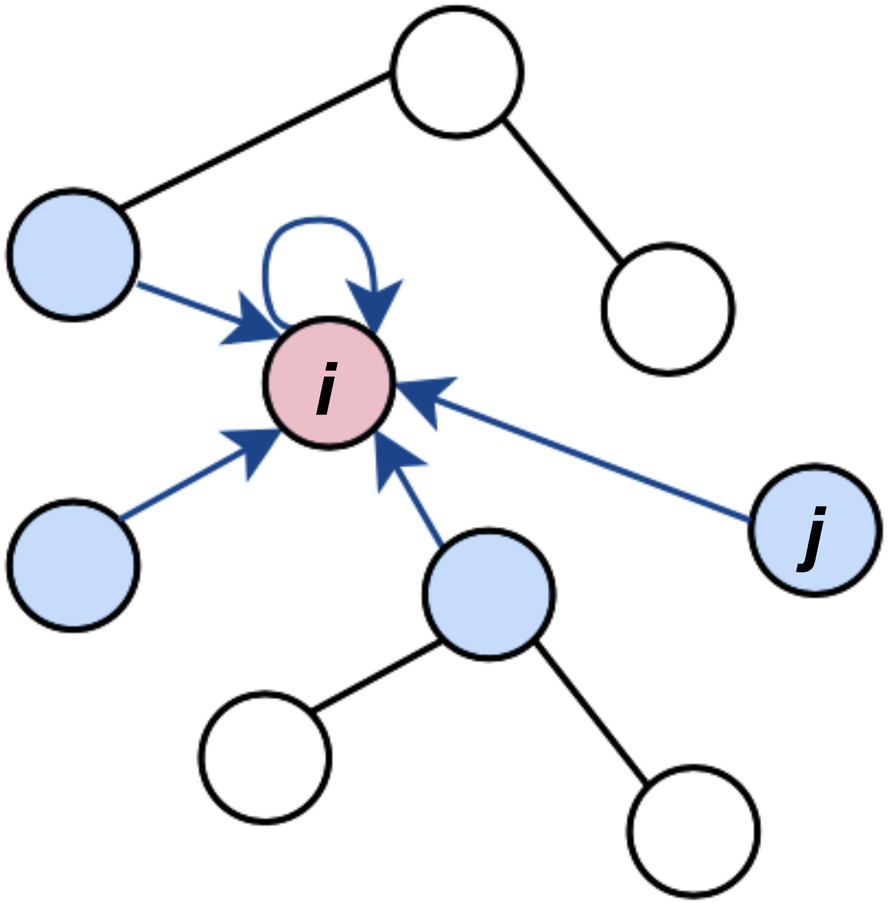
Molecules

Point Clouds



Physics Simulations

Message passing



Neighborhood information

Process

Learning on **nodes**

Aggregate

Learning on **edges**

QMD

Quantum extension of the classical molecular dynamics

Nucleons ⟶ Gaussian wave packet

$$\varphi_i(\mathbf{r}) = \frac{1}{(2\pi L)^{3/4}} \exp\left(-\frac{(\mathbf{r}-\mathbf{r}_i)^2}{4L} + \frac{i}{\hbar}\mathbf{r}\cdot\mathbf{p}_i\right)$$

Covariant Hamiltonian

$$H = \sum_i \sqrt{\mathbf{p}_i^2 + m_i^2 + 2m_i V_i}$$

Equation of motion

$$\dot{\mathbf{r}}_i = \frac{\partial H}{\partial \mathbf{p}_i}, \qquad \dot{\mathbf{p}}_i = -\frac{\partial H}{\partial \mathbf{r}_i}$$

QMD Potential

QMD

LiQMD

$$V_i = \frac{A}{2\rho_0}\sum_{j\neq i}\rho_{ij} + \frac{B}{\gamma+1}\frac{1}{\rho_0^{\gamma}}\left(\sum_{j\neq i}\rho_{ij}\right)^{\gamma}$$

$$+ \frac{g_0}{2\rho_0}\sum_{j\neq i} f_{sij}\,\rho_{ij} + \frac{g_\tau}{\rho_0^{\eta}}\left(\sum_{j\neq i}\rho_{ij}\right)^{\eta}$$

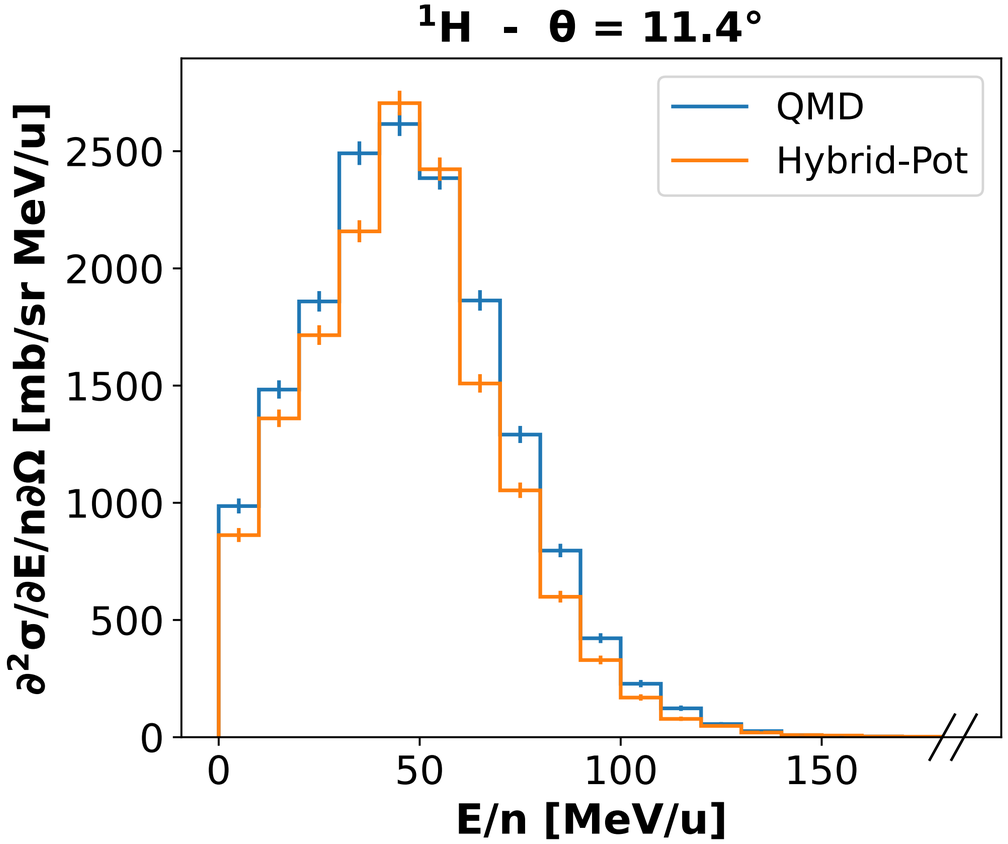
$$+ \frac{C_s}{2\rho_0}\sum_{j\neq i}\tau_{3,i}\tau_{3,j}\rho_{ij}(1-\kappa_s f_{sij})$$

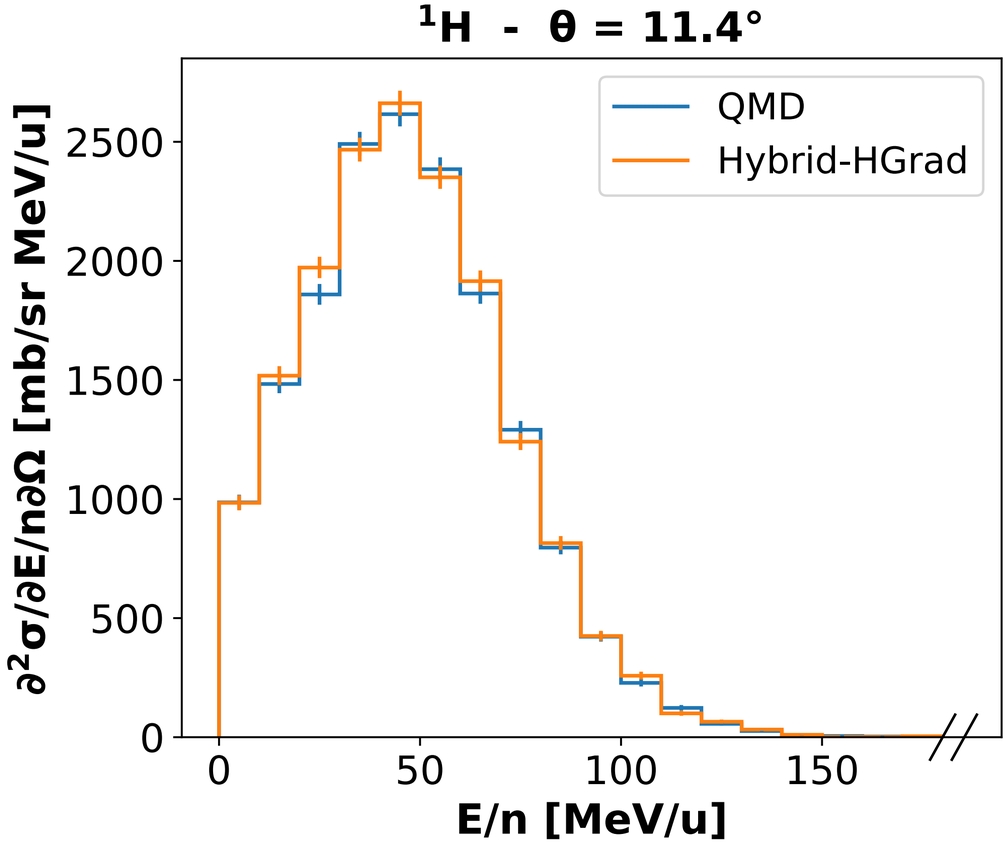
$$+ \frac{\alpha\hbar c}{2}\sum_{j\neq i}\frac{c_i c_j}{|\boldsymbol{r}_i-\boldsymbol{r}_j|}\,\mathrm{erf}\left(\frac{|\boldsymbol{r}_i-\boldsymbol{r}_j|}{\sqrt{4L}}\right)$$

Double differential cross sections

Lighter fragments

^{12}C on ^{nat}C at 62 MeV/u





Double differential cross sections

Heavier fragments

^{12}C on ^{nat}C at 62 MeV/u