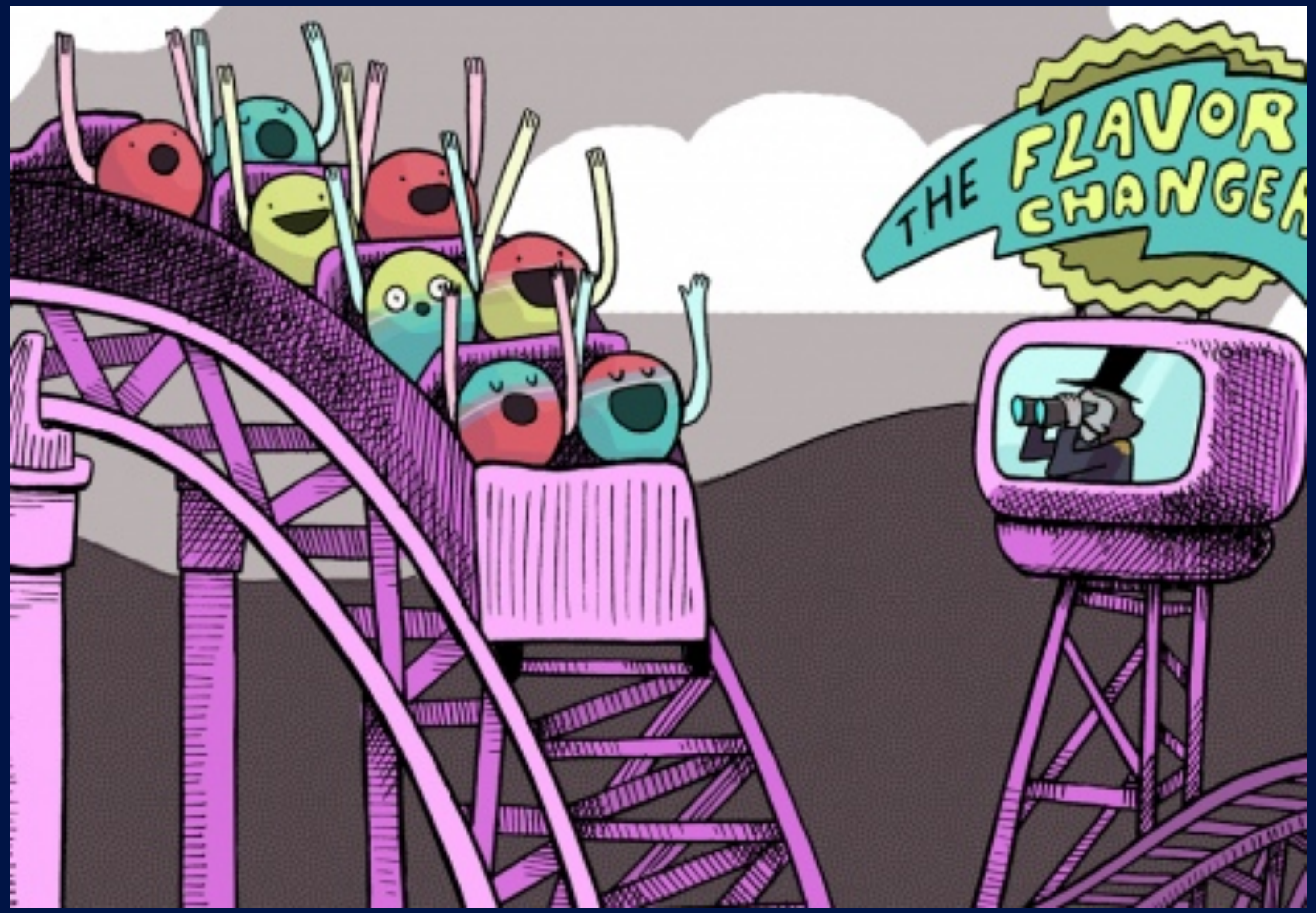
Machine Learning for Neutrino Cross Section Modeling

Karla Tame, S. Li, J. Isaacson, M. Wagman



Imperial College London

UNIVERSITY OF NOTRE DAME

Fermilab

Seattle

Nov-2-23

Outline

- **Motivation:** Needs and challenges for accurately modeling Neutrino physics cross section

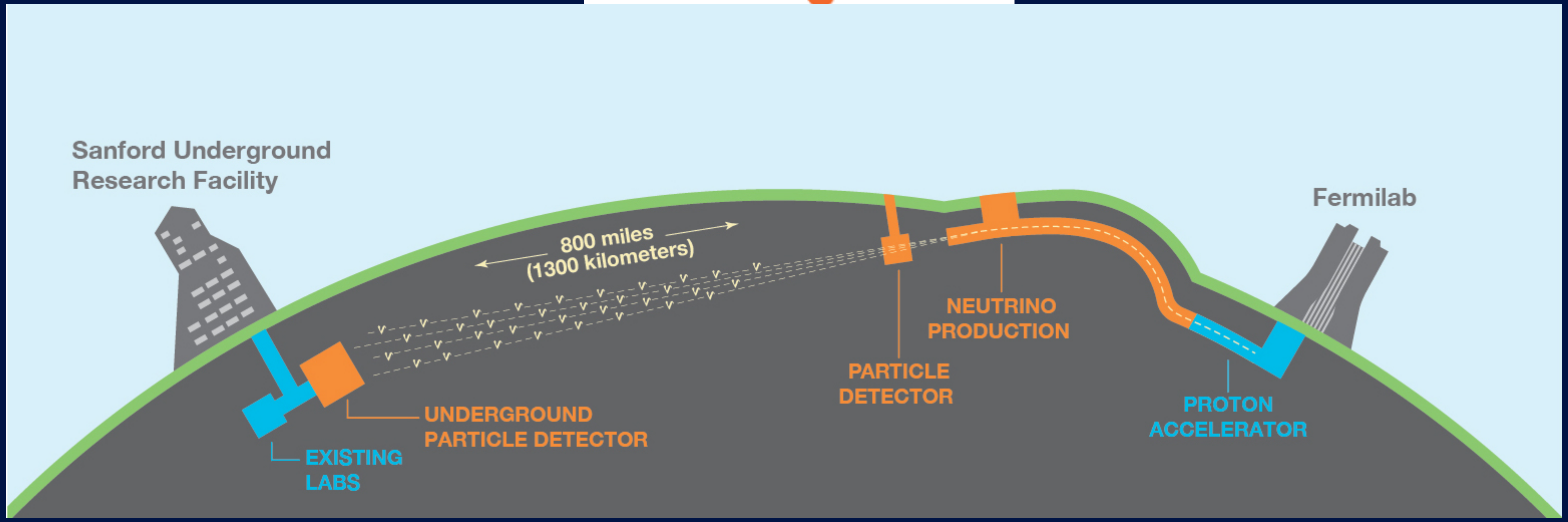
- **Machine Learning** Approach to model cross section

  Witchcraft or Math, unclear boundary?

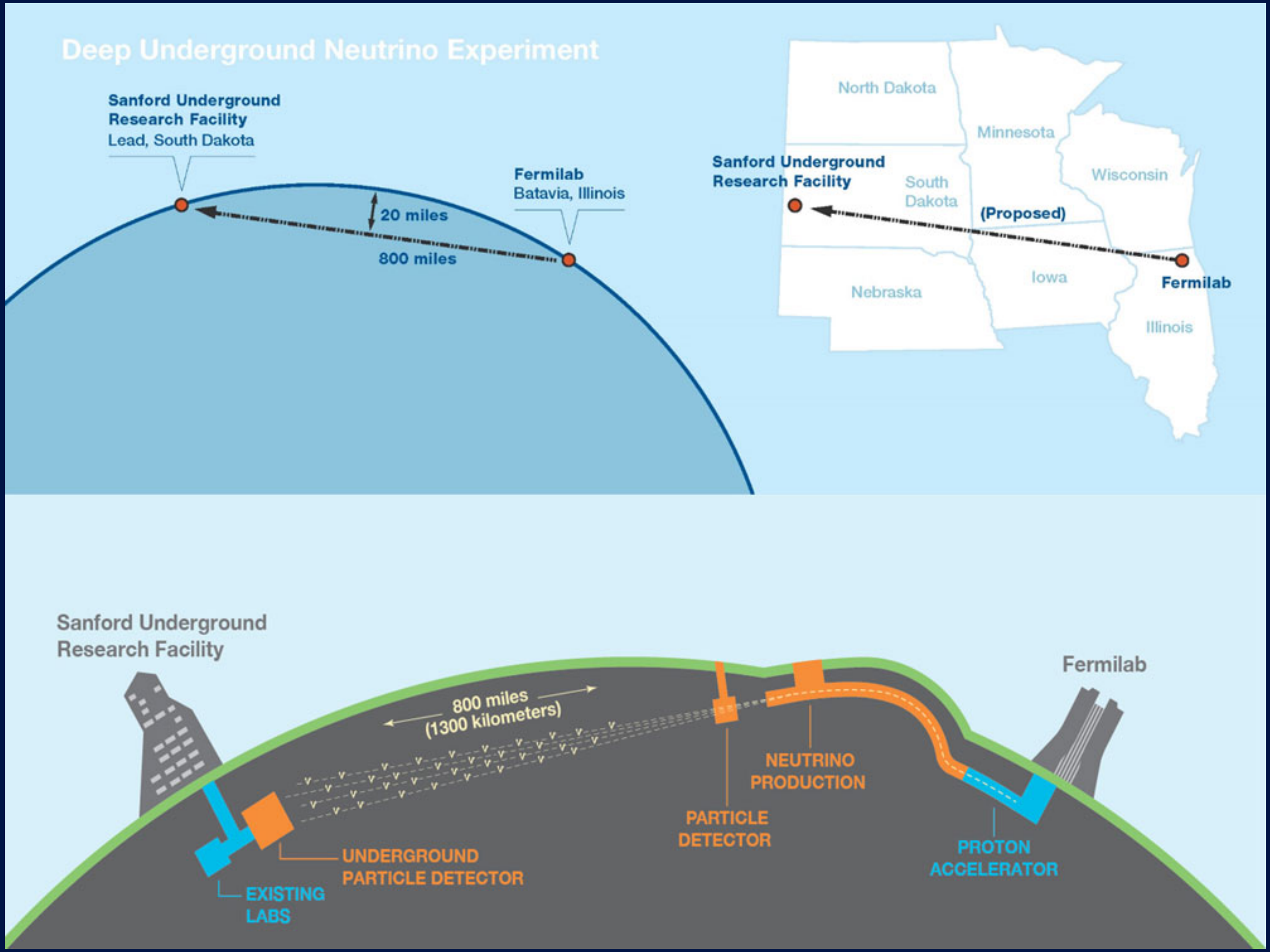
- **Results:** Preliminary results and future plans

Why do we Need Accurate Neutrino Cross-Section Modeling?

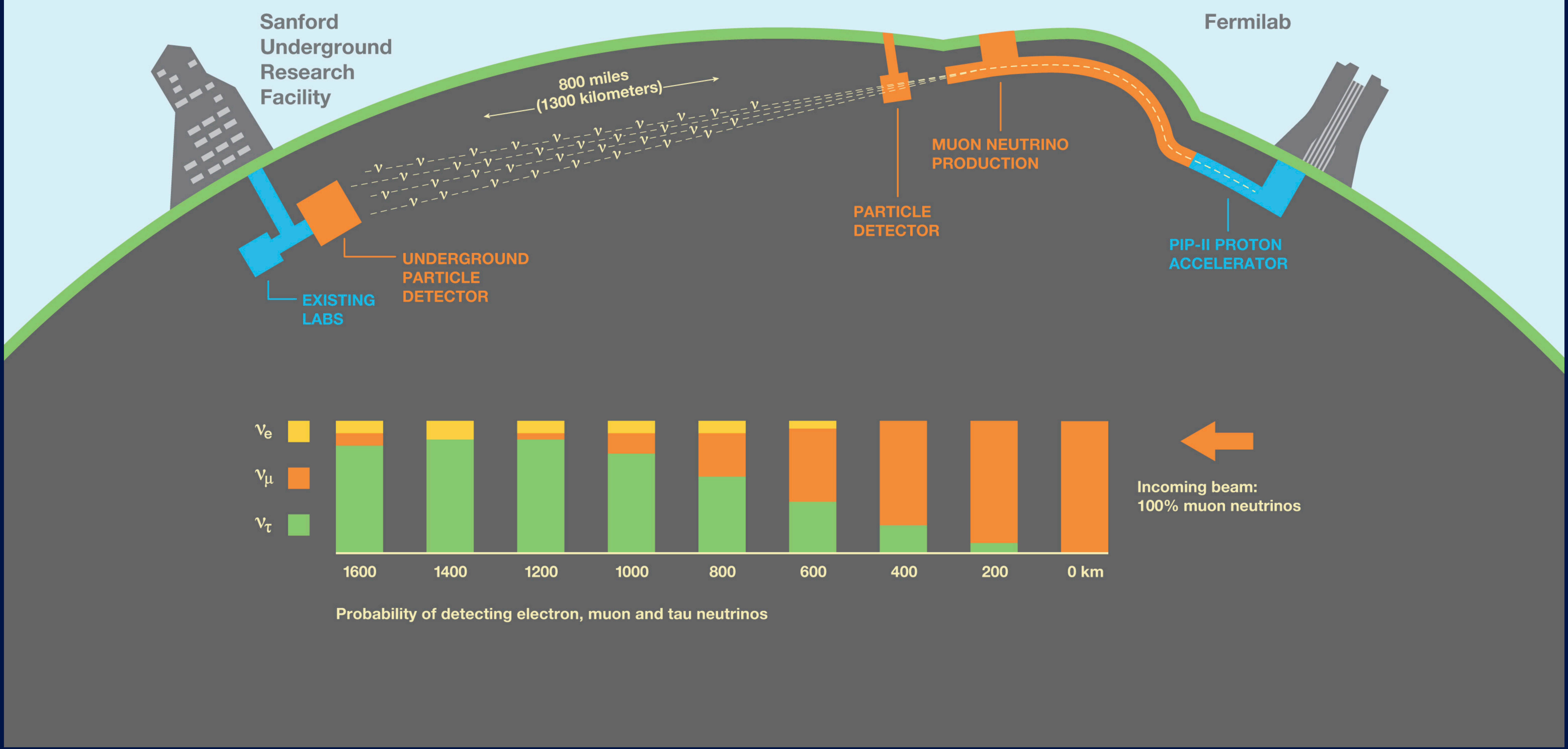
Why do we Need Accurate Neutrino Cross-Section Modeling?



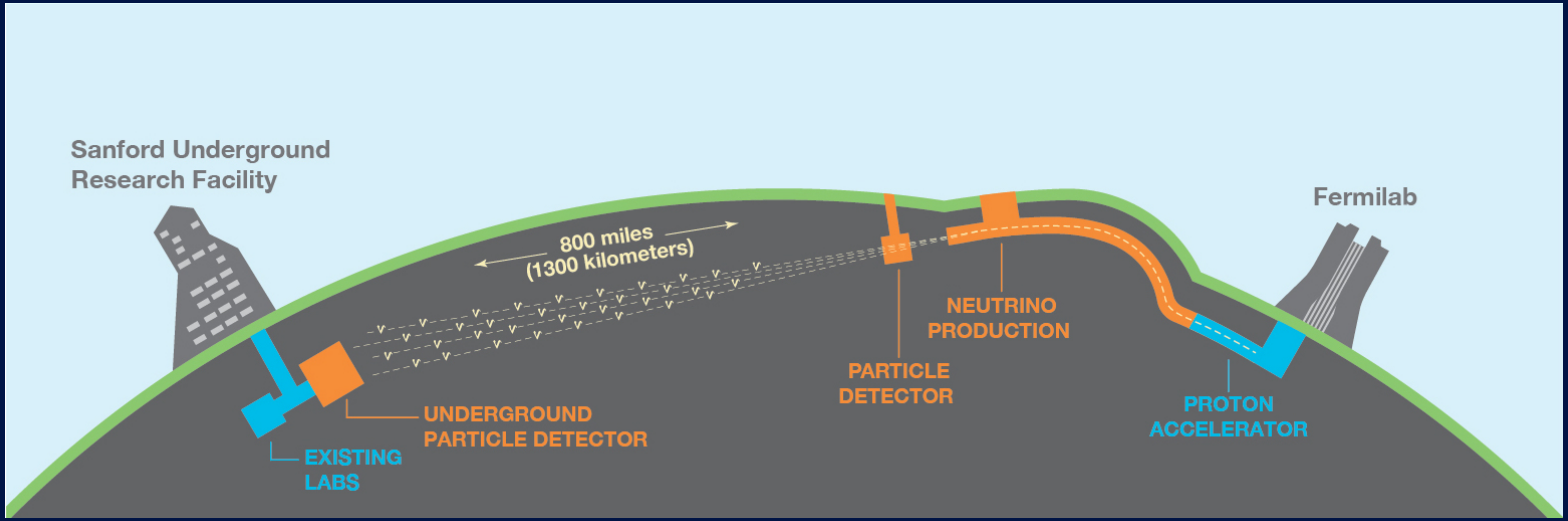
The next-generation experiments, such as DUNE and Hyper-Kamiokande, will achieve an unprecedented level of accuracy in measuring neutrino oscillation parameters.


Deep Underground Neutrino Experiment
Sanford Underground
Research Facility
Lead, South Dakota
Fermilab
Batavia, Illinois
20 miles
800 miles
North Dakota
Minnesota
Wisconsin
South
Dakota
Sanford Underground
Research Facility
(Proposed)
Nebraska
Iowa
Fermilab
Illinois
Sanford Underground
Research Facility
800 miles
(1300 kilometers)
Fermilab
NEUTRINO
PRODUCTION
PARTICLE
DETECTOR
PROTON
ACCELERATOR
UNDERGROUND
PARTICLE DETECTOR
EXISTING
LABS

Deep Underground Neutrino Experiment

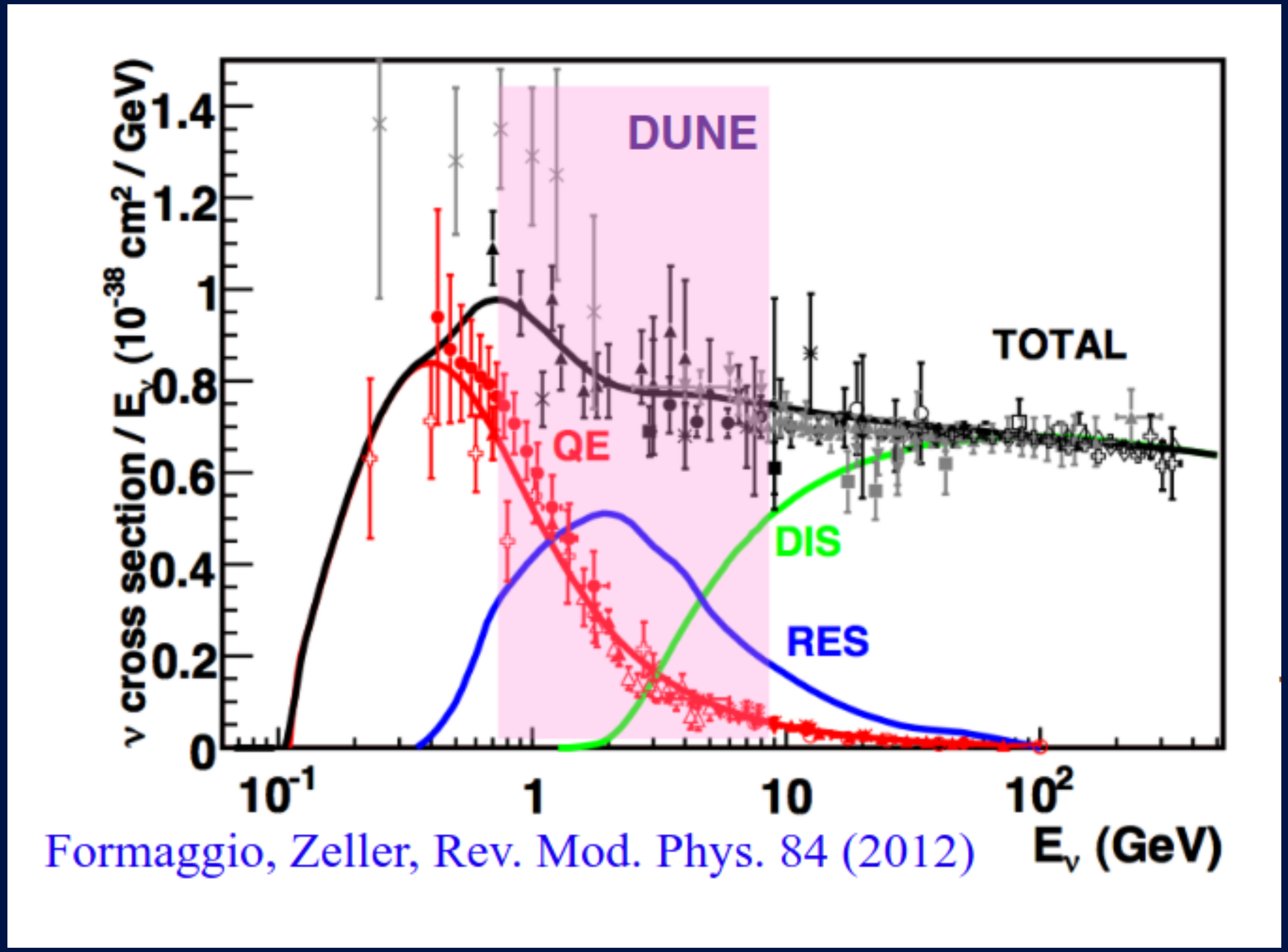


Neutrino experiments have detected compelling evidence of oscillations through the examination of fluxes from both near and far detectors.



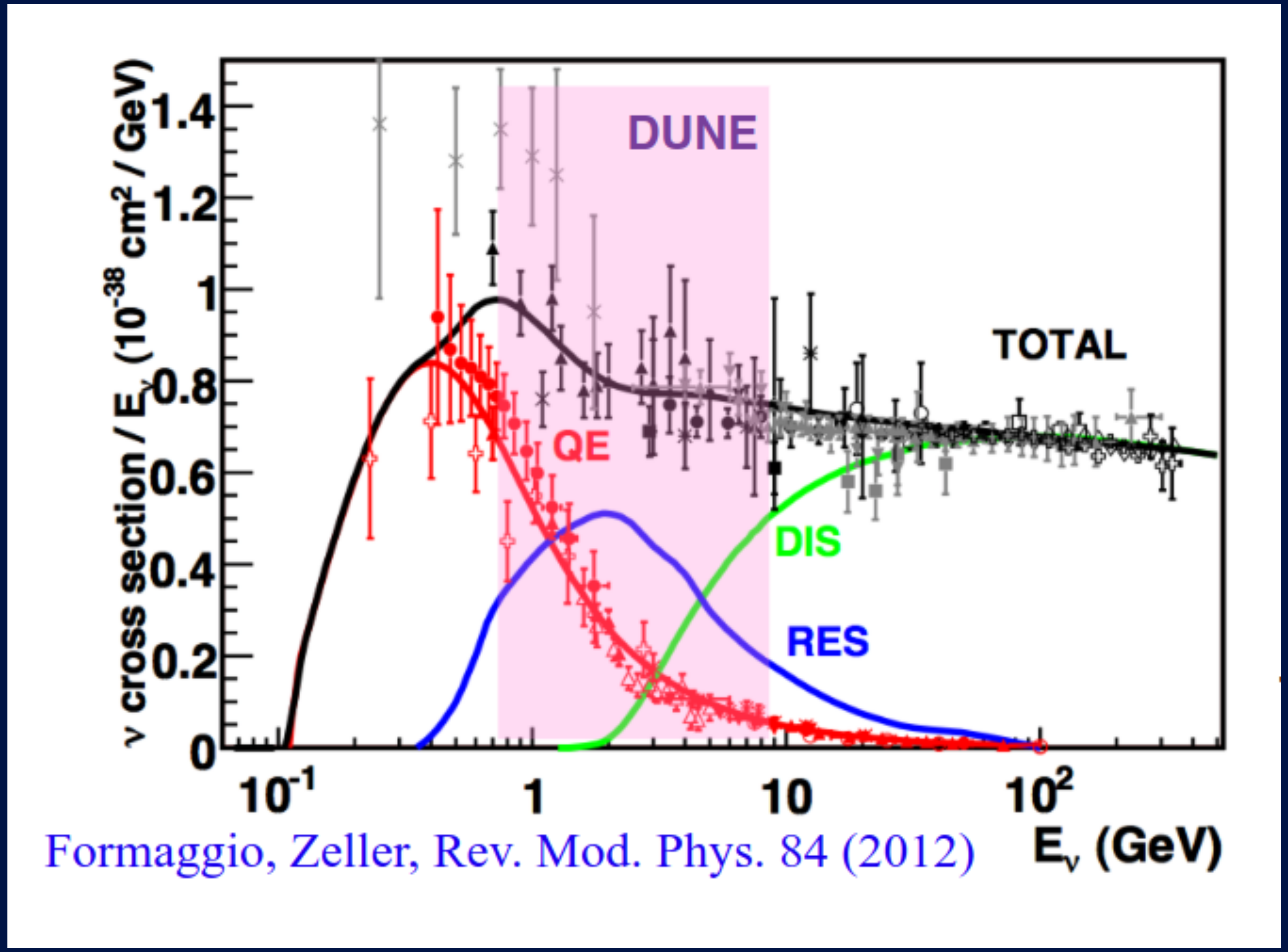
- To establish a connection between the detector fluxes, it will be crucial to have precise predictions for the cross-section.
- The errors in cross-sections directly lead to errors in oscillation probabilities, potentially resulting in distorted signatures of new physics.

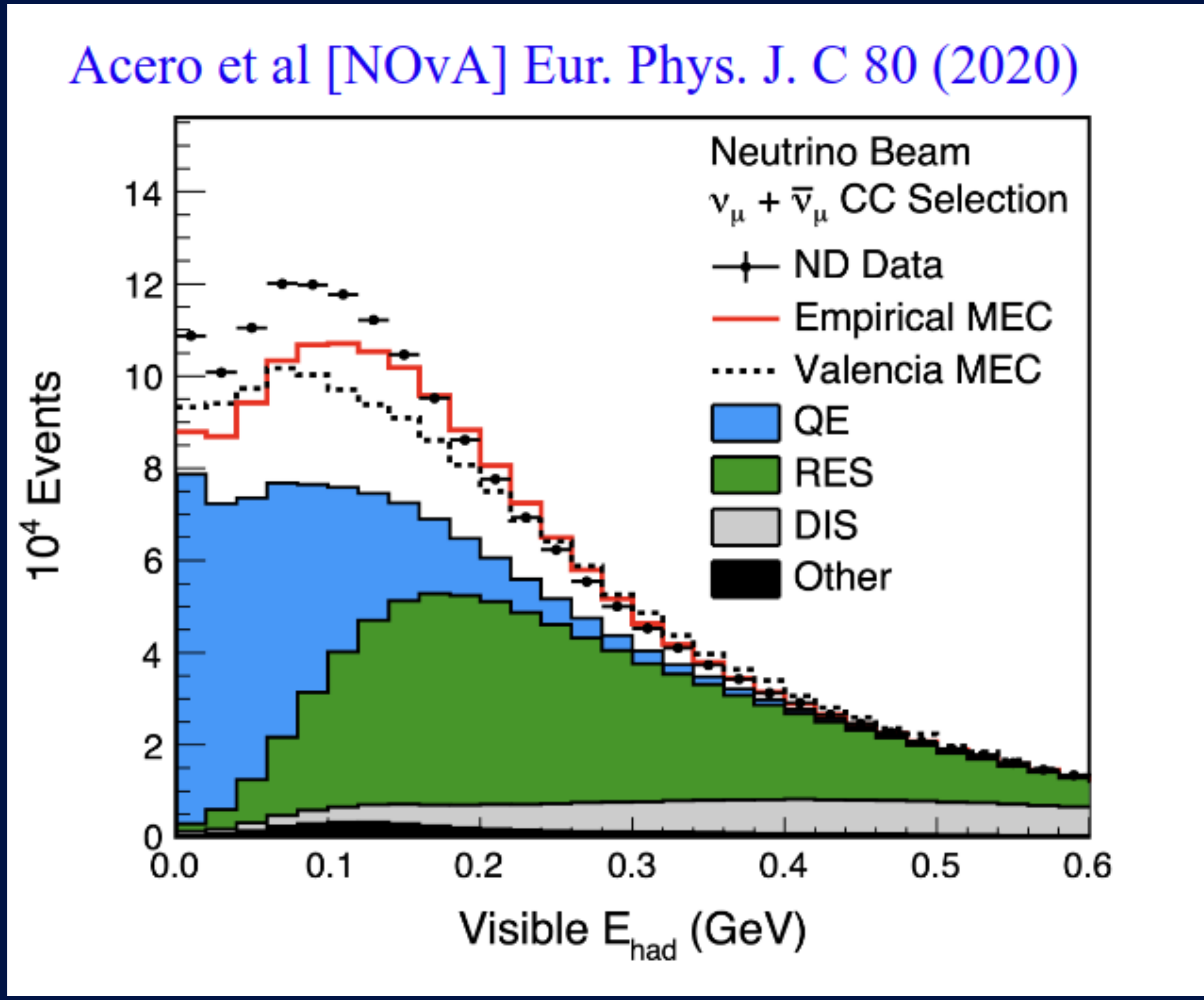
Challenges we currently face



Each energy region is dominated by certain production mechanism which is important for theory calculations

Challenges we currently face



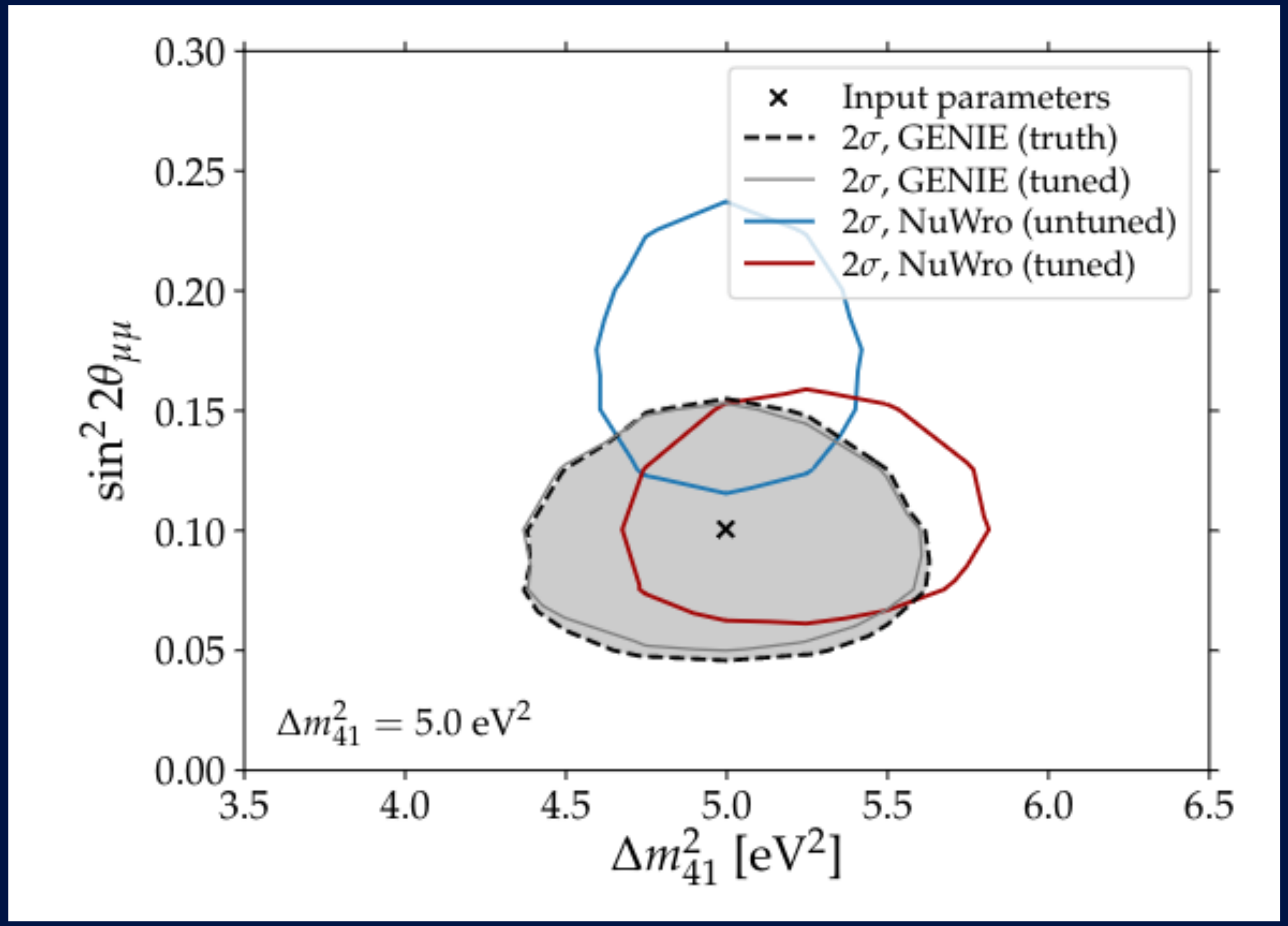


Each energy region is dominated by certain production mechanism which is important for theory calculations

There are discrepancies between generators and data.

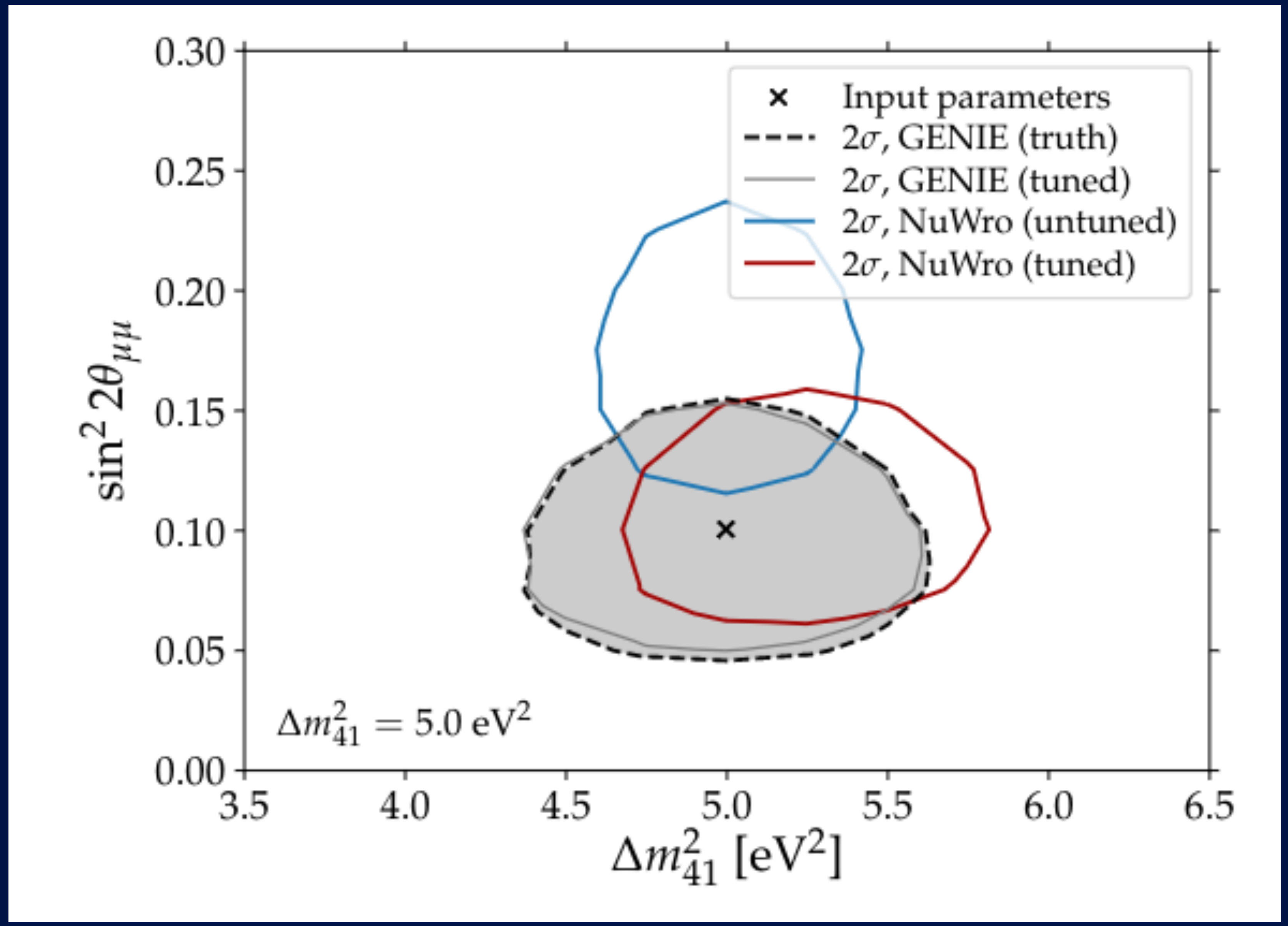
Challenges we currently face

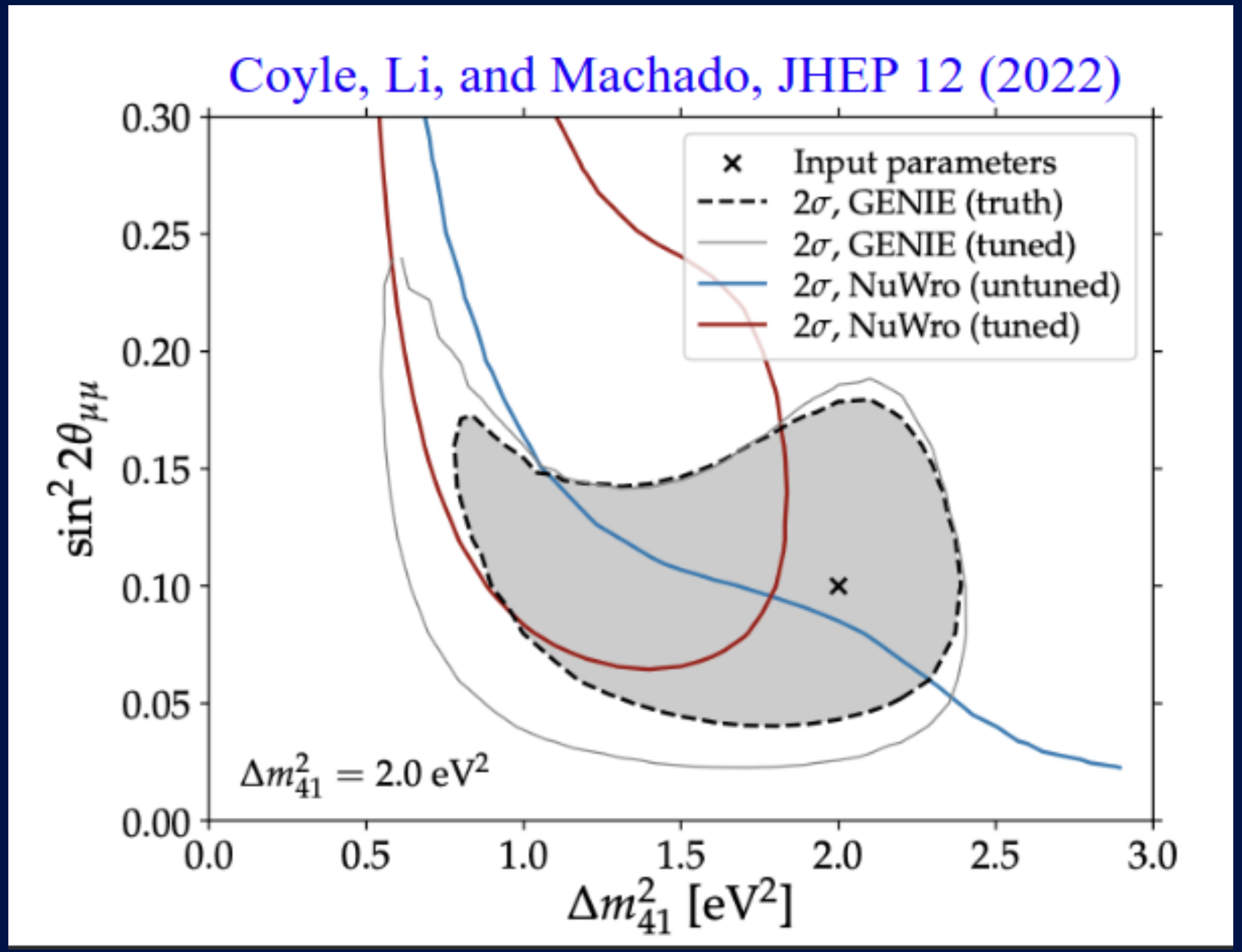
These differences are commonly addressed through tuning.



Challenges we currently face

Yet, tuning for one process does not guarantee accurate predictions for other processes or energy levels.





In other words, tuning the near detector alone may not always be adequate for accurately extracting new physics signals, primarily due to assumptions made in cross-section models.

How are we trying to address this problems?

- No ad hoc theory assumptions
- Use Fundamental theory plus parametrization of our ignorance
- Pure data driven approach (Machine Learning)

$$\frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta)}{dE' d\cos\theta} > 0$$ → Probability distribution

- Normalizing Flows

Part I: No Fundamental Theory Assumptions

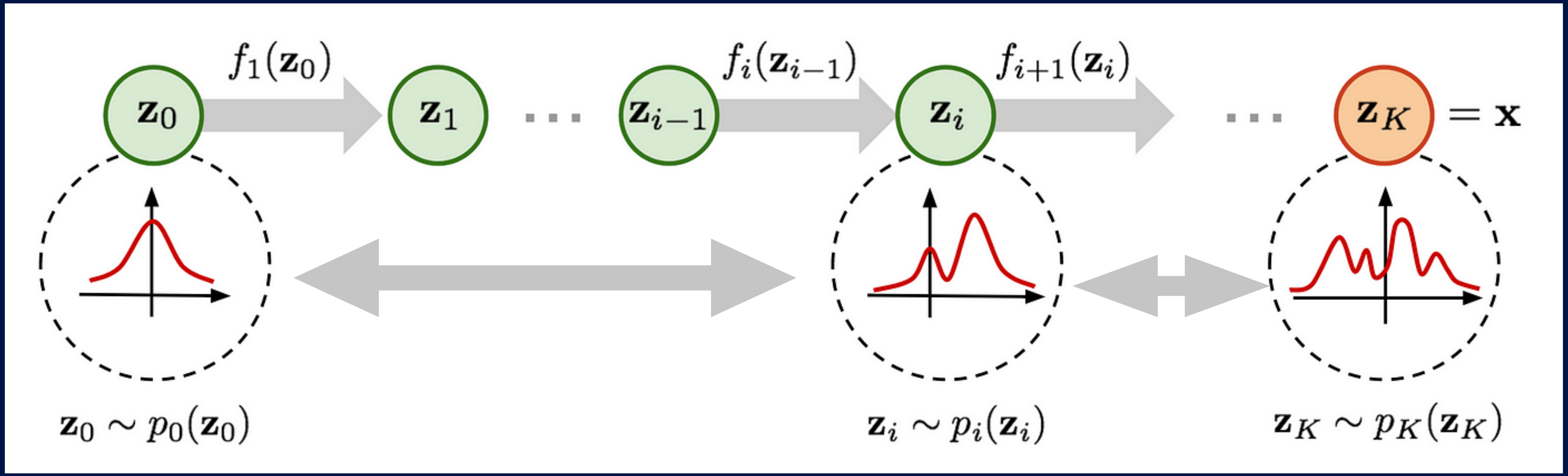
Part II: Some FTA, in particular Structure Functions

12

Part I: No Fundamental Theory Assumptions

Normalizing Flows in a Nutshell

The core idea of normalizing flows is to apply a series of invertible transformations to the samples drawn from the base (simple) distribution. These transformations are typically parameterized by neural networks.



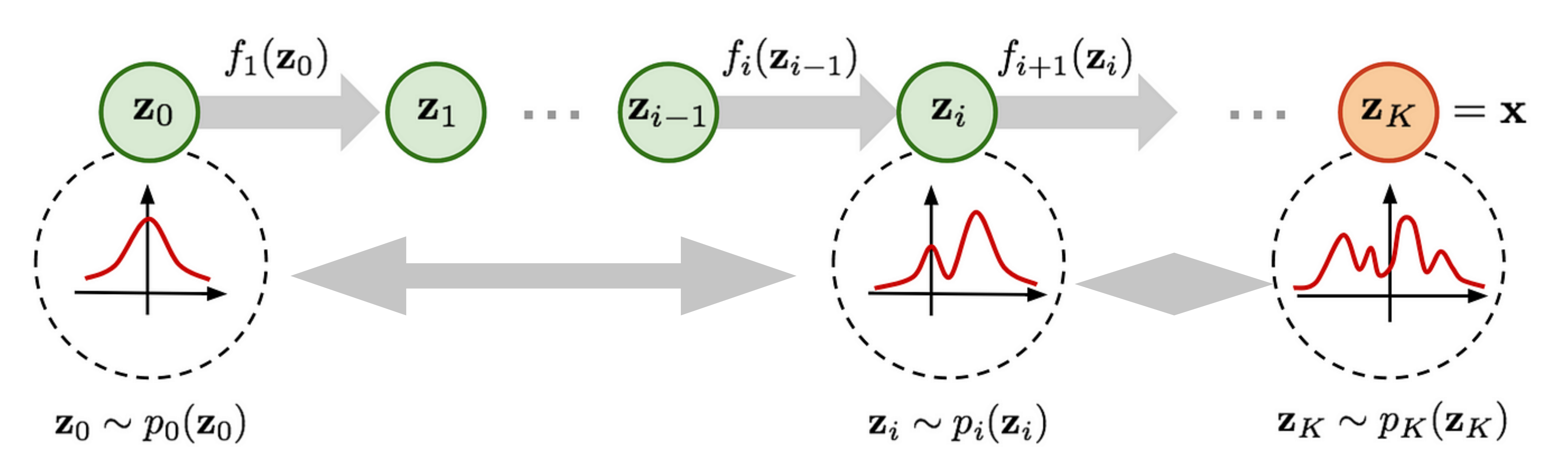
Flow-based Deep Generative Models

Normalizing Flows in a Nutshell

- Normalizing Flows involves a series of **reversible** steps, enabling retrieval of the original distribution.

- Several transformations are applied sequentially to progressively approximate the desired distribution.

- The transformation parameters are learned **from data**, usually parametrized by Neural Networks, allowing adaptation to specific (complicated) distributions.



Flow-based Deep Generative Models

Change of Variables Formula in Normalizing Flows context

$$X = f(Z) \text{ and } Z = f^{-1}(X)$$

$$P_X(X) = P_Z(f^{-1}(X)) \left| det \frac{\partial f^{-1}(X)}{\partial_X} \right|$$

Change of Variables Formula in Normalizing Flows context

$$X = f(Z) \text{ and } Z = f^{-1}(X)$$

$$P_X(X) = P_Z(f^{-1}(X)) \left| det \frac{\partial f^{-1}(X)}{\partial x} \right|$$

base distribution
(uniform, gaussian)

Change of Variables Formula in Normalizing Flows context

$$X = f(Z) \text{ and } Z = f^{-1}(X)$$

$$P_X(X) = P_Z(f^{-1}(X)) \left| det \frac{\partial f^{-1}(X)}{\partial X} \right|$$

base distribution
(uniform, gaussian)

target distribution
exp.data

Change of Variables Formula in Normalizing Flows context

$$X = f(Z) \ \text{and} \ \ Z = f^{-1}(X)$$

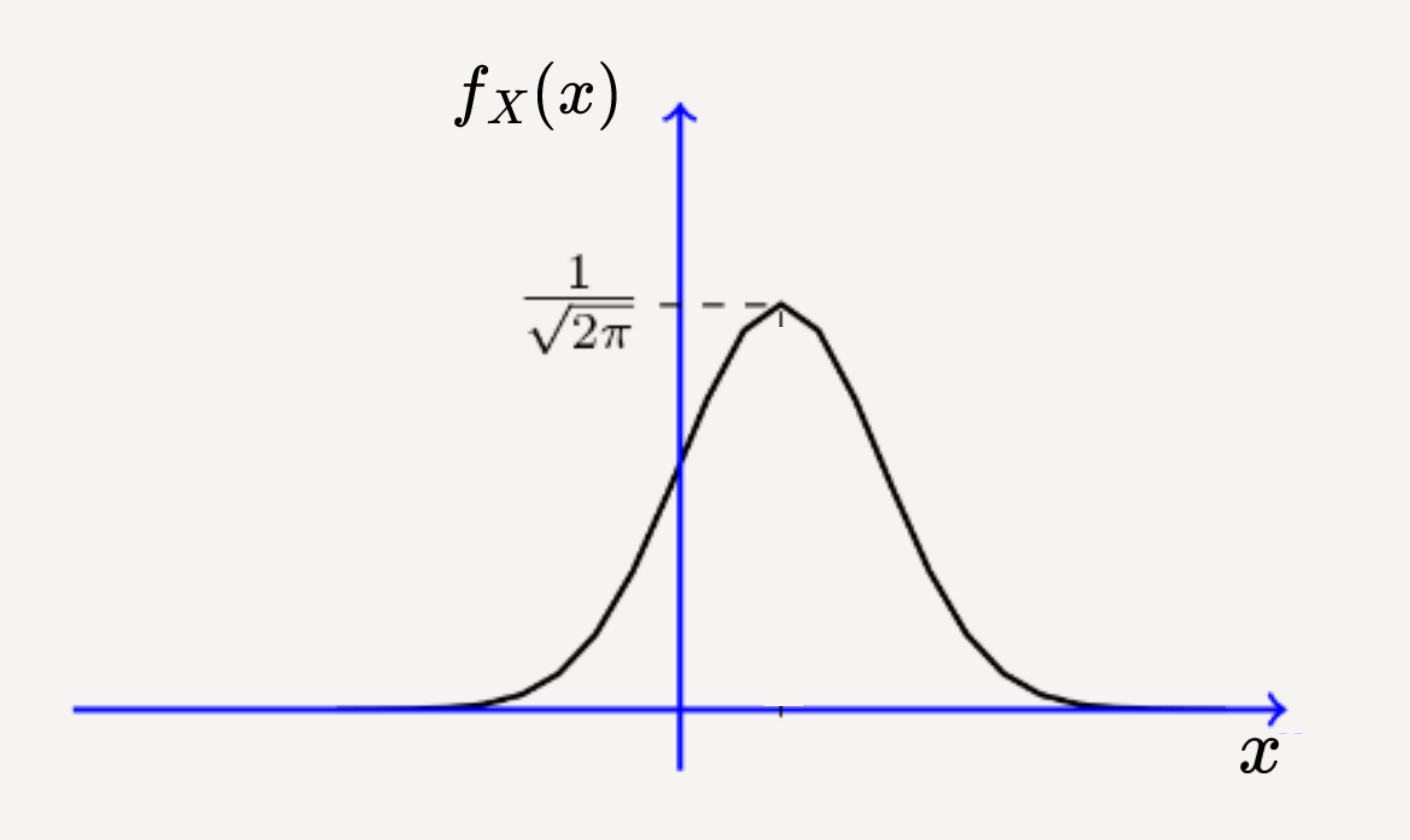
$$P_X(X) = P_Z(f^{-1}(X)) \left| det \frac{\partial f^{-1}(X)}{\partial_X} \right|$$

Jacobian matrix
parameterized with NN

base distribution
(uniform, gaussian)

target distribution
exp.data

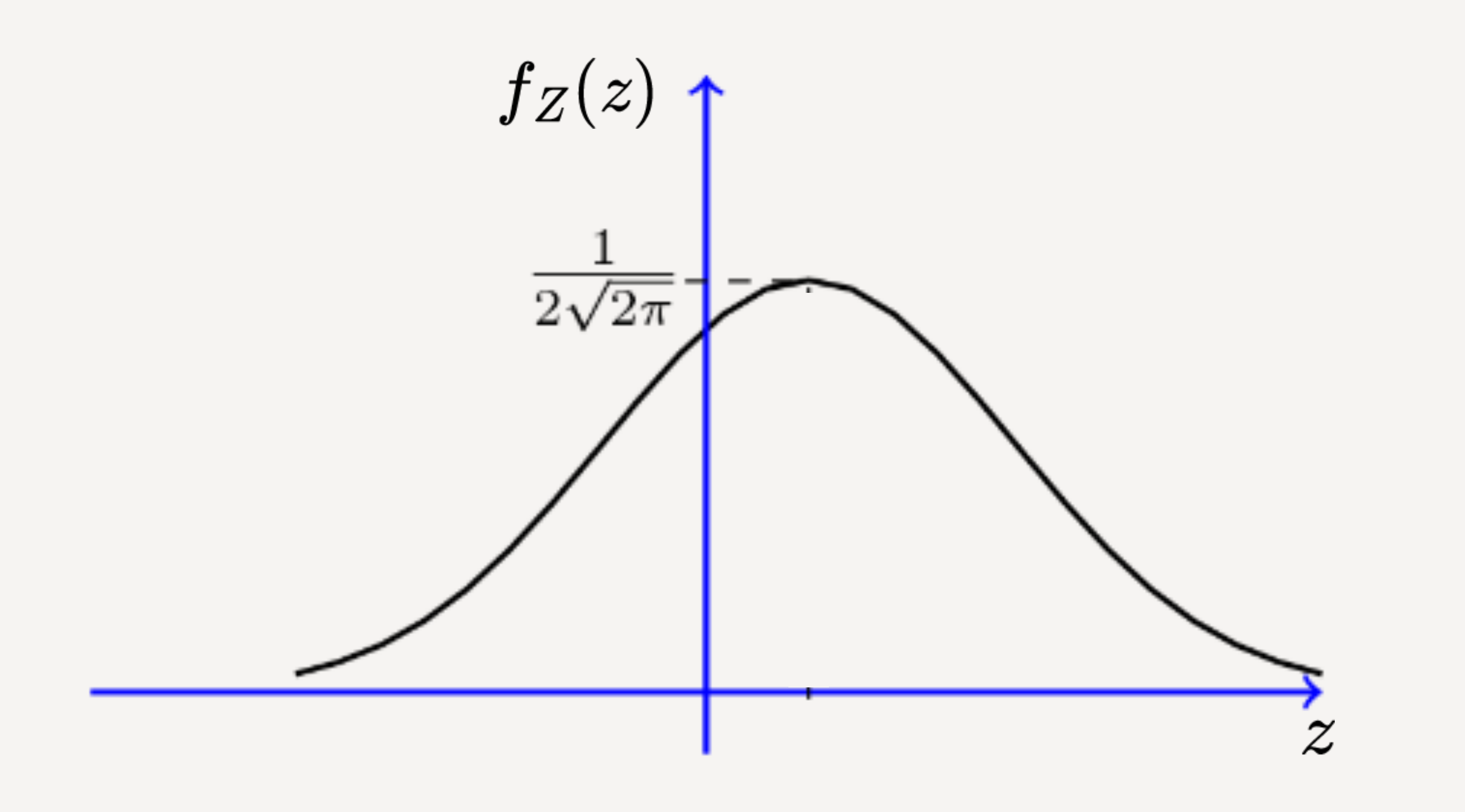
Normalizing Flows in a Nutshell



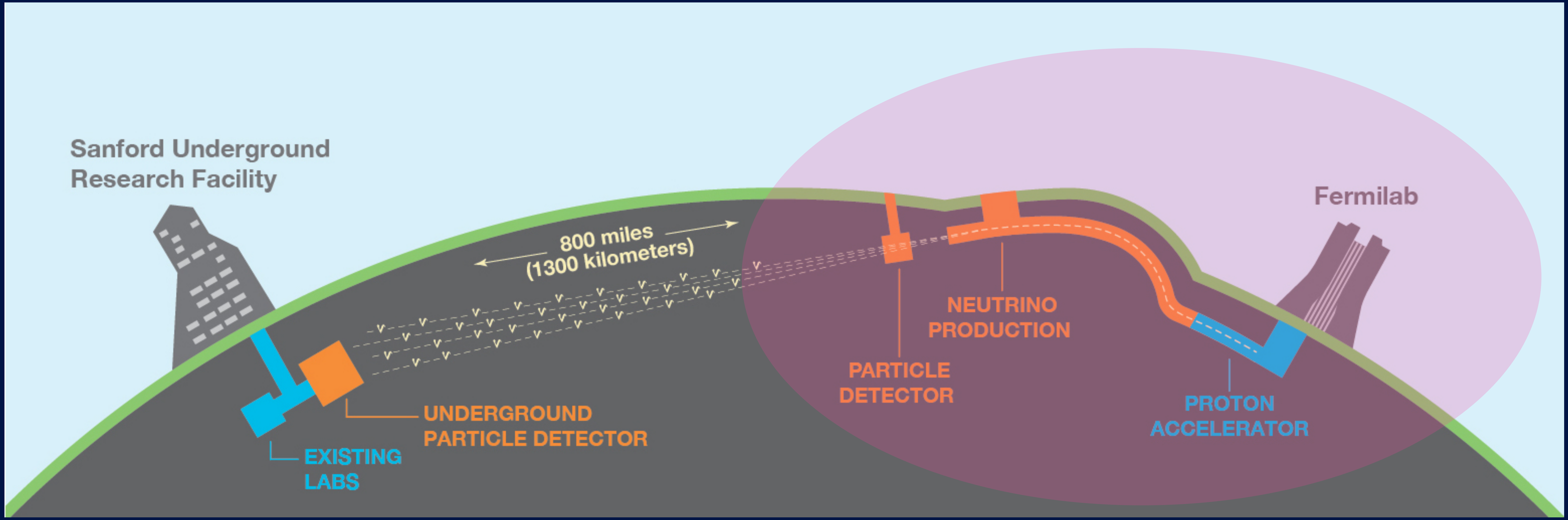
$$z = 2x$$

$$P(x) = P(z)\frac{dz}{dx}$$

$$\frac{dz}{dx} = 2$$



Near Detector



$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3) \approx \frac{1}{\mathcal{N}_{flow}} \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3) \approx \frac{1}{\mathcal{N}_{flow}} \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

Probability distribution
(target)

$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3) \approx \frac{1}{N_{flow}} \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

Probability distribution (target)

Jacobians

$$J(E_\nu, E', \cos\theta; \lambda) = \left| \frac{\partial(E_\nu, E', \cos\theta)}{\partial(z^1, z^2, z^3)} \right|$$

Trainable network parameters

$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3) \approx \frac{1}{\mathcal{N}_{flow}} \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

Probability distribution (target)

Jacobians

$$J(E_\nu, E', \cos\theta; \lambda) = \left| \frac{\partial(E_\nu, E', \cos\theta)}{\partial(z^1, z^2, z^3)} \right|$$

Trainable network parameters

Base distribution (simple)

$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3) \approx \frac{1}{\mathcal{N}_{flow}} \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

Probability distribution (target)

Jacobians

$$J(E_\nu, E', \cos\theta; \lambda) = \left| \frac{\partial(E_\nu, E', \cos\theta)}{\partial(z^1, z^2, z^3)} \right|$$

Trainable network parameters

Base distribution (simple)

$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3) \approx \frac{1}{N_{flow}} \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

Probability distribution (target)

Jacobians

expressively parameterized functions like NNets

$$J(E_\nu, E', \cos\theta; \lambda) = \left| \frac{\partial(E_\nu, E', \cos\theta)}{\partial(z^1, z^2, z^3)} \right|$$

Near Detector

$$Q_{ND}(E', \cos\theta) = \frac{1}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta)}{dE' d\cos\theta}$$

Near Detector

$$Q_{ND}(E', \cos\theta) = \frac{1}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta)}{dE' d\cos\theta}$$

2D distribution
(GENIE Data)

Near Detector

$$Q_{ND}(E', \cos\theta) = \frac{1}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta)}{dE' d\cos\theta}$$

2D distribution (GENIE Data)

Normalization factor

Near Detector

Near detector Neutrino-Flux
(GENIE Data)

$$Q_{ND}(E', \cos\theta) = \frac{1}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta)}{dE' d\cos\theta}$$

2D distribution
(GENIE Data)

Normalization
factor

Near Detector

$$Q_{ND}(E', \cos\theta) = \frac{1}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta)}{dE' d\cos\theta}$$

2D distribution (GENIE Data)

Normalization factor

Near detector Neutrino-Flux (GENIE Data)

Cross-section

Near Detector

$$Q_{ND}(E', \cos\theta) = \frac{1}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta)}{dE' d\cos\theta}$$

2D distribution (GENIE Data)

Normalization factor

Near detector Neutrino-Flux (GENIE Data)

Cross-section

We aim to estimate the unknown 3D probability distribution, which is proportional to the cross section, by employing normalizing flows.

Note that the normalization factors between the 2D and 3D distributions differ as follows

Note that the normalization factors between the 2D and 3D distributions differ as follows

$$\mathcal{N}_{ND} = \int dE' d\cos\theta dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

Normalization factor for near detector depending on neutrino flux

Note that the normalization factors between the 2D and 3D distributions differ as follows

$$\mathcal{N}_{ND} = \int dE' d\cos\theta dE_\nu \Phi_{ND}(E_\nu) \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

→ Normalization factor for near detector depending on neutrino flux

$$\mathcal{N}_{flow} = \int dE_\nu dE' d\cos\theta \frac{d^2\sigma^{(\ell A)}}{dE' d\cos\theta}$$

→ Normalization factor for estimated probability without neutrino flux dependence

We then marginalize the probability distribution over the neutrino energy weighted by neutrino flux

$$\widetilde{P}(E', \cos\theta; \lambda) = \frac{\mathcal{N}_{flow}}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda) \approx Q_{ND}(E', \cos\theta)$$

We then marginalize the probability distribution over the neutrino energy weighted by neutrino flux

$$\widetilde{P}(E', \cos\theta; \lambda) = \frac{\mathcal{N}_{flow}}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda) \approx Q_{ND}(E', \cos\theta)$$

Where the normalization factor ratio is

$$\frac{\mathcal{N}_{ND}}{\mathcal{N}_{flow}} = \int dE_\nu dE' d\cos\theta \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)$$

Where the normalization factor ratio is

$$\frac{\mathcal{N}_{ND}}{\mathcal{N}_{flow}} = \int dE_\nu dE' d\cos\theta \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)$$

To compute the integral we can sample over a flat distribution

$$F(E', \cos\theta) = \frac{1}{\mathcal{N}_F}$$

where

$$\mathcal{N}_F = \int_0^{E'_{\max}} \int_{-1}^{1} d\cos\theta = 2E'_{\max}$$

Therefore

$$\frac{\mathcal{N}_{ND}}{\mathcal{N}_{flow}} = \left\langle \langle P(E_\nu, E', \cos\theta; \lambda) \rangle_{\Phi_{ND}} \right\rangle_F$$

We have everything to compute the Loss Function to train our ND Network

We have everything to compute the Loss Function to train our ND Network

Loss Function: KL-Divergence

Metric function encoding distance between probability distributions

$$D(p_1||p_2) = \int dx\, p_1(x) \ln \left[\frac{p_1(x)}{p_2(x)} \right]$$

$$KL = 0 \ < \ \text{if } \ p_1 = p_2$$

$$KL > 0 \ \text{if } \ p_1 \neq p_2$$

Loss Function: KL-Divergence

$$D(Q_{ND}||\widetilde{P}) = \int dE' d\cos\theta \, Q_{ND}(E', \cos\theta) \ln \left[\frac{Q_{ND}(E', \cos\theta)}{\widetilde{P}(E', \cos\theta; \lambda)} \right]$$

Loss Function: KL-Divergence

$$D(Q_{ND}||\widetilde{P}) = \int dE' d\cos\theta\, Q_{ND}(E', \cos\theta) \ln\left[\frac{Q_{ND}(E', \cos\theta)}{\widetilde{P}(E', \cos\theta; \lambda)}\right]$$

$$\equiv \mathcal{C}(E', \cos\theta) - \mathcal{L}(E', \cos\theta; \lambda)$$

Irrelevant constant term

Loss Function: KL-Divergence

$$D(Q_{ND}||\widetilde{P}) = \int dE' d\cos\theta\, Q_{ND}(E', \cos\theta) \ln\left[\frac{Q_{ND}(E', \cos\theta)}{\widetilde{P}(E', \cos\theta; \lambda)}\right]$$

$$\equiv \mathcal{C}(E', \cos\theta) - \mathcal{L}(E', \cos\theta; \lambda)$$

Irrelevant constant term

Loss function we minimize

Loss Function: KL-Divergence

$$D(Q_{ND}||\widetilde{P}) = \int dE' d\cos\theta \, Q_{ND}(E', \cos\theta) \ln \left[\frac{Q_{ND}(E', \cos\theta)}{\widetilde{P}(E', \cos\theta; \lambda)} \right]$$

$$\equiv \mathcal{C}(E', \cos\theta) - \mathcal{L}(E', \cos\theta; \lambda)$$

$$\mathcal{L} = -\int dE' d\cos\theta \, Q_{ND}(E', \cos\theta) \ln \left[\widetilde{P}(E', \cos\theta; \lambda) \right]$$

$$\equiv -\left\langle \ln \left[\widetilde{P}(E', \cos\theta; \lambda) \right] \right\rangle_{Q_{ND}}$$

Expanding the terms in the loss

$$\mathcal{L} = -\left\langle \ln \left[\frac{\int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)}{\left\langle \langle P(E_\nu, E', \cos\theta; \lambda) \rangle_{\Phi_{ND}} \right\rangle_F} \right] \right\rangle_{Q_{ND}}$$

$$\mathcal{L} = -\left\langle \ln \left[\widetilde{P}(E', \cos\theta; \lambda) \right] \right\rangle_{Q_{ND}}$$

$$\widetilde{P}(E', \cos\theta; \lambda) = \frac{\mathcal{N}_{flow}}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)$$

$$\frac{\mathcal{N}_{ND}}{\mathcal{N}_{flow}} = \left\langle \langle P(E_\nu, E', \cos\theta; \lambda) \rangle_{\Phi_{ND}} \right\rangle_F$$

Expanding the terms in the loss

$$\mathcal{L} = -\left\langle \ln\left[\widetilde{P}(E', \cos\theta; \lambda)\right]\right\rangle_{Q_{ND}}$$

$$\widetilde{P}(E', \cos\theta; \lambda) = \frac{\mathcal{N}_{flow}}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)$$

$$\frac{\mathcal{N}_{ND}}{\mathcal{N}_{flow}} = \left\langle \langle P(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}} \right\rangle_F$$

$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3)$$

$$\mathcal{L} = -\left\langle \ln\left[\frac{\int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)}{\left\langle \langle P(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}} \right\rangle_F}\right]\right\rangle_{Q_{ND}}$$

$$= -\left\langle \ln\left[\frac{\int dE_\nu \Phi_{ND}(E_\nu) J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3)}{\left\langle \langle J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3\rangle_{\Phi_{ND}} \right\rangle_F}\right]\right\rangle_{Q_{ND}}$$

Expanding the terms in the loss

$$\mathcal{L} = -\left\langle \ln\left[\widetilde{P}(E', \cos\theta; \lambda)\right]\right\rangle_{Q_{ND}}$$

$$\widetilde{P}(E', \cos\theta; \lambda) = \frac{\mathcal{N}_{flow}}{\mathcal{N}_{ND}} \int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)$$

$$\frac{\mathcal{N}_{ND}}{\mathcal{N}_{flow}} = \left\langle \langle P(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}}\right\rangle_F$$

$$P(E_\nu, E', \cos\theta; \lambda) = J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3)$$

$$\mathcal{L} = -\left\langle \ln\left[\frac{\int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda)}{\left\langle \langle P(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}}\right\rangle_F}\right]\right\rangle_{Q_{ND}}$$

$$= -\left\langle \ln\left[\frac{\int dE_\nu \Phi_{ND}(E_\nu) J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3)}{\left\langle \langle J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3\rangle_{\Phi_{ND}}\right\rangle_F}\right]\right\rangle_{Q_{ND}}$$

$$= -\left\langle \ln\left[\frac{\langle J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3)\rangle_{\Phi_{ND}}}{\left\langle \langle J(E_\nu, E', \cos\theta; \lambda) B(z^1, z^2, z^3\rangle_{\Phi_{ND}}\right\rangle_F}\right]\right\rangle_{Q_{ND}}$$

If we take $B(z^1, z^2, z^3)$ to be a uniform distribution it can be reabsorbed in the overall constant

$$\mathcal{L} = -\left\langle \ln\left[\frac{\langle J(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}}}{\left\langle\langle J(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}}\right\rangle_F}\right]\right\rangle_{Q_{ND}}$$

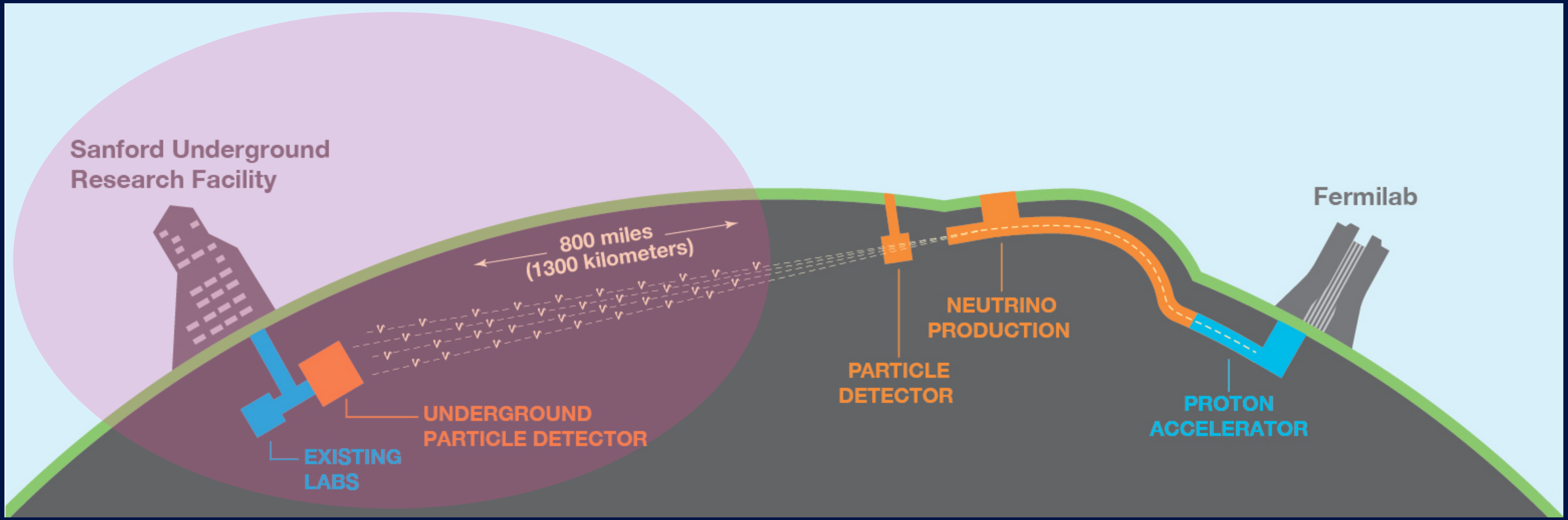
If we take $B(z^1, z^2, z^3)$ to be a uniform distribution it can be reabsorbed in the overall constant

$$\mathcal{L} = -\left\langle \ln\left[\frac{\langle J(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}}}{\langle\langle J(E_\nu, E', \cos\theta; \lambda)\rangle_{\Phi_{ND}}\rangle_F}\right]\right\rangle_{Q_{ND}}$$

$$= \ln\left[\left\langle\left\langle J(E_\nu, E', \cos\theta; \lambda)\right\rangle_{\Phi_{ND}}\right\rangle_F\right] - \left\langle \ln\left[\left\langle J(E_\nu, E', \cos\theta; \lambda)\right\rangle_{\Phi_{ND}}\right]\right\rangle_{Q_{ND}}$$

Straightforward to compute using Monte Carlo ensemble averages over data / flux / flat distributions

Far Detector



Far Detector

$$Q_{FD}(E', \cos\theta) = \frac{1}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{FD}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta))}{dE' d\cos\theta}$$

52

Far Detector

$$Q_{FD}(E', \cos\theta) = \frac{1}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{FD}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta))}{dE' d\cos\theta}$$

Multiplying and dividing by the ND normalization factor and ND flux we get

$$Q_{FD}(E', \cos\theta) = \frac{\mathcal{N}_{ND}}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{ND}(E_\nu) \left[\frac{\Phi_{FD}(E_\nu)}{\Phi_{ND}(E_\nu)}\right] P(E_\nu, E', \cos\theta; \lambda^*)$$

53

Far Detector

$$Q_{FD}(E', \cos\theta) = \frac{1}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{FD}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta))}{dE' d\cos\theta}$$

Multiplying and dividing by the ND normalization factor and ND flux we get

$$Q_{FD}(E', \cos\theta) = \frac{\mathcal{N}_{ND}}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{ND}(E_\nu) \left[\frac{\Phi_{FD}(E_\nu)}{\Phi_{ND}(E_\nu)}\right] P(E_\nu, E', \cos\theta; \lambda^*)$$

The oscillation probability

54

Far Detector

$$Q_{FD}(E', \cos\theta) = \frac{1}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{FD}(E_\nu) \frac{d^2\sigma^{(\ell A)}(E_\nu, E', \cos\theta))}{dE' d\cos\theta}$$

Multiplying and dividing by the ND normalization factor and ND flux we get

$$Q_{FD}(E', \cos\theta) = \frac{\mathcal{N}_{ND}}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{ND}(E_\nu) \left[\frac{\Phi_{FD}(E_\nu)}{\Phi_{ND}(E_\nu)}\right] P(E_\nu, E', \cos\theta; \lambda^*)$$

The oscillation probability

Fixed network parameters after ND training (best cross-section approximation)

55

$$Q_{FD}(E', \cos\theta) = \frac{\mathcal{N}_{ND}}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{ND}(E_\nu) \left[\frac{\Phi_{FD}(E_\nu)}{\Phi_{ND}(E_\nu)} \right] P(E_\nu, E', \cos\theta; \lambda^*)$$

$$\mathcal{P}(E_\nu; U_{PMNS}) \approx \frac{\Phi_{FD}(E_\nu)}{\Phi_{ND}(E_\nu)}$$

56

$$Q_{FD}(E', \cos\theta) = \frac{\mathcal{N}_{ND}}{\mathcal{N}_{FD}} \int dE_\nu \Phi_{ND}(E_\nu) \left[\frac{\Phi_{FD}(E_\nu)}{\Phi_{ND}(E_\nu)}\right] P(E_\nu, E', \cos\theta; \lambda^*)$$

$$\left\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \right\rangle_{\Phi_{ND}}$$

$$\mathcal{P}(E_\nu; U_{PMNS}) \approx \frac{\Phi_{FD}(E_\nu)}{\Phi_{ND}(E_\nu)}$$

57

Substituting all together

$$Q_{FD}(E', \cos\theta) \approx \frac{\int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS})}{\left\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \right\rangle_{\Phi_{ND}}}$$

58

Substituting all together

$$Q_{FD}(E', \cos\theta) \approx \frac{\int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS})}{\left\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \right\rangle_{\Phi_{ND}}}$$

$$= \frac{\langle P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \rangle_{\Phi_{ND}}}{\left\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \right\rangle_{\Phi_{ND}}}$$

59

Substituting all together

$$Q_{FD}(E', \cos\theta) \approx \frac{\int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda^*)\mathcal{P}(E_\nu; U_{PMNS})}{\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*)\mathcal{P}(E_\nu; U_{PMNS})\rangle_{\Phi_{ND}}}$$

$$= \frac{\langle P(E_\nu, E', \cos\theta; \lambda^*)\mathcal{P}(E_\nu; U_{PMNS})\rangle_{\Phi_{ND}}}{\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*)\mathcal{P}(E_\nu; U_{PMNS})\rangle_{\Phi_{ND}}}$$

$$\equiv R(E', \cos\theta; \lambda^*, U_{PMNS})$$

60

Substituting all together

$$Q_{FD}(E', \cos\theta) \approx \frac{\int dE_\nu \Phi_{ND}(E_\nu) P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS})}{\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \rangle_{\Phi_{ND}}}$$

$$= \frac{\langle P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \rangle_{\Phi_{ND}}}{\langle \int dE' d\cos\theta \; P(E_\nu, E', \cos\theta; \lambda^*) \mathcal{P}(E_\nu; U_{PMNS}) \rangle_{\Phi_{ND}}}$$

$$\equiv R(E', \cos\theta; \lambda^*, U_{PMNS})$$

We have everything to compute the Loss Function to train our FD Model

61

$$D(Q_{FD}||R) = \int dE' d\cos\theta \, Q_{FD}(E', \cos\theta) \ln\left[\frac{Q_{FD}(E', \cos\theta)}{R(E', \cos\theta; \lambda^*, U_{PMNS})}\right]$$

62

$$D(Q_{FD}||R) = \int dE' d\cos\theta \, Q_{FD}(E', \cos\theta) \ln\left[\frac{Q_{FD}(E', \cos\theta)}{R(E', \cos\theta; \lambda^*, U_{PMNS})}\right]$$

$$\equiv \mathcal{C}_{FD}(E', \cos\theta) - \mathcal{L}_{FD}(E_\nu, E', \cos\theta; \lambda^*, U_{PMNS})$$

Irrelevant constant term

Loss function we minimize

63

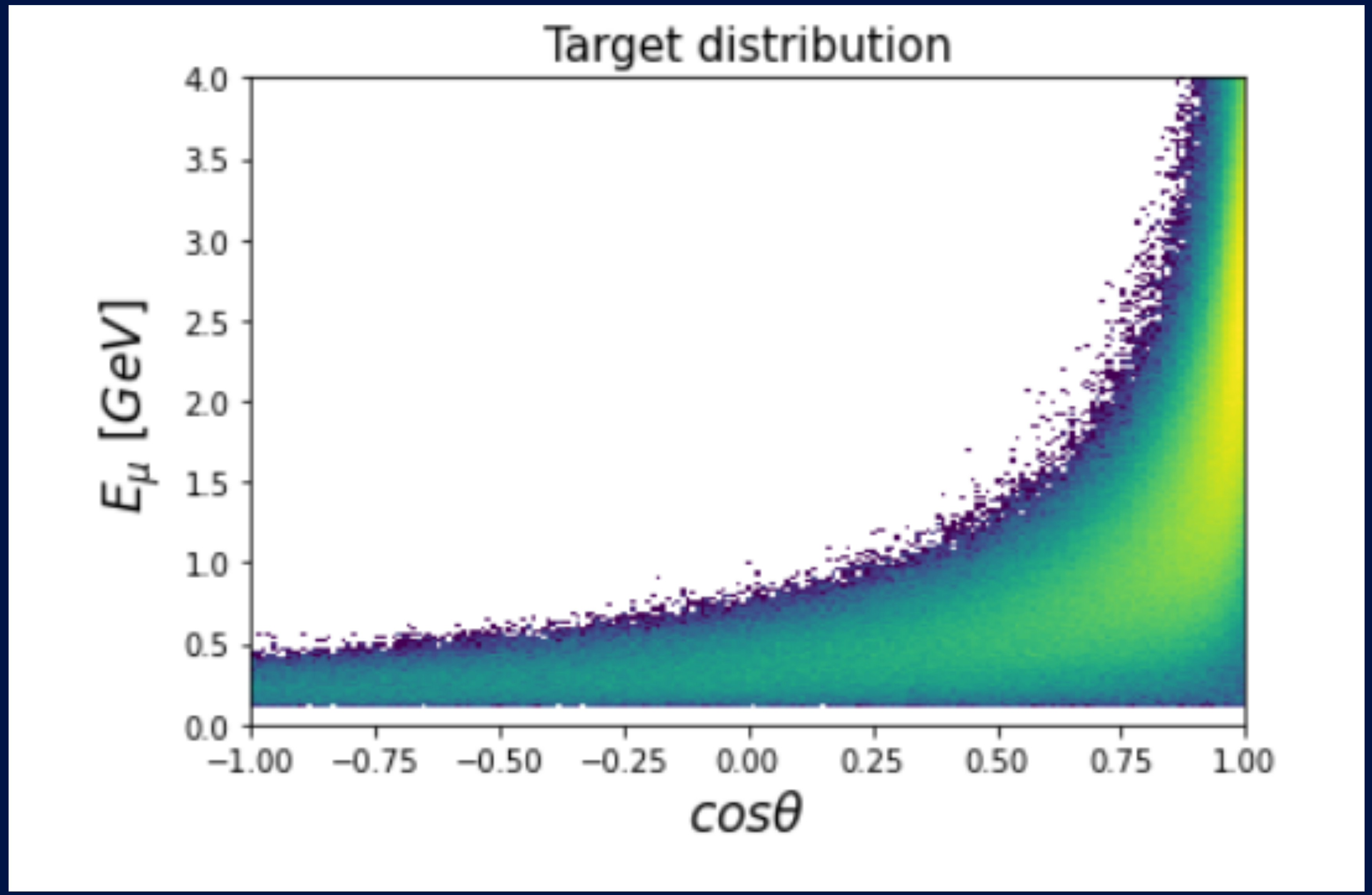
$$D(Q_{FD}||R) = \int dE' d\cos\theta \, Q_{FD}(E', \cos\theta) \ln\left[\frac{Q_{FD}(E', \cos\theta)}{R(E', \cos\theta; \lambda^*, U_{PMNS})}\right]$$

$$\equiv \mathcal{C}_{FD}(E', \cos\theta) - \mathcal{L}_{FD}(E_\nu, E', \cos\theta; \lambda^*, U_{PMNS})$$

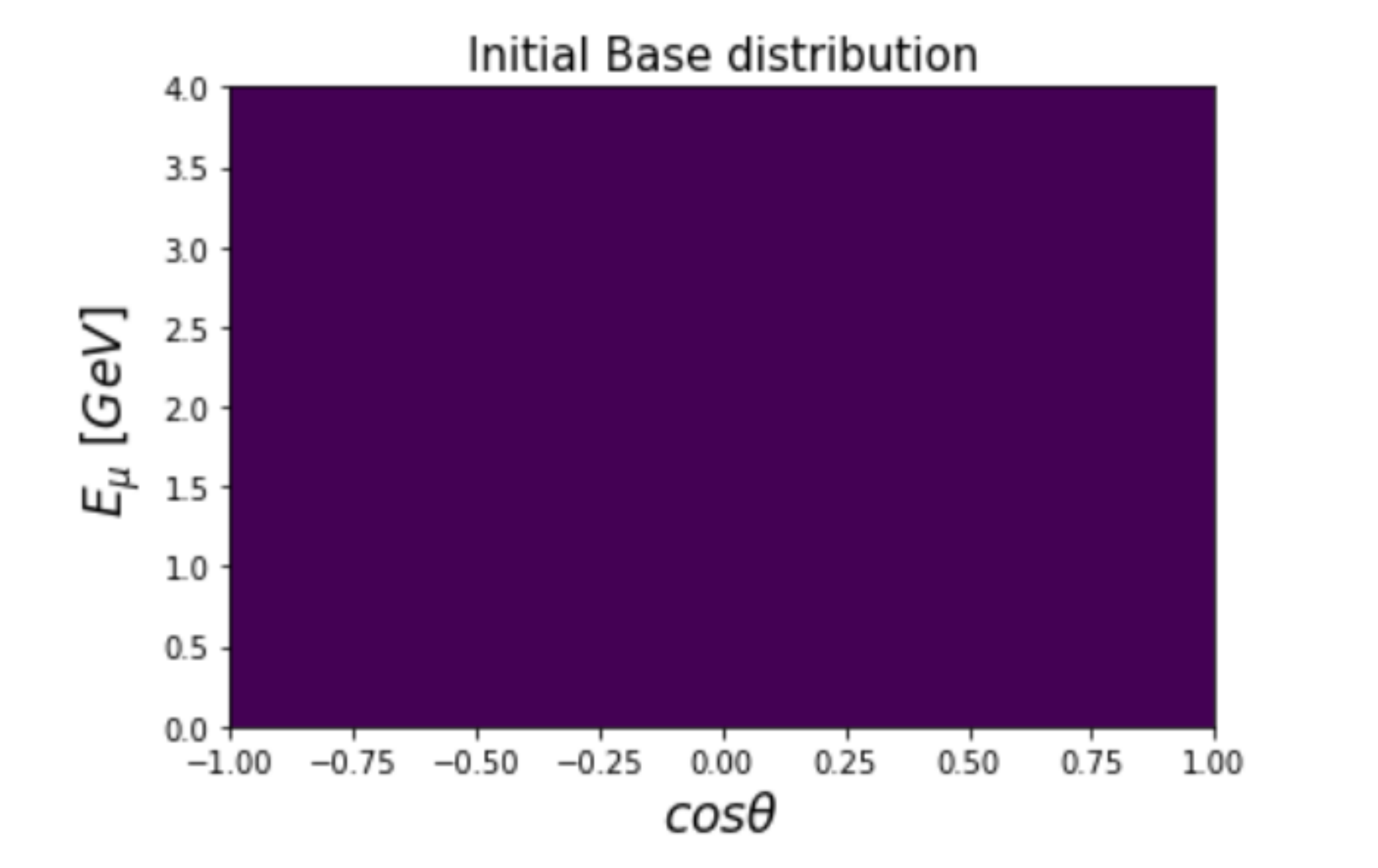
Doing some math and averaging over a flat distribution (as in ND)

$$\mathcal{L}_{FD} = \ln\left[\left\langle\left\langle J(E_\nu, E', \cos\theta; \lambda^*)\mathcal{P}(E_\nu; U_{PMNS})\right\rangle_F\right\rangle_{\Phi_{ND}}\right]$$
$$-\left\langle \ln\left[\left\langle J(E_\nu, E', \cos\theta; \lambda^*)\mathcal{P}(E_\nu; U_{PMNS})\right\rangle_{\Phi_{ND}}\right]\right\rangle_{Q_{FD}}$$

Preliminary Results

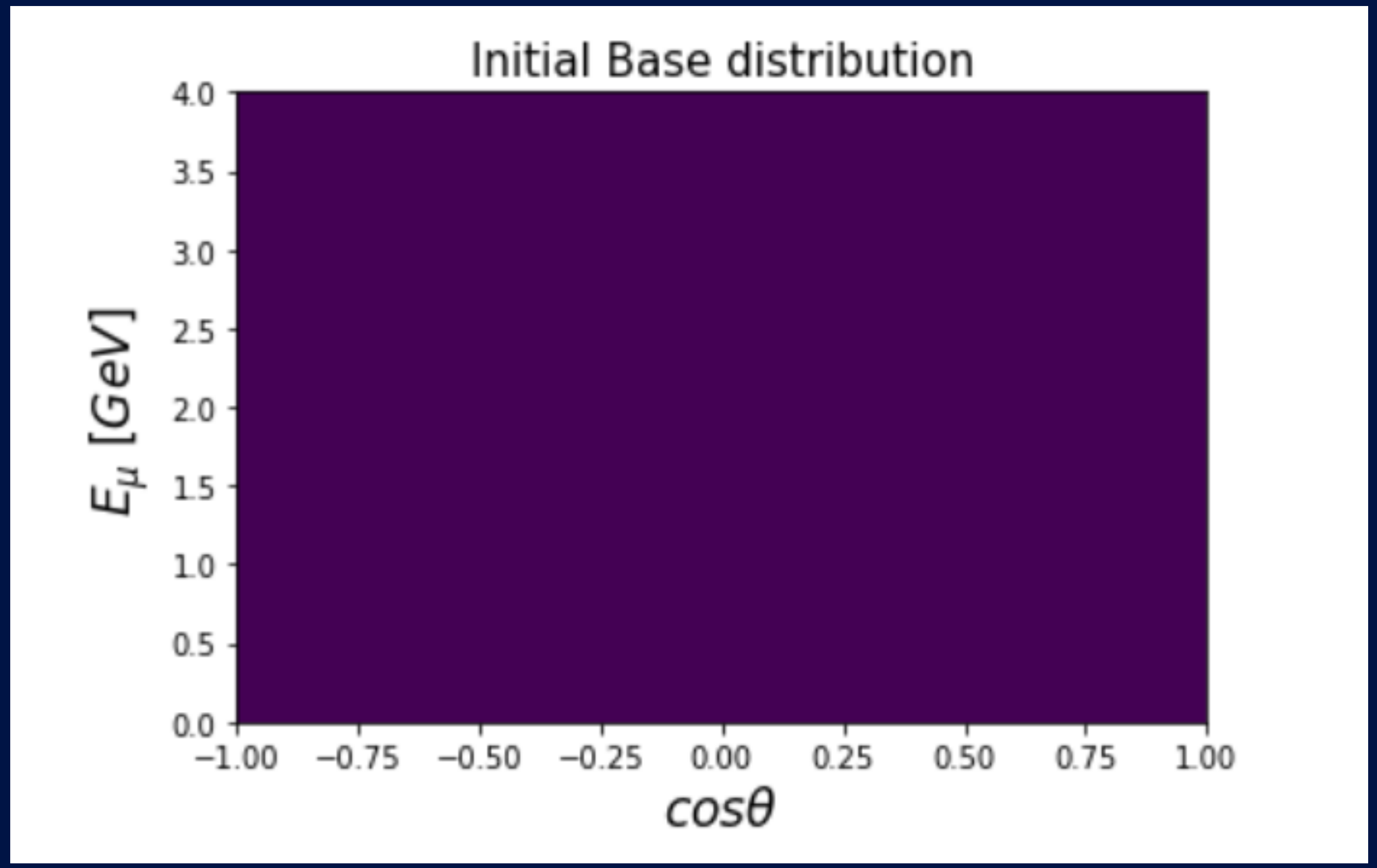


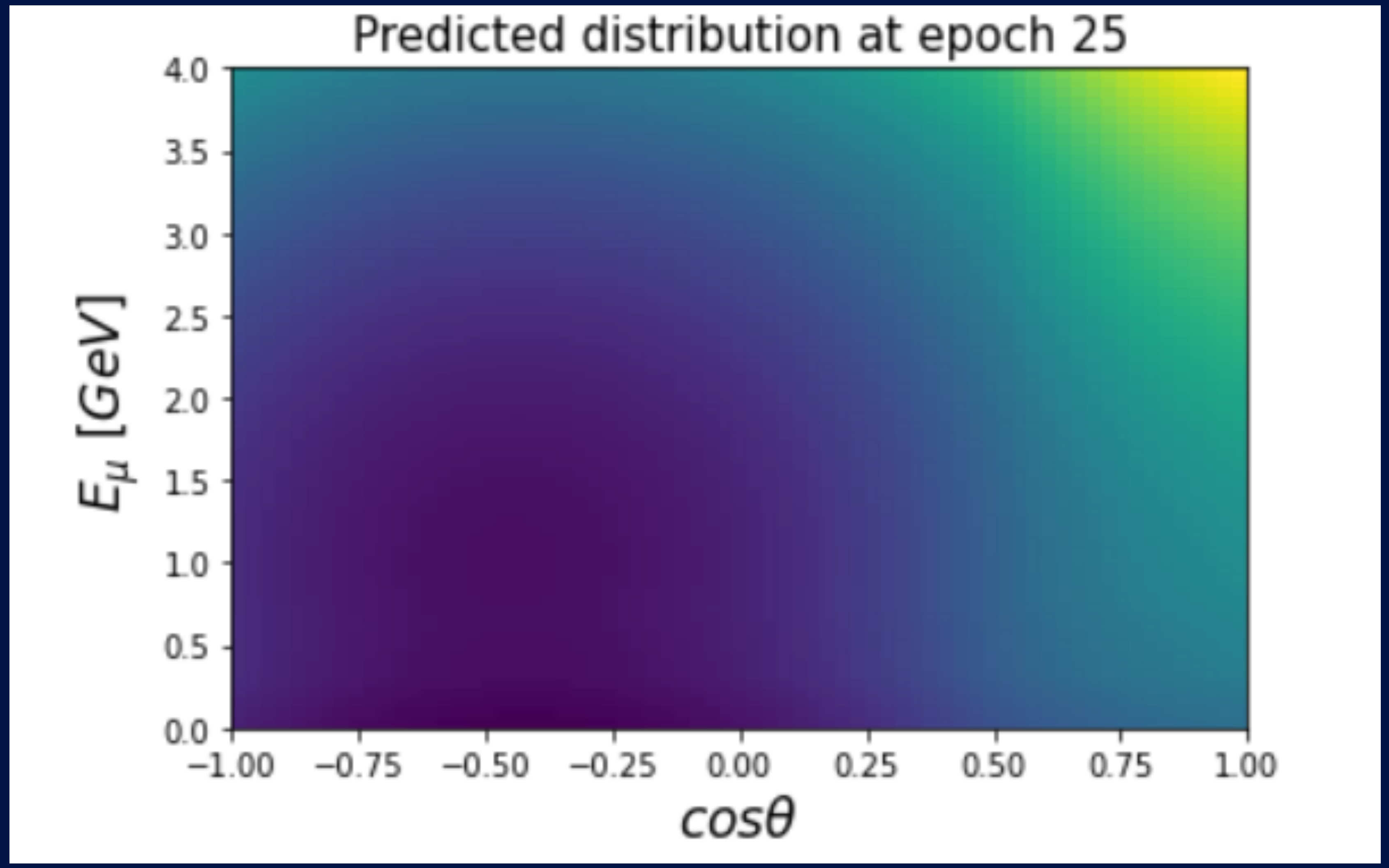
Near Detector simulated data

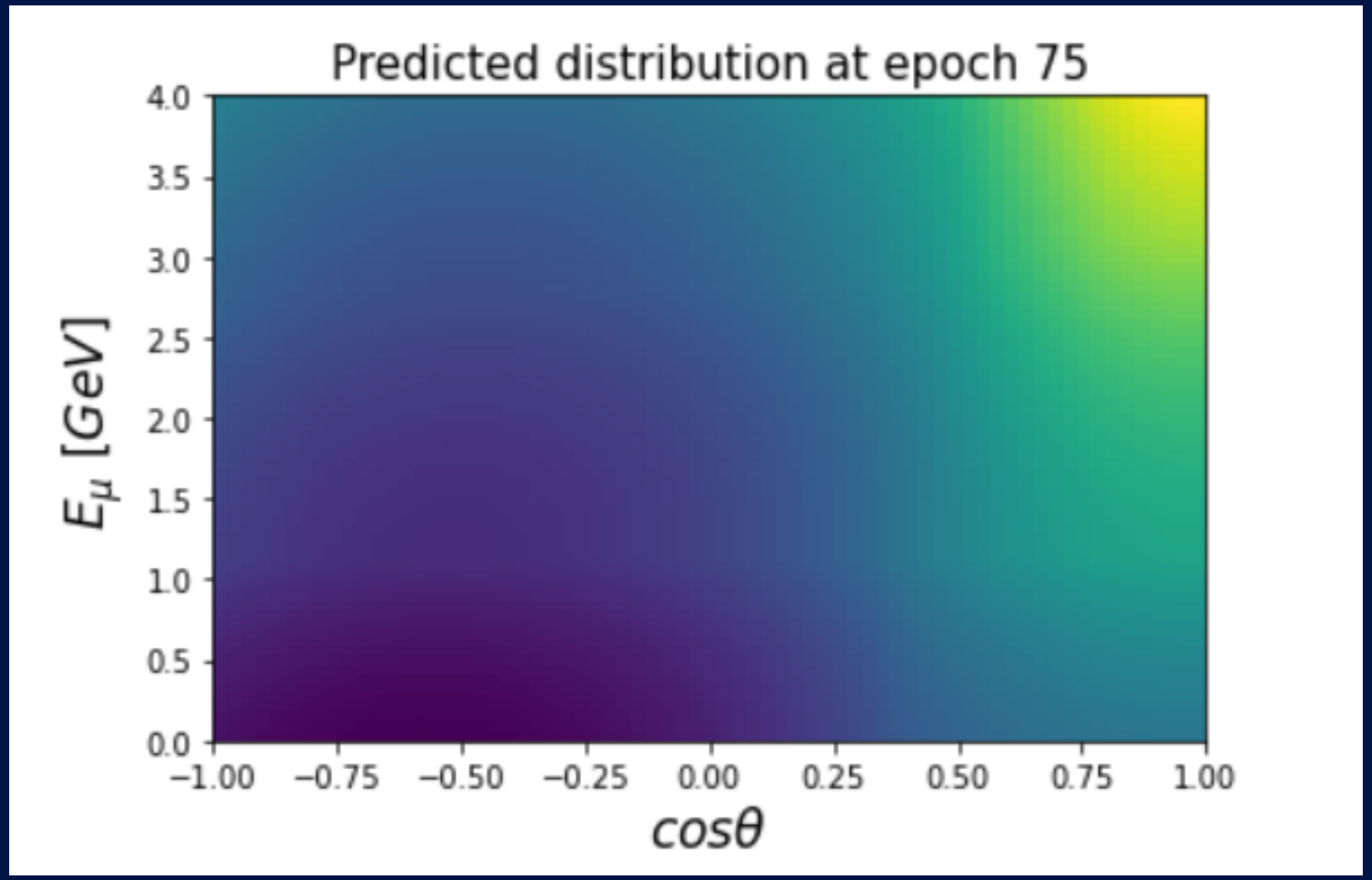


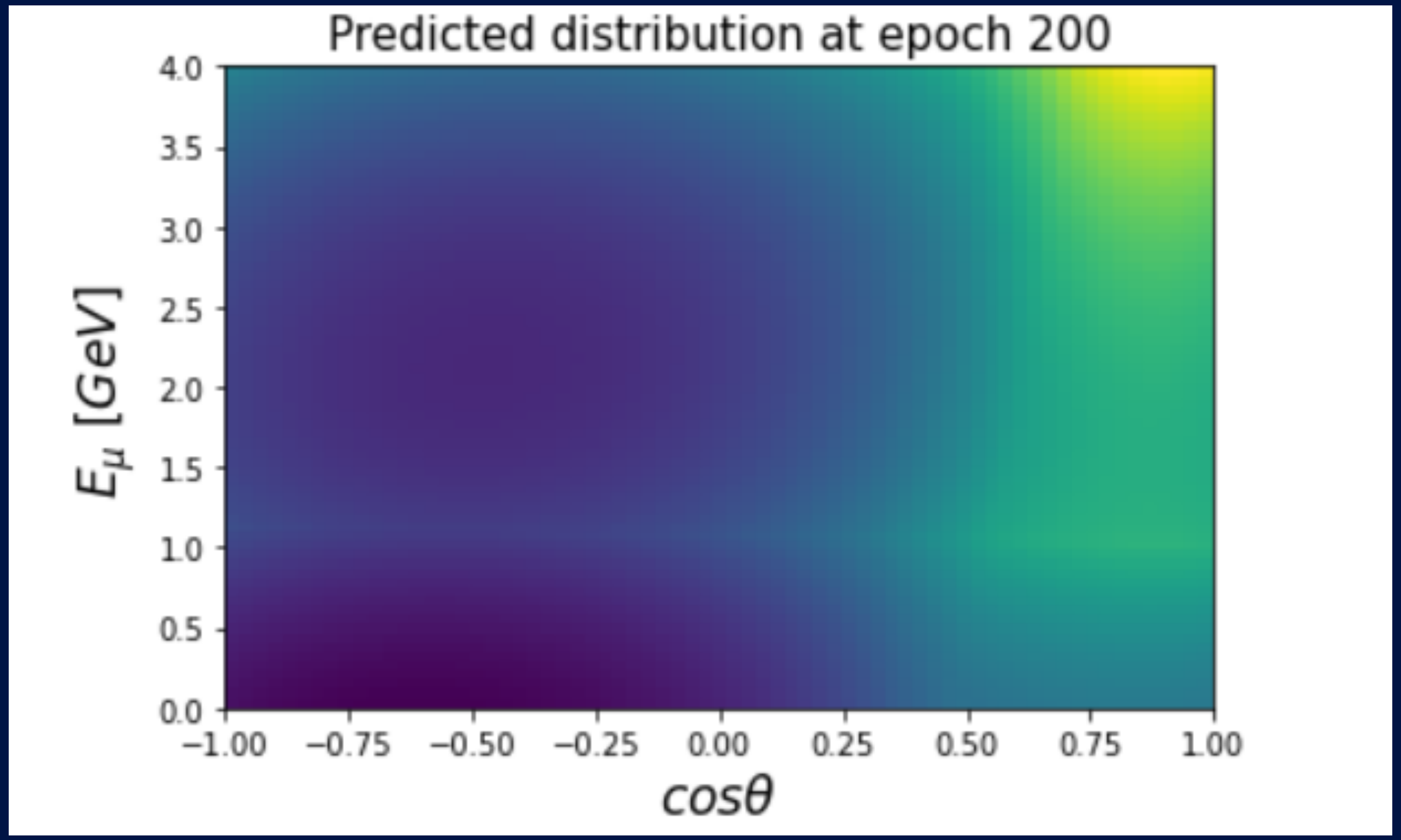
Uniform distribution

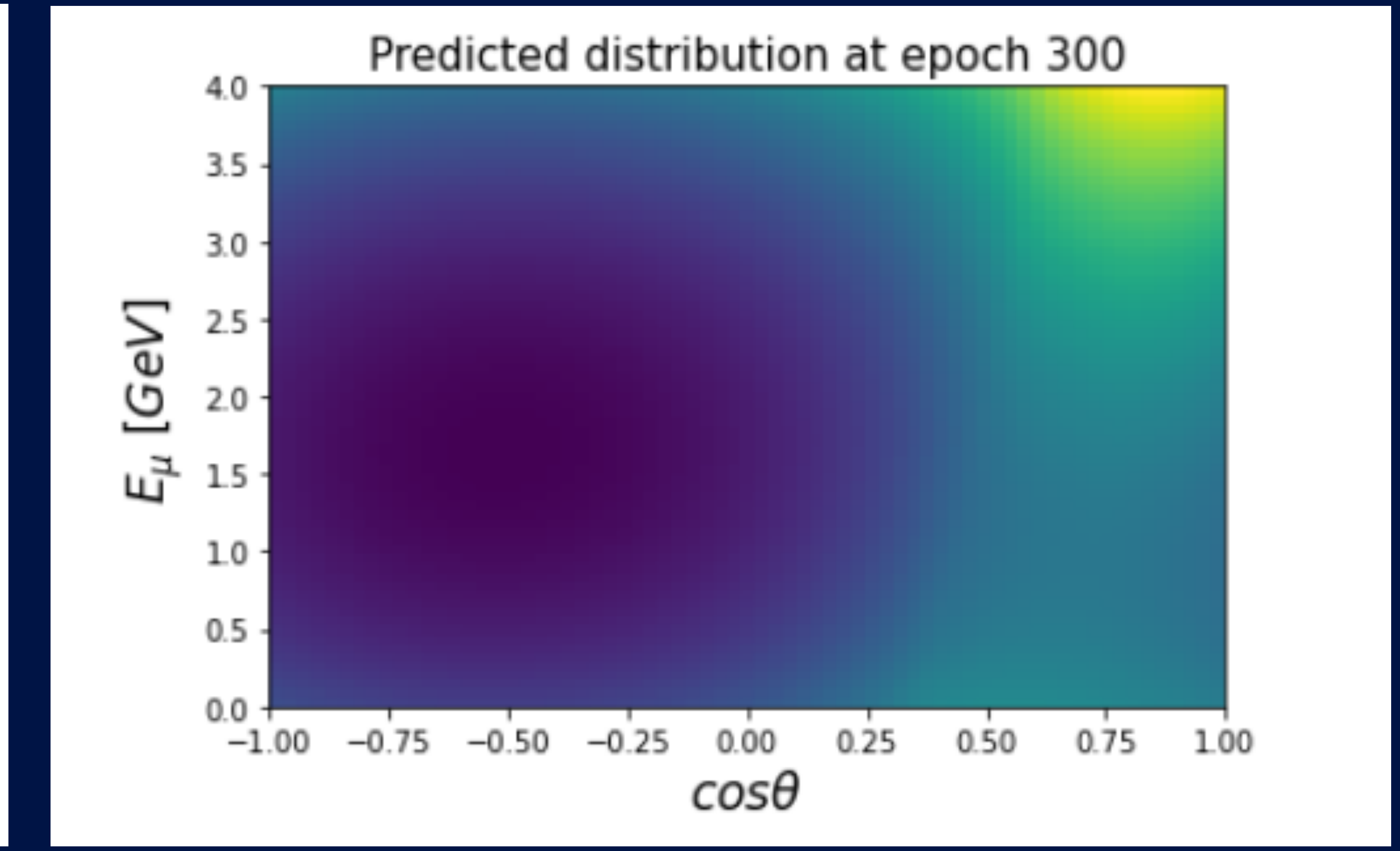
Training

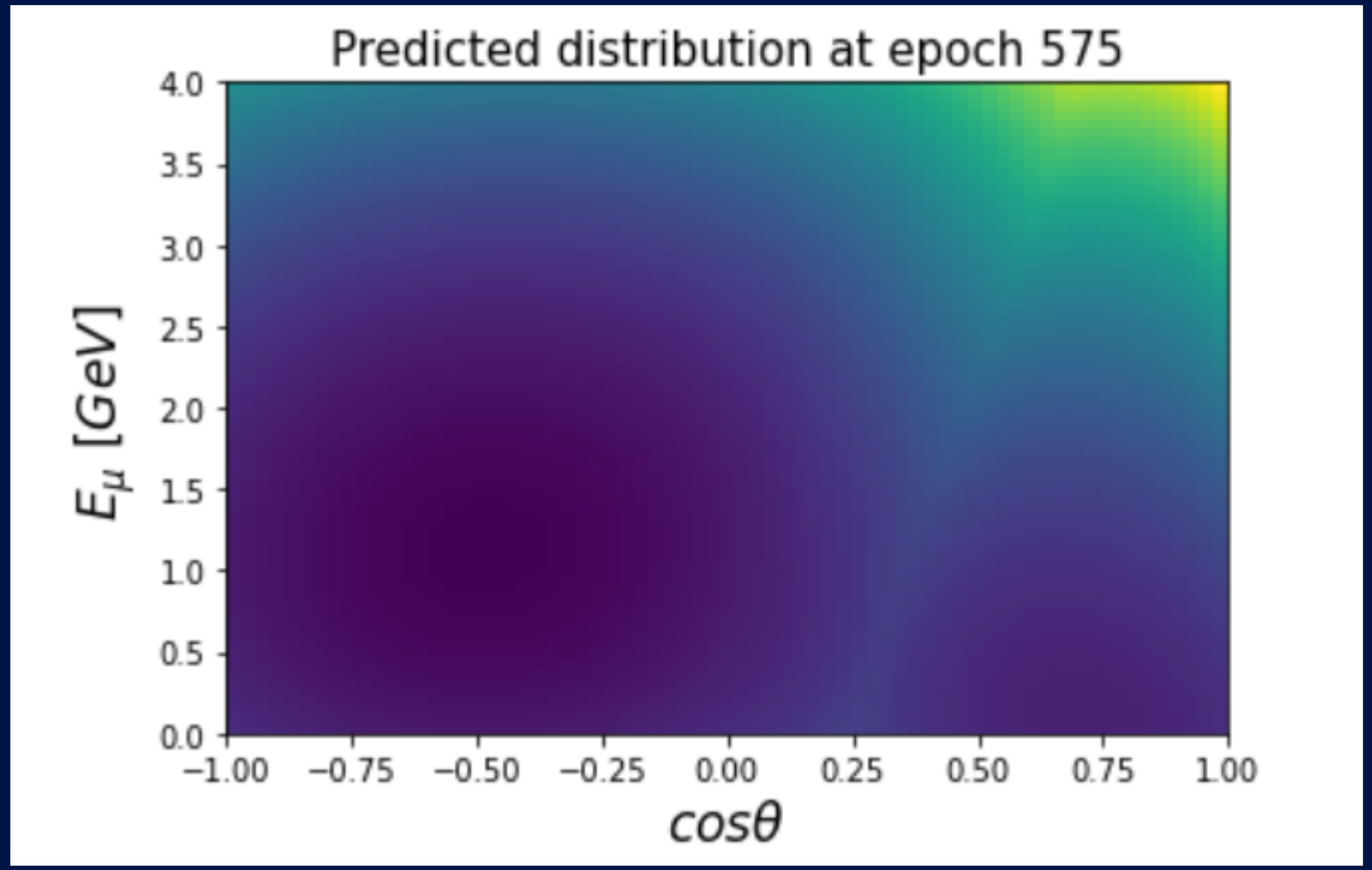

Initial Base distribution
E_μ [GeV]
$cos\theta$


Predicted distribution at epoch 25
E_μ [GeV]
$cos\theta$

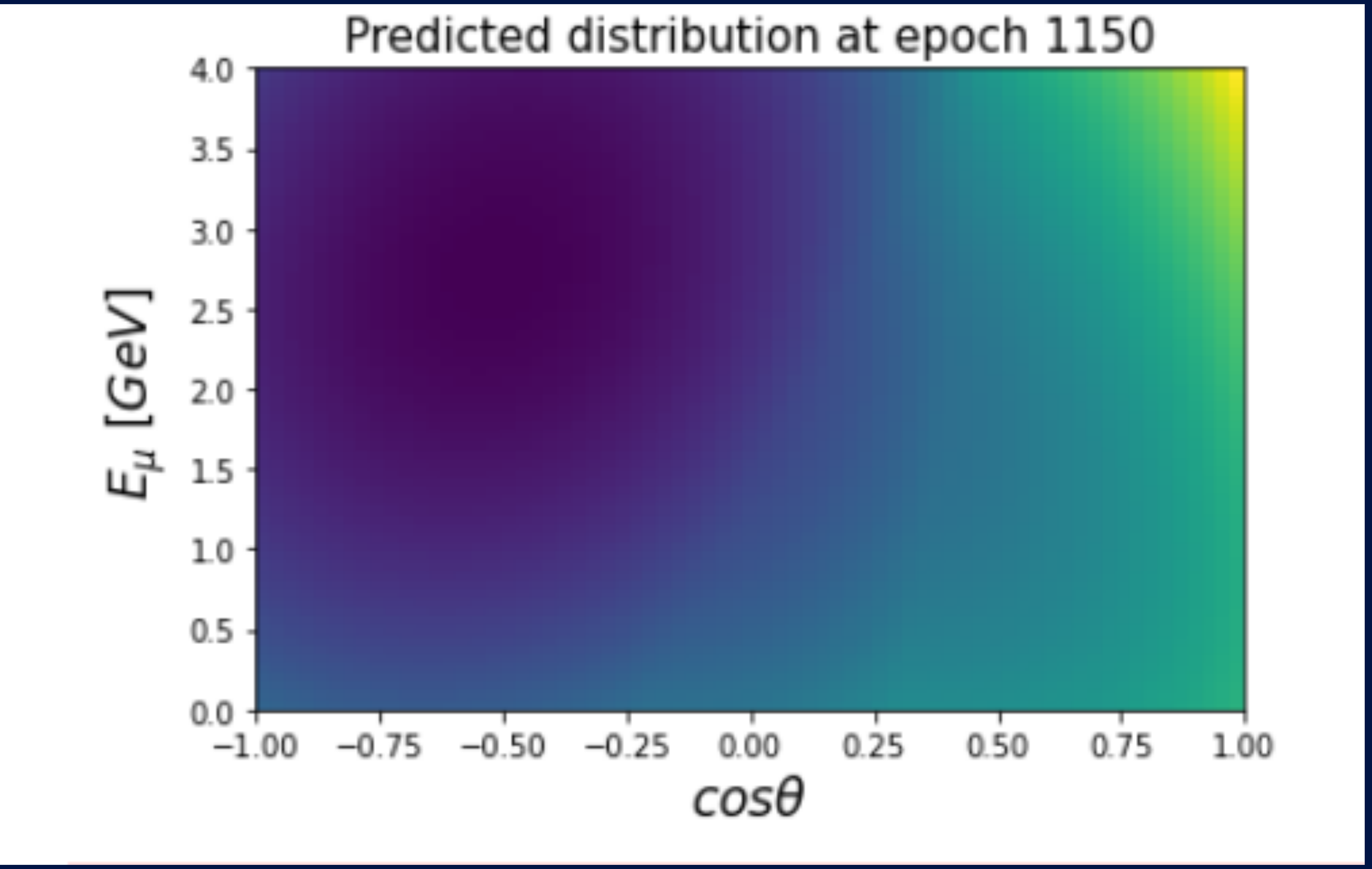

Predicted distribution at epoch 75
E_μ [GeV]
$cos\theta$

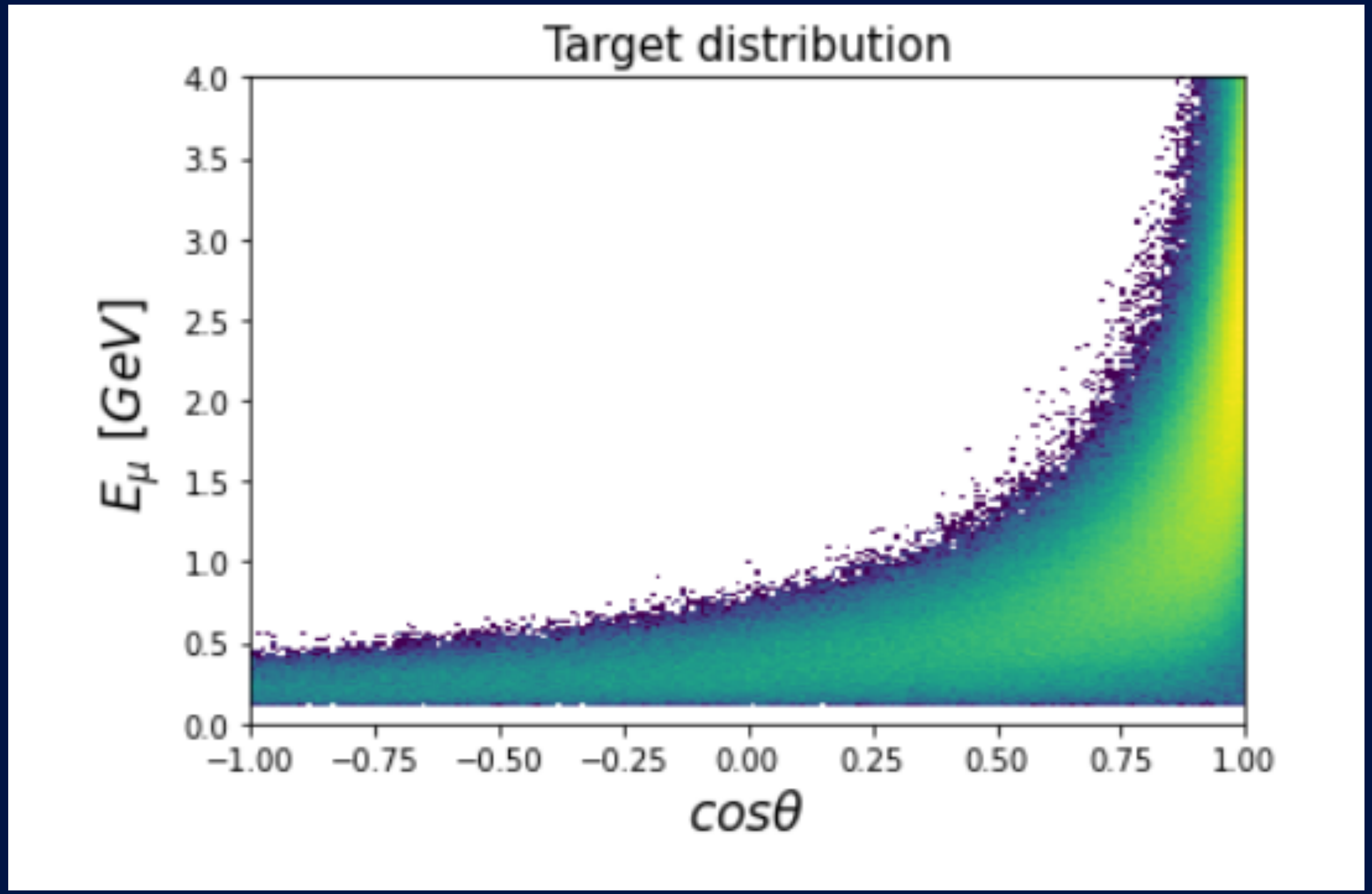

Predicted distribution at epoch 200
E_μ [GeV]
$cos\theta$


Predicted distribution at epoch 300
E_μ [GeV]
$cos\theta$

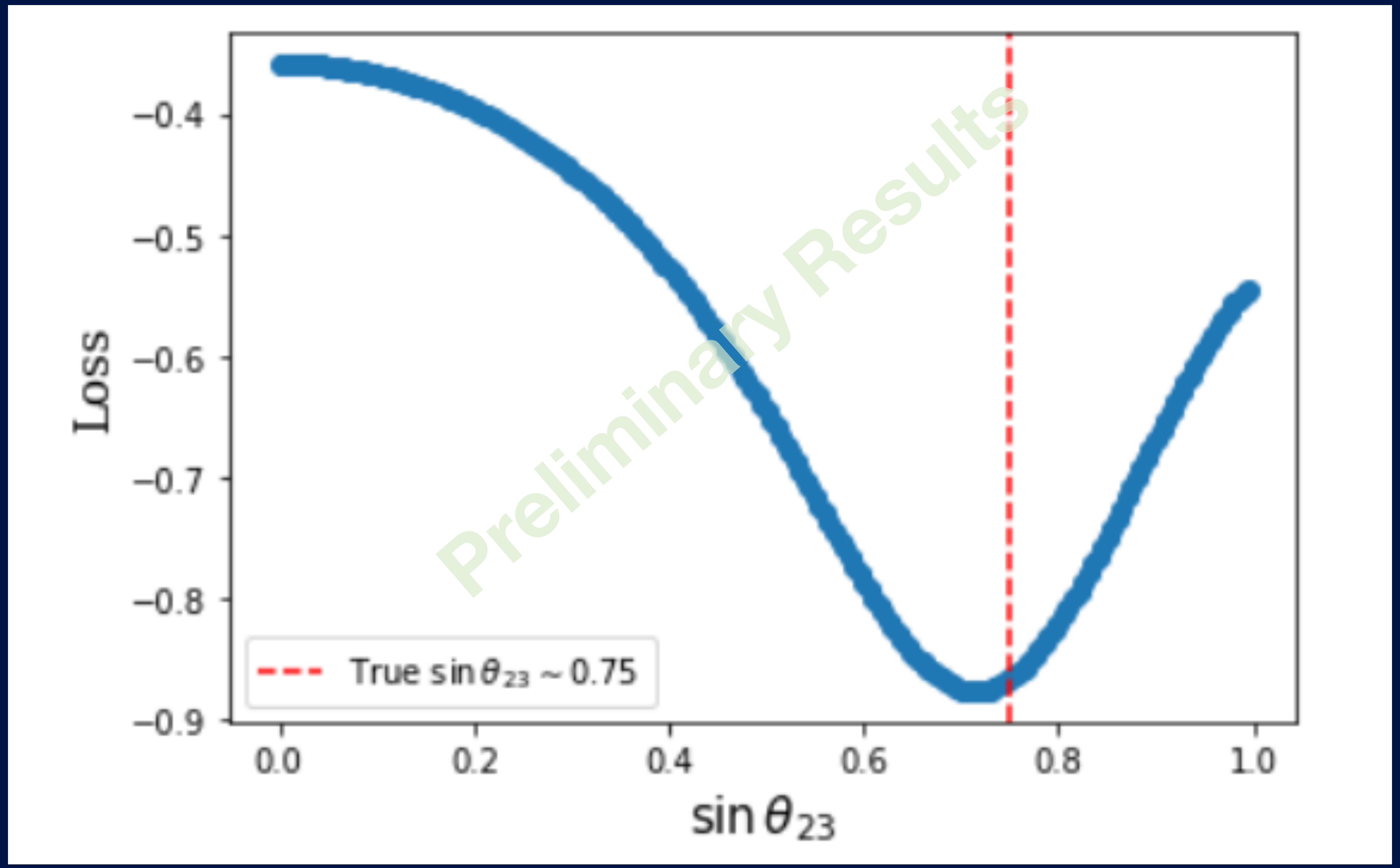

Predicted distribution at epoch 575
E_μ [GeV]
$cos\theta$

Training


Predicted distribution at epoch 1150
E_{μ} [GeV]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
cosθ

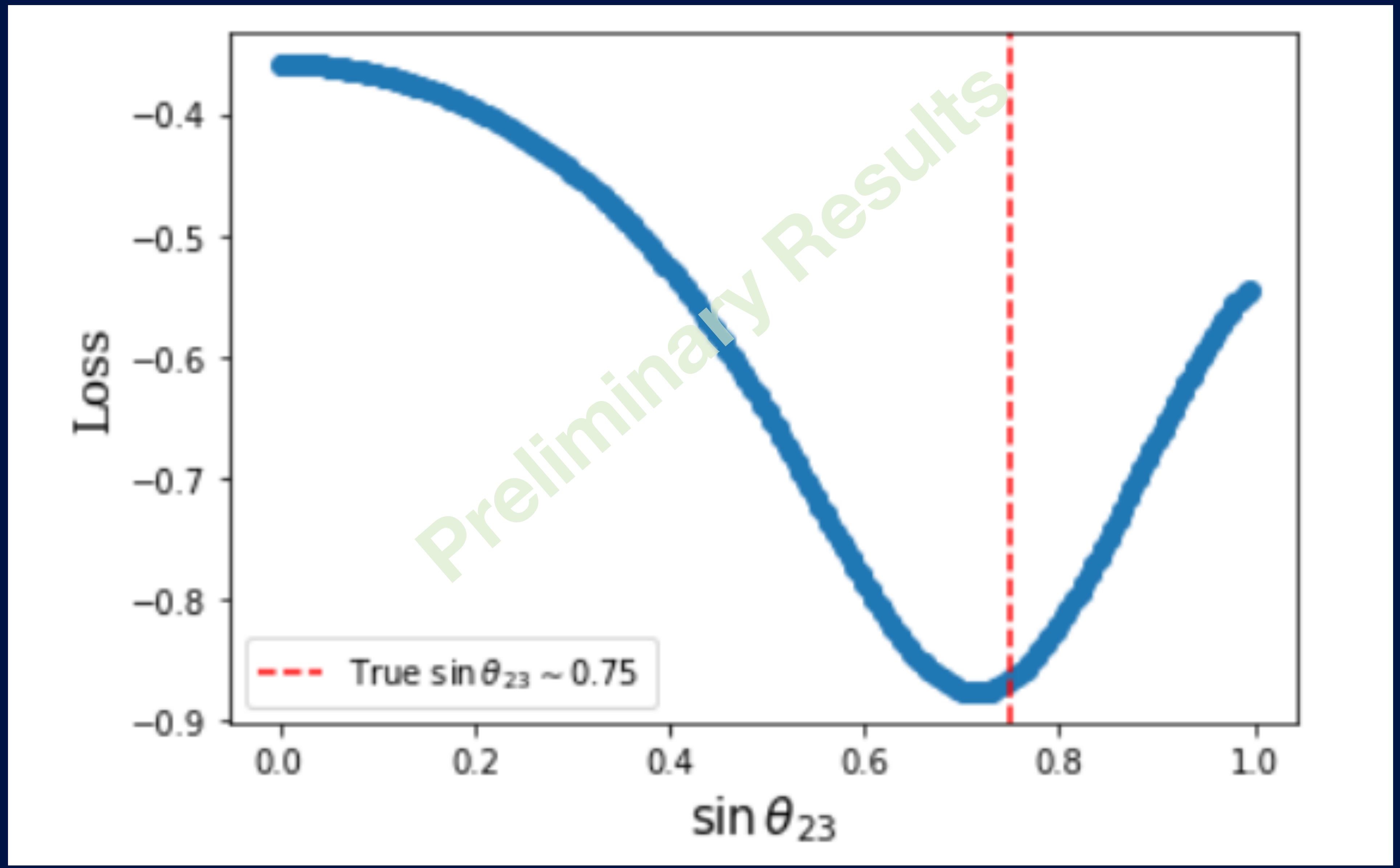

Target distribution
E_{μ} [GeV]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
cosθ

Preliminary Results



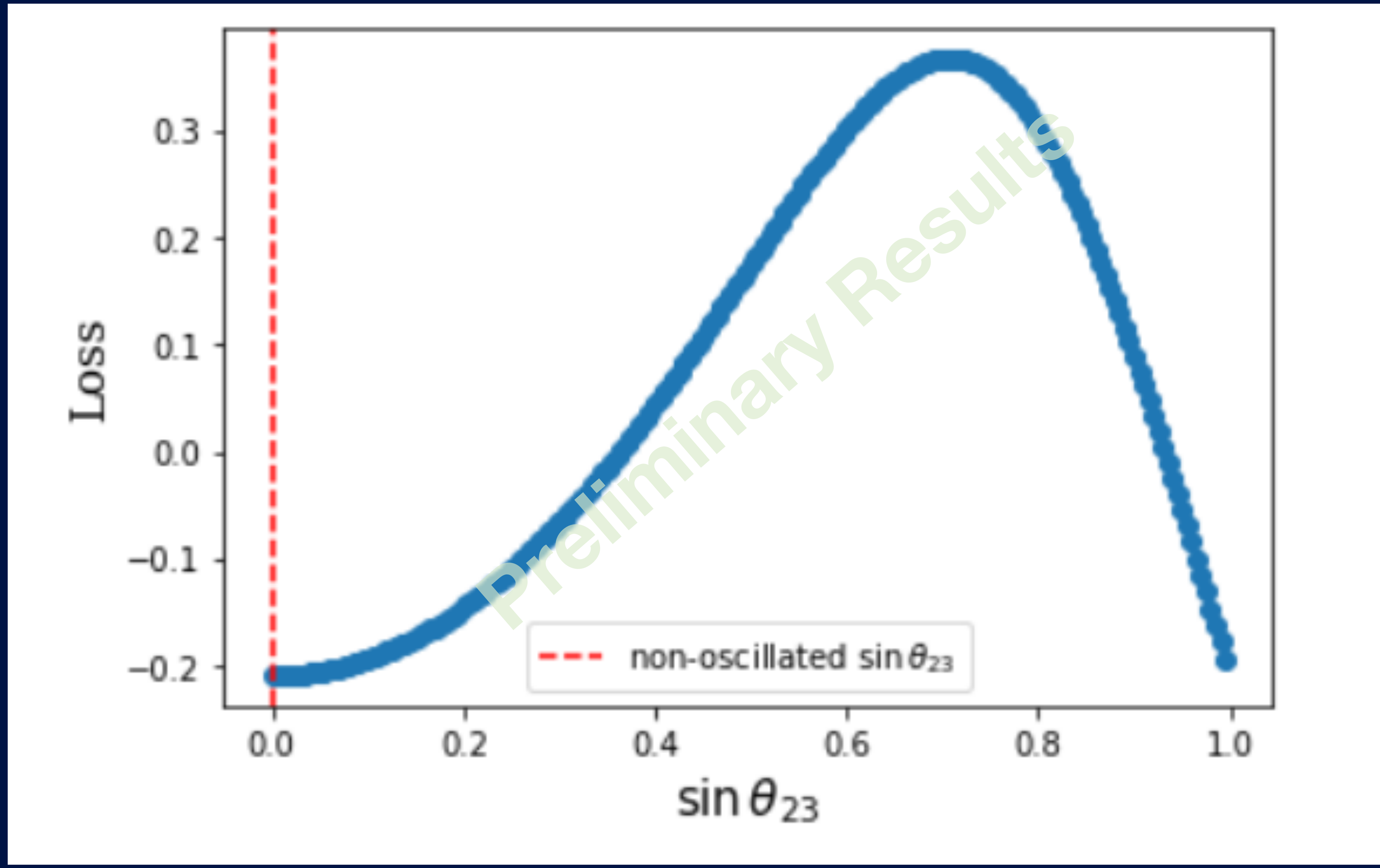
Far Detector simulated data
with oscillations ▲

Preliminary Results



Far Detector simulated data
with oscillations ▲

Far Detector simulated data
without oscillations ▼

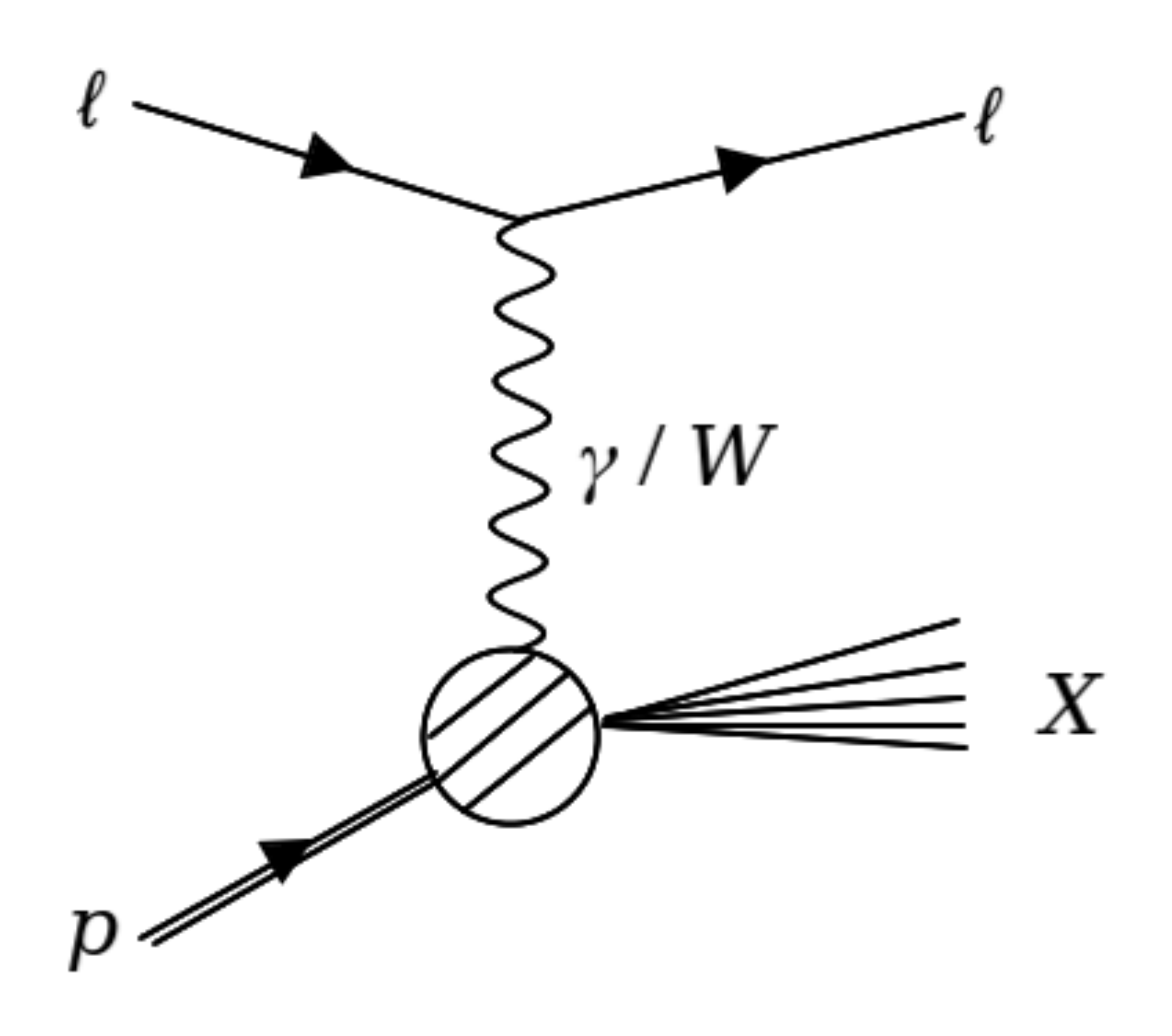


What could go wrong?

- 3D cross section model trained to learn one 2D marginalization (near detector), then used to predict results for another 2D marginalization (far detector)
- Generalization to out of training data generic challenge for machine learning
- Can we encode cross section physics in a 2D distribution?

Part II: Fundamental Theory Assumptions

(Structure Functions)

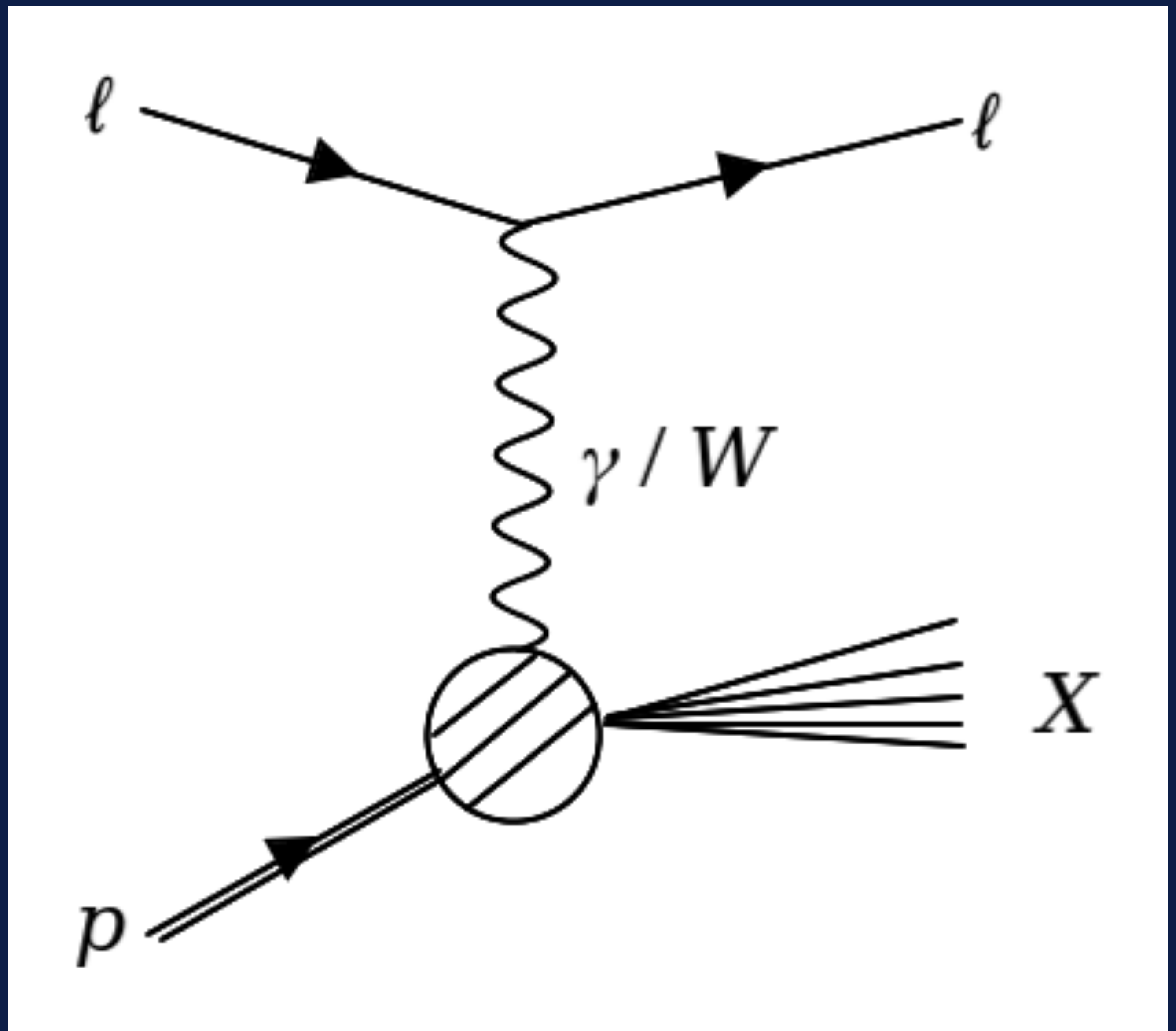


$q = k - k'$

k = incoming lepton momentum

p = nucleus with momentum $(M_A, \vec{0})$

p' = hadronic remnant X momentum $p' = k + p - k'$



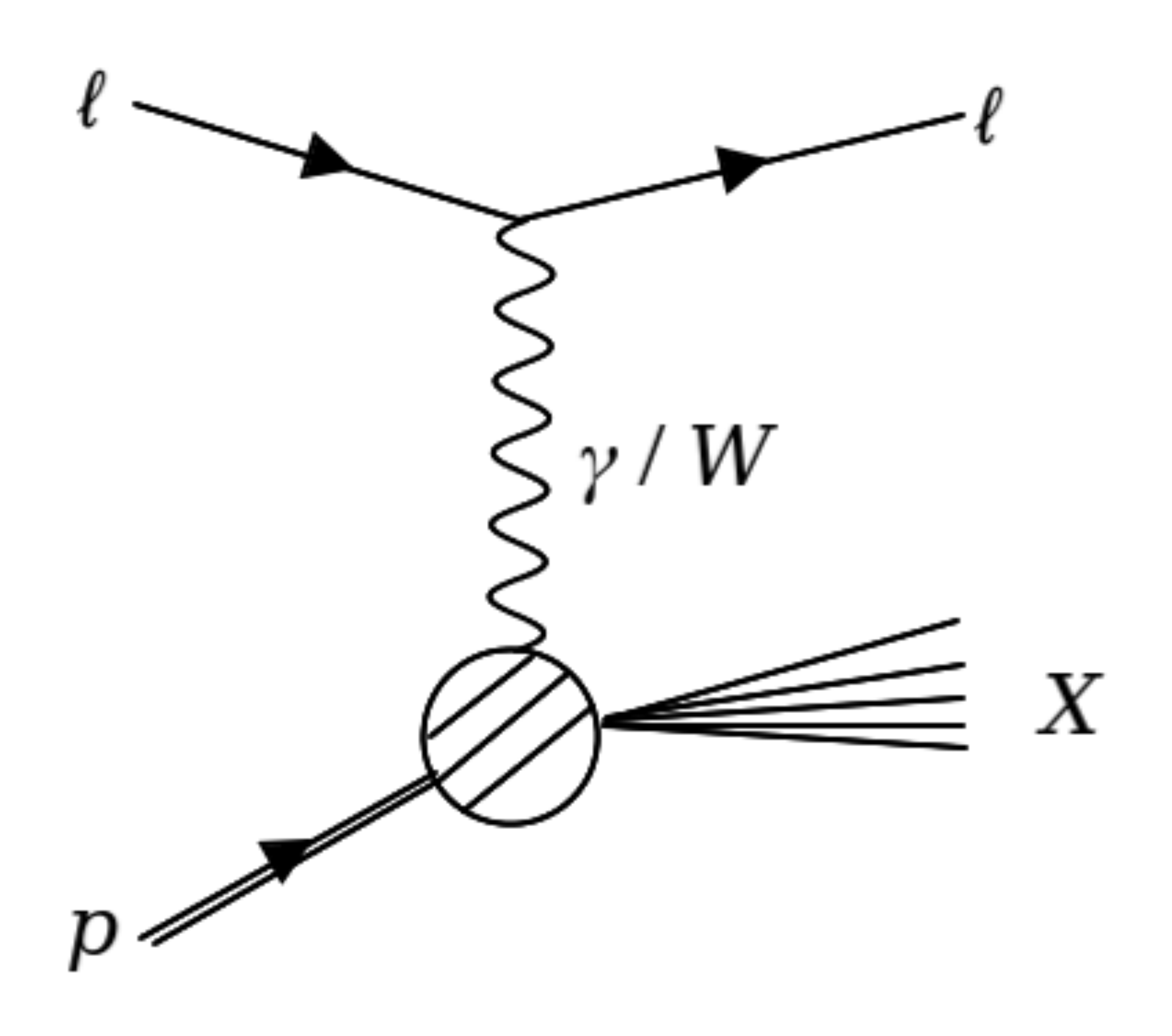
$q = k - k'$

k = incoming lepton momentum

p = nucleus with momentum $(M_A, \vec{0})$

p' = hadronic remnant X momentum $p' = k + p - k'$

$$\mathcal{M}^{(\nu A)} = \frac{\sqrt{2} G_F}{1 + q^2/M_W^2} \left[\overline{u}(k', \lambda') \left(\frac{1 - \gamma_5}{2} \right) \gamma^\mu u(k, \lambda) \right] \langle X(p') V_\mu^+ - A_\mu^+ A(p) \rangle$$



$q = k - k'$

k = incoming lepton momentum

p = nucleus with momentum $(M_A, \vec{0})$

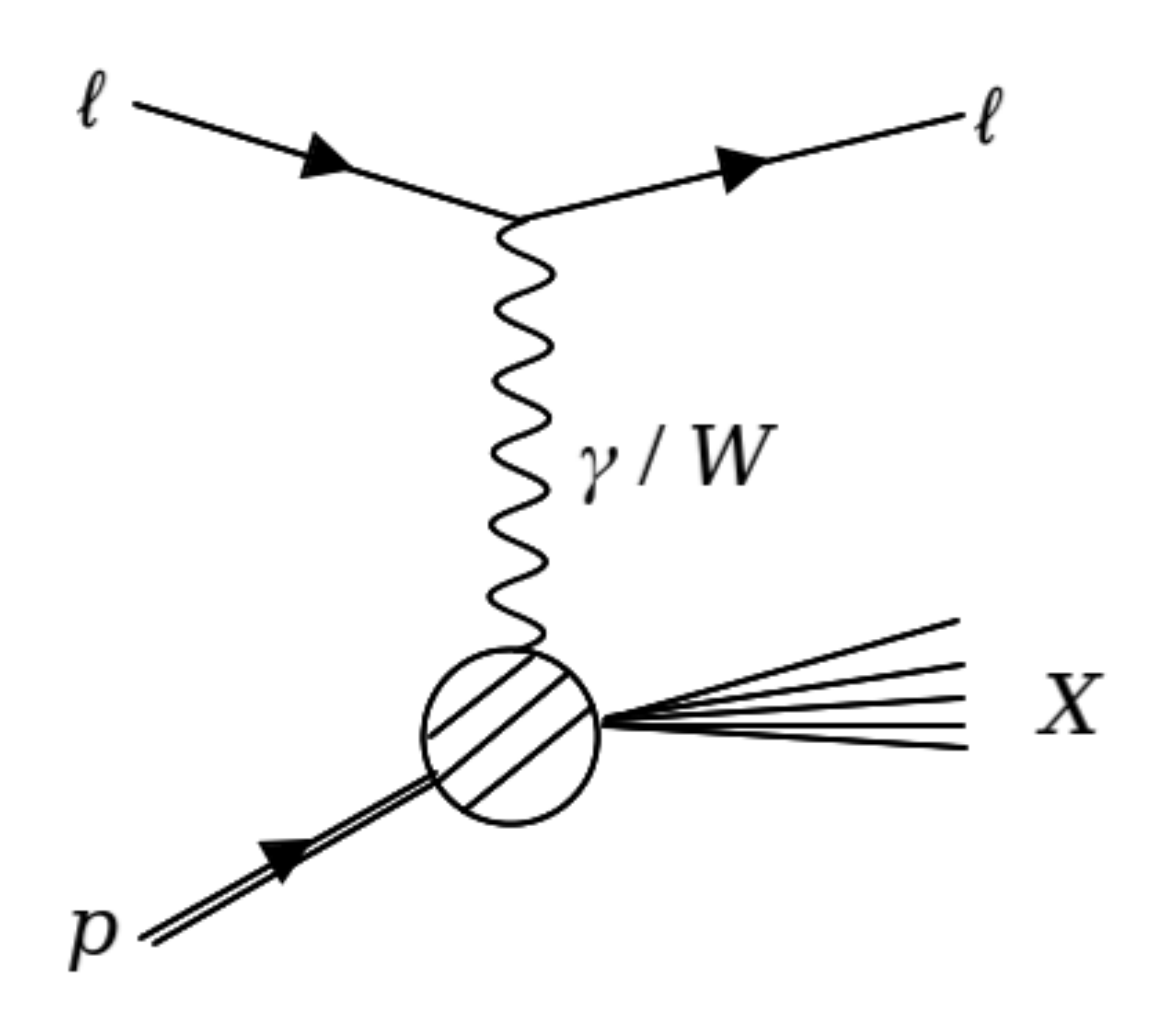
p' = hadronic remnant X momentum $p' = k + p - k'$

$$\mathcal{M}^{(\nu A)} = \frac{\sqrt{2} G_F}{1 + q^2/M_W^2} \left[\overline{u}(k', \lambda') \left(\frac{1-\gamma_5}{2} \right) \gamma^\mu u(k, \lambda) \right] \langle X(p') V_\mu^+ - A_\mu^+ A(p) \rangle$$

$$\frac{d^2\sigma^{(\nu A)}}{dE' d\cos\theta} = C \ell_{\mu\nu} W^{\mu\nu} = \sum_i K_i(E_\nu, x, Q^2) W_i(x, Q^2)$$

$$x = \frac{q^2}{p \cdot q}$$

$$Q^2 = (q^0)^2 - (\vec{q})^2$$



$q = k - k'$

k = incoming lepton momentum

p = nucleus with momentum $(M_A, \vec{0})$

p' = hadronic remnant X momentum $p' = k + p - k'$

$$\mathcal{M}^{(\nu A)} = \frac{\sqrt{2}G_F}{1 + q^2/M_W^2}\left[\overline{u}(k',\lambda')\left(\frac{1-\gamma_5}{2}\right)\gamma^\mu u(k,\lambda)\right]\langle X(p')V_\mu^+ - A_\mu^+ A(p)\rangle$$

$$\frac{d^2\sigma^{(\nu A)}}{dE' d\cos\theta} = C\ell_{\mu\nu}W^{\mu\nu} = \sum_i K_i(E_\nu, x, Q^2)W_i(x, Q^2)$$

$$x = \frac{q^2}{p \cdot q}$$

$$K_i(E_\nu, x, Q^2) = \ell_{\mu\nu}B_i^{\mu\nu}$$

$$W^{\mu\nu}(x, Q^2) = \sum_i W_i(x, Q^2)B_i^{\mu\nu}$$

$$Q^2 = (q^0)^2 - (\vec{q})^2$$

$B_i^{\mu\nu}$ -> Basis tensors

known kinematic functions

$$\ell_{\mu\nu}^{(\nu A)} = \frac{1}{2}\sum_{\lambda,\lambda'}\left[\overline{u}(k',\lambda')\left(\frac{1-\gamma_5}{2}\right)\gamma_\mu u(k,\lambda)\right]\left[\overline{u}(k,\lambda)\left(\frac{1-\gamma_5}{2}\right)\gamma_\nu u(k',\lambda')\right]$$

$$= k_\mu k'_\nu + k'_\mu k_\nu - g_{\mu\nu}(k\cdot k') + i\epsilon_{\mu\nu\rho\sigma}k^\rho k'^\sigma$$

known kinematic functions

$$\ell_{\mu\nu}^{(\nu A)} = \frac{1}{2}\sum_{\lambda,\lambda'}\left[\overline{u}(k',\lambda')\left(\frac{1-\gamma_5}{2}\right)\gamma_\mu u(k,\lambda)\right]\left[\overline{u}(k,\lambda)\left(\frac{1-\gamma_5}{2}\right)\gamma_\nu u(k',\lambda')\right]$$

$$= k_\mu k'_\nu + k'_\mu k_\nu - g_{\mu\nu}(k\cdot k') + i\epsilon_{\mu\nu\rho\sigma}k^\rho k'^\sigma$$

Structure Functions

$$W_{\mu\nu}^{(\nu A)} = \frac{1}{x}g_{\mu\nu}W_1^{(\nu A)}(x,q^2) + \left(\frac{p_\mu p_\nu}{M_A^2}\right)W_2^{(\nu A)}(x,q^2) + i\epsilon_{\mu\nu\rho\sigma}\left(\frac{p^\rho q^\sigma}{q^2}\right)W_3^{(\nu A)}(x,q^2)$$

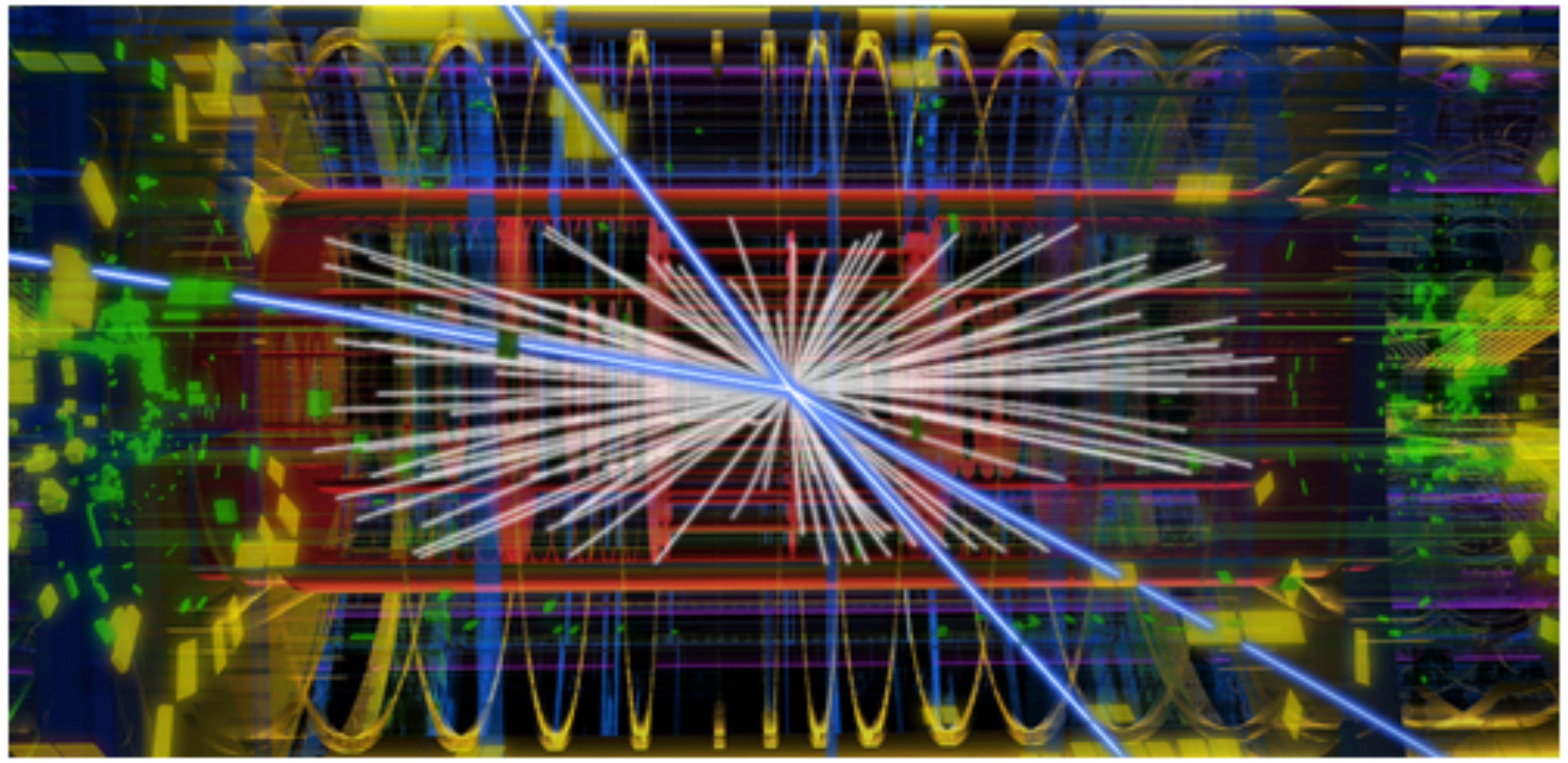
$$+\frac{q_\mu q_\nu}{xq^2}W_4^{(\nu A)}(x,q^2) + \frac{2}{q^2}(p_\mu q_\nu + q_\mu p_\nu)W_5^{(\nu A)}(x,q^2)$$

Parameterizing our ignorance

Difference between I and II

	Positive $0 >$	Loss	Parametrization
$\frac{d^2\sigma^{(\bar{\nu}A)}}{dE' d\cos\theta}$	✓	KL-Divergence	Normalizing Flows
W_i	✗	-	Neural Networks
$K_i \cdot W_i$	✓	KL-Divergence	-

NNPDF Home The collaboration Research For users Documents Media For the public



The NNPDF collaboration performs research in the field of high-energy physics. The collaboration is not for profit, and funded by national and international educational and research institutions and funding agencies, such as Universities, Research councils, and Research laboratories. The scientific output of the collaboration is mostly in the form of scientific publications and software, and it is all freely available to the public, to scientists, and to interested parties, through the arXiv, journal repositories, and software repositories. The NNPDF collaboration also effectively acts as an educational and training service, through the affiliation of several of its members with various undergraduate and graduate schools: a large number of theses has been performed in the framework of the collaboration.

The NNPDF collaboration determines the structure of the proton using contemporary methods of artificial intelligence. A precise knowledge of the so-called Parton Distribution Functions (PDFs) of the proton, which describe their structure in terms of their quark and gluon constituents, is a crucial ingredient of the physics program of the Large Hadron Collider of CERN. It has played an important role in the discovery of the Higgs boson. Its incomplete knowledge is one of the main limitations in searches of new physics.
PDFs cannot be computed from first principles: they have to be extracted from the data, through a careful comparison of theoretical predictions and experimental results. NNPDF determines PDFs using as an unbiased modeling tool Neural Networks, trained using Genetic Algorithms, and used to construct a Monte Carlo representation of PDFs and their uncertainties: a probability distribution in a space of functions.

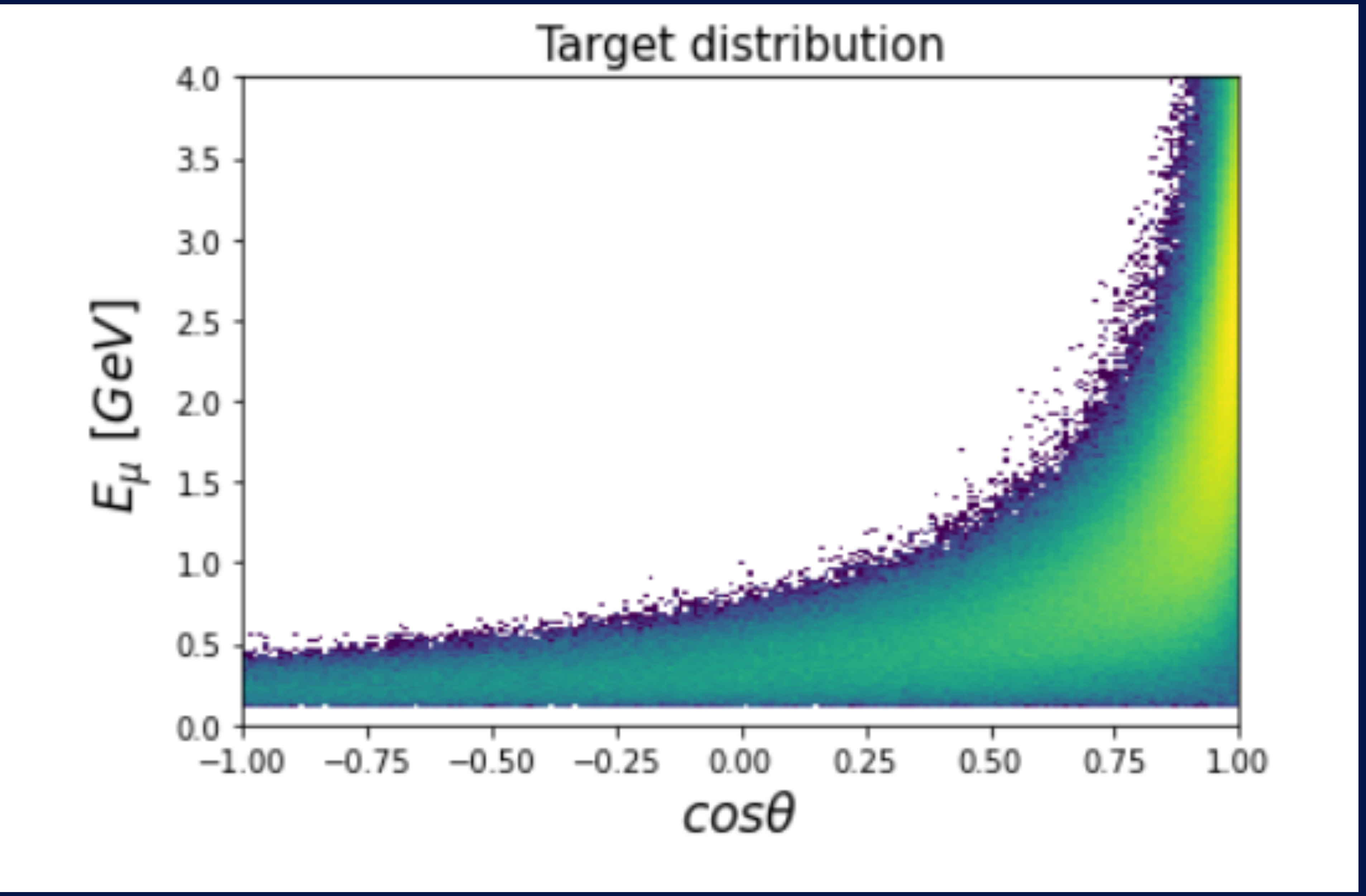
This site provides information on NNPDF for the general public, for physicists, and for PDF users. Among others, a description of our main research tools, user manuals and documentation, talks and publications, including theses, and links to analysis tools. The NNPDF code, including extensive documentation, is available open source. All NNPDF PDF sets are publicly available from LHAPDF.

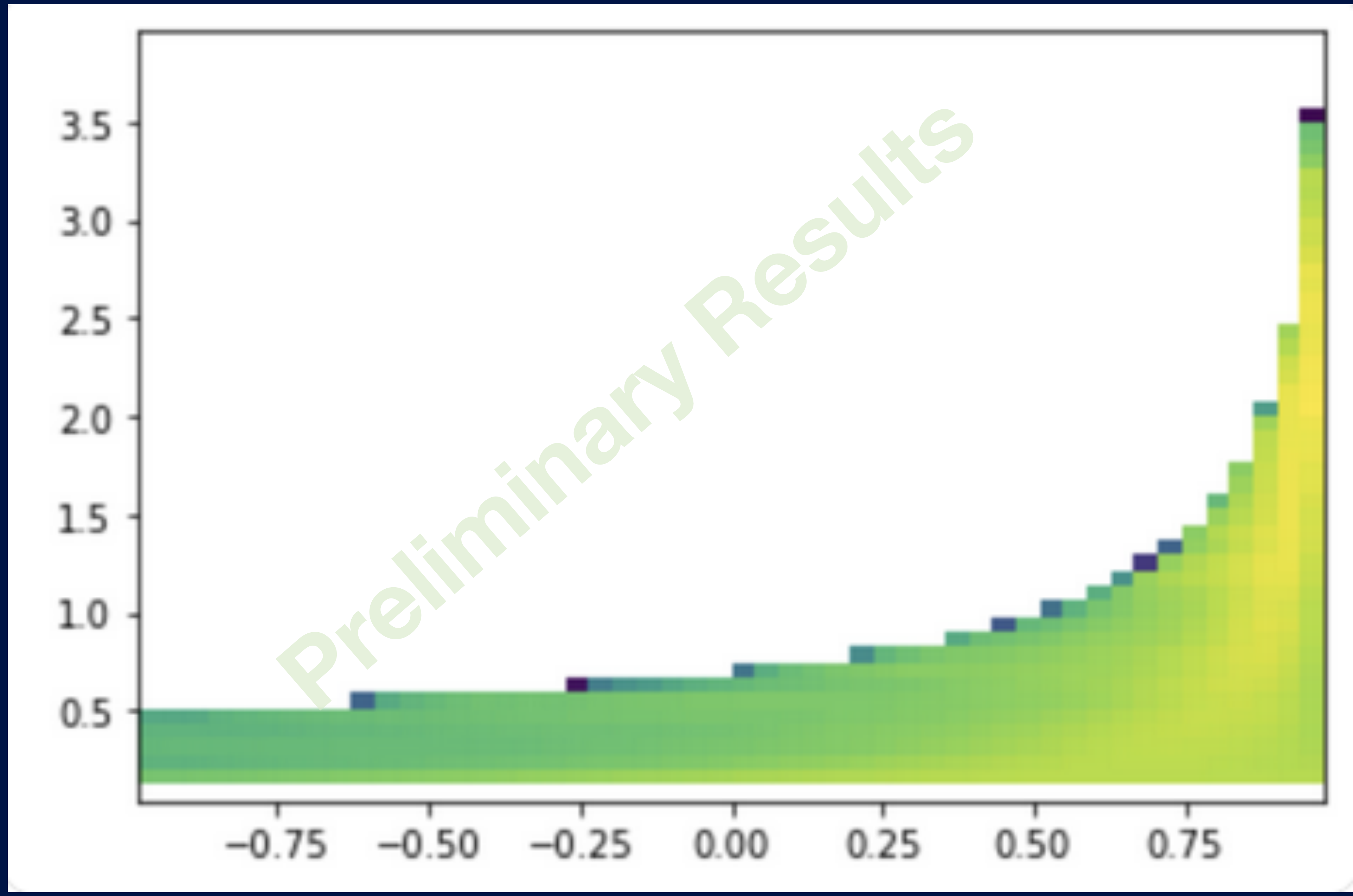
NNPDF determines PDFs using as an unbiased modeling tool Neural Networks, trained using Genetic Algorithms, and used to construct a Monte Carlo representation of PDFs and their uncertainties: a probability distribution in a space of functions

Loss Function

$$\mathcal{L} = \ln\left[\left\langle\left\langle\sum_i K_i(E_\nu, x, Q^2)W_i(x, Q^2)\right\rangle_{\Phi_{ND}}\right\rangle_F\right] - \left\langle \ln\left[\left\langle\sum_i K_i(E_\nu, x, Q^2)W_i(x, Q^2)\right\rangle_{\Phi_{ND}}\right]\right\rangle_{Q_{ND}}$$

Preliminary Results





Uncertainties

Uncertainties

Training and data uncertainties can be calculated simultaneously using **Bootstrap** 🎉🎉

Uncertainties

Training and data uncertainties can be calculated simultaneously using **Bootstrap** 🎉

Near Detector

1. Bootstrap sample from ND data and ND flux
2. Sample over random network initializations

Uncertainties

Training and data uncertainties can be calculated simultaneously using **Bootstrap**

Near Detector

1. Bootstrap sample from ND data and ND flux
2. Sample over random network initializations

Far Detector

1. Bootstrap sample from FD data and ND flux
2. Sample over random initializations of PMNS matrix converting ND to FD flux

Uncertainties

Training and data uncertainties can be calculated simultaneously using **Bootstrap** 🎉

Near Detector

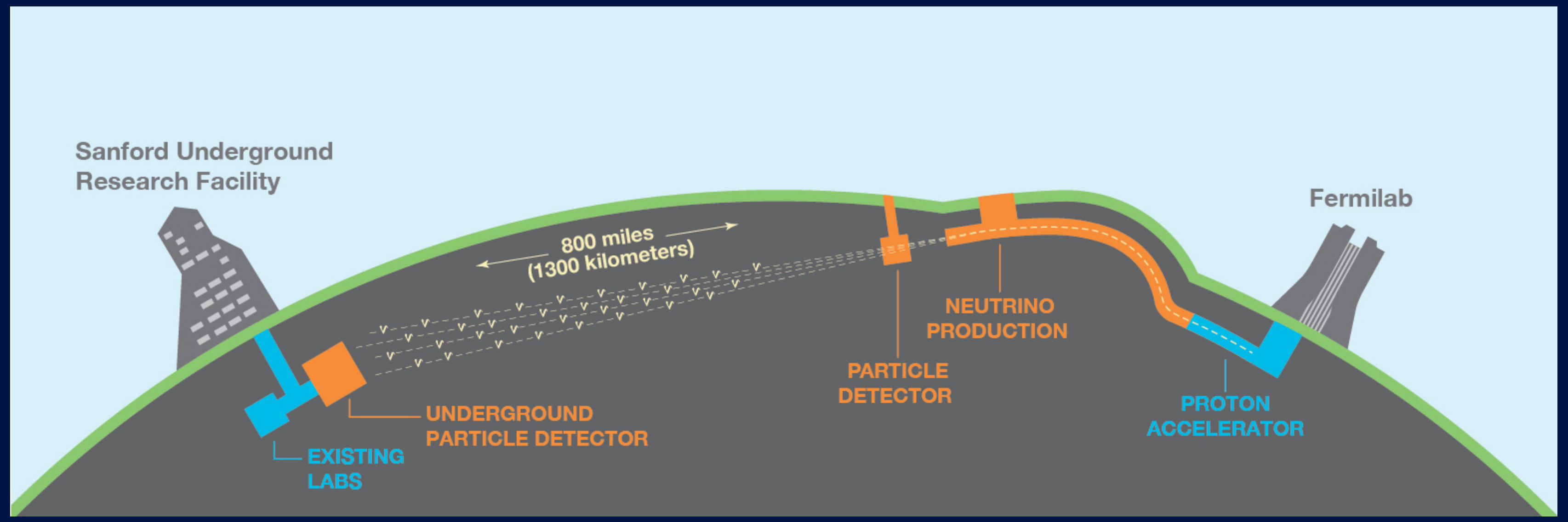
1. Bootstrap sample from ND data and ND flux
2. Sample over random network initializations

Far Detector

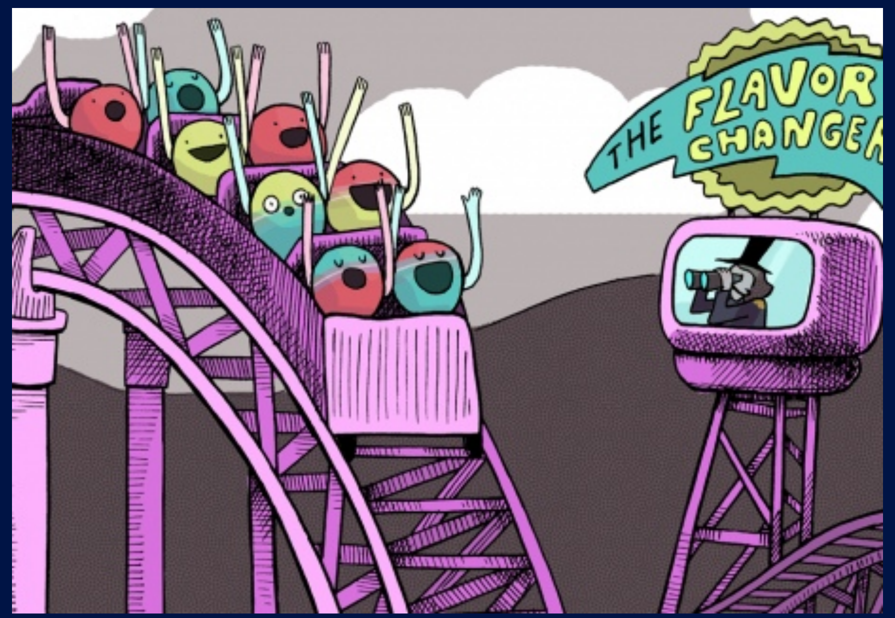
1. Bootstrap sample from FD data and ND flux
2. Sample over random initializations of PMNS matrix converting ND to FD flux

Statistical error from generated/obtained data

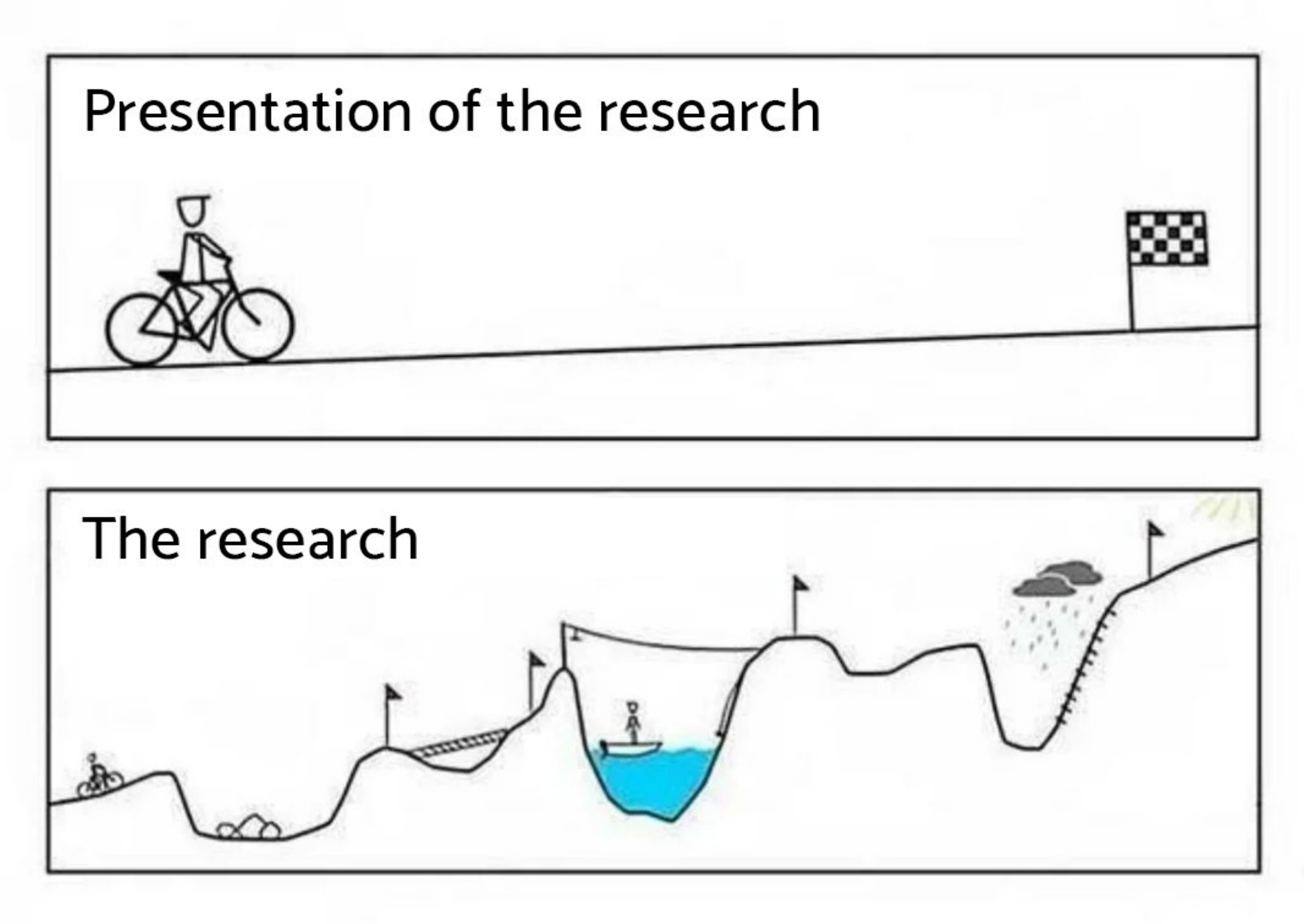
Conclusions



- We have developed a method to perform neutrino oscillation analysis without a microscopic model of the neutrino-nucleus cross section
- Normalizing Flow model of cross section fit to near detector data and neutrino flux used to extract oscillations from far detector data
- Data driven complement to theory driven modeling efforts
- Work in progress: Understand network training uncertainties and incorporate microscopic theory training
- We're still figuring out the best way to do this and especially how to quantify all of the uncertainties in our modeling, and we are open to suggestions

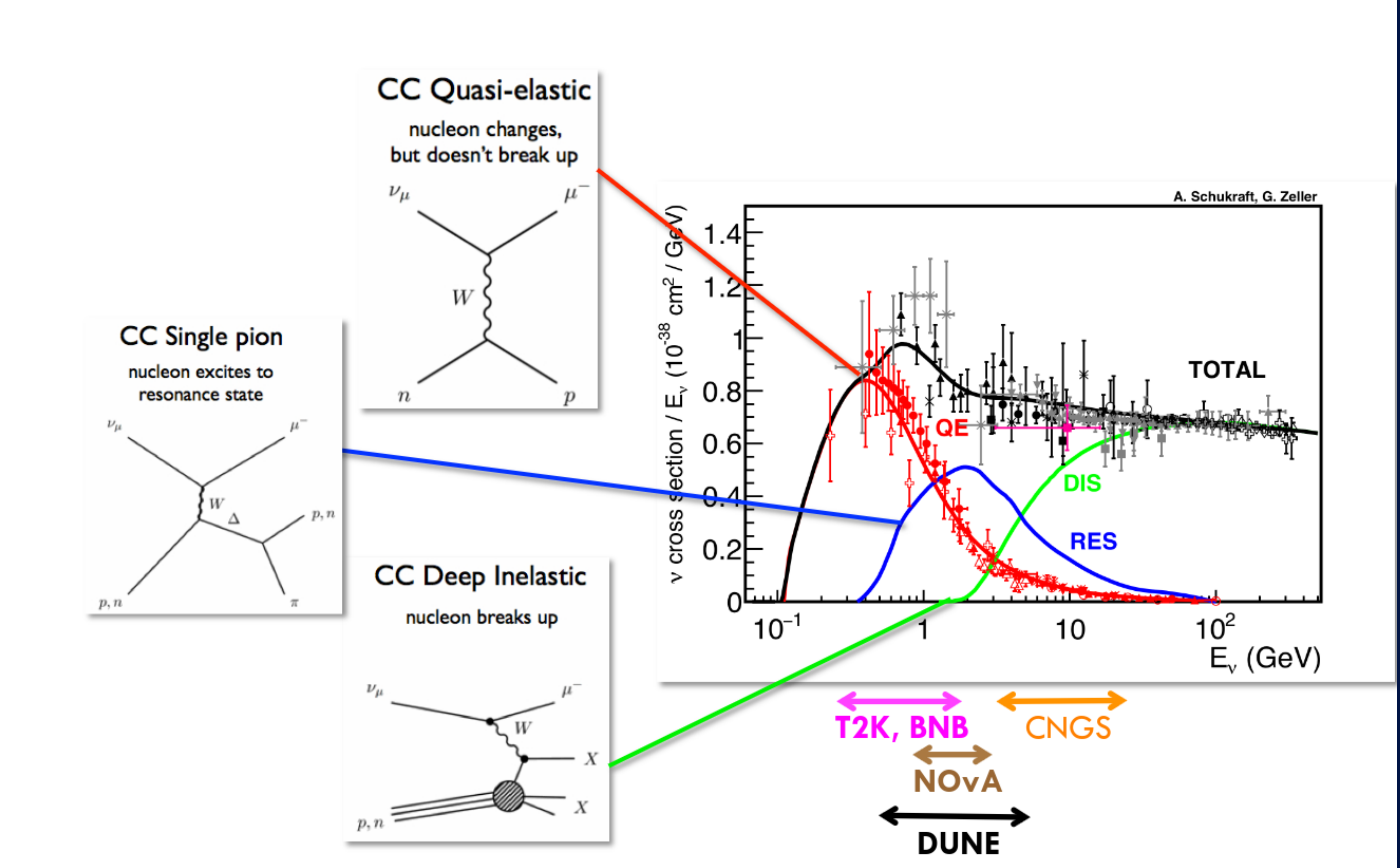


Thanks!



fin.

Backup



S. Zeller, JLAB Workshop, May 2015