Differentiable nuclear deexcitation simulation for low energy neutrino physics

Pablo Barham Alzás[1] and **Radi Radev**[1]
[1]European Organisation for Nuclear Research (CERN)

Low energy ν-N interactions

The modelling of **neutrino-nucleus interactions** plays an important role in present and future neutrino experiments. The accurate simulation of these interactions at **low energies (<100 MeV)** is crucial for the detection and study of **supernova, solar and atmospheric neutrinos** [1].

Reconstruction of the incoming neutrino properties depends on the ability to measure the products from the **deexcitation of the final state nucleus** after the initial neutrino-nucleus scattering reaction. A realistic **nuclear deexcitation model** that can correctly manage the theoretical uncertainties in the process is key to determine the response of a detector to low energy neutrinos.

The Hauser-Feshbach nuclear deexcitation model

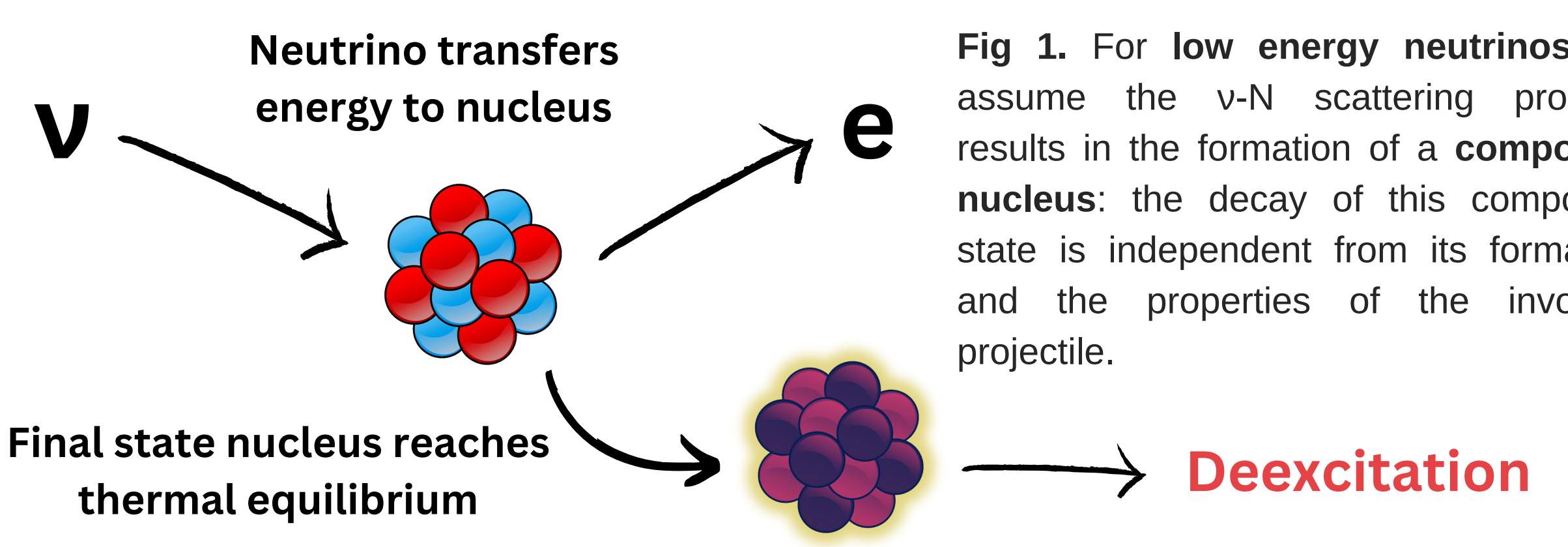
ν
Neutrino transfers energy to nucleus
e
Final state nucleus reaches thermal equilibrium
Deexcitation

Fig 1. For **low energy neutrinos** we assume the ν-N scattering process results in the formation of a **compound nucleus**: the decay of this compound state is independent from its formation and the properties of the involved projectile.

The **Hauser-Feshbach statistical emission model** [2] is used by event generators like MARLEY [3], the standard generator for low energy nuclear physics in Liquid Argon TPCs for experiments like DUNE. It exploits the small distance between highly excited nuclear states to derive the decay rate at a given nuclear energy using statistical methods, by averaging over many states of similar energies.

The **differential decay width** to a given state by emitting a particular nuclear fragment can be written as:

$$\frac{d\Gamma_a}{dE_f}=\frac{1}{2\pi\rho_i\left(E_i,\alpha\right)}\sum_{\alpha'}\overset{\text{Transition strength}}{T_{\alpha'}(\varepsilon)}\underset{\text{Event density}}{\rho_f\left(E_f,\alpha'\right)}$$

where sums are over the different quantum numbers α (EM transition type and multipolarity for γ-rays or angular momentum for massive fragments, and isospin and total angular momenta of the initial and final nuclear states).

Results

Sampling and inference

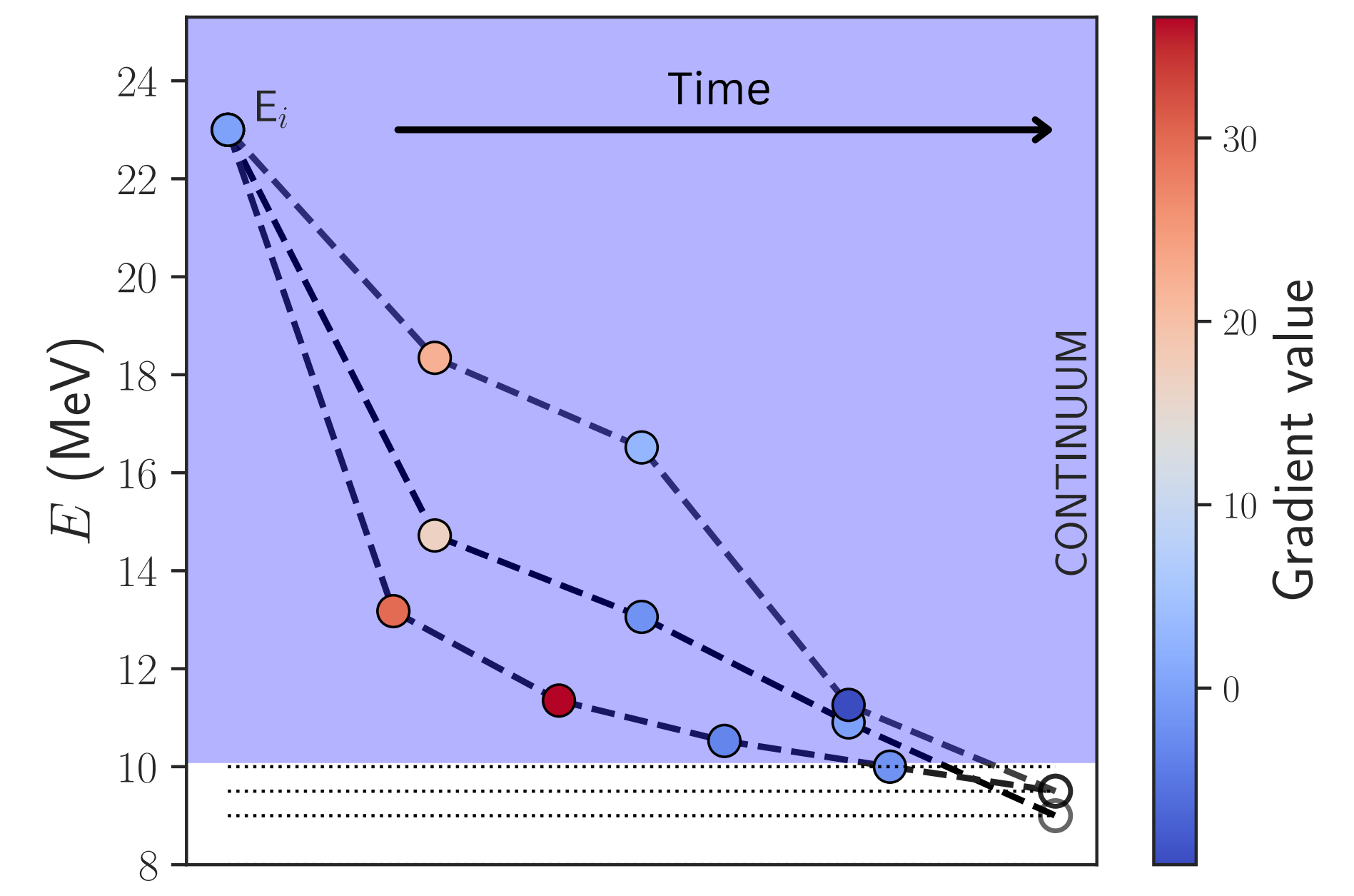

Fig 2. Three deexcitation paths in the continuum. Each point represents a sampled nuclear level, coloured according to the gradient w.r.t. the λ parameter. This parameter controls how shifted into the continuum the event density function is, modelled as a backshifted Fermi gas.

- Our sampling procedure implemented in JAX is **vectorised**, **runs on the GPU** and automatically yields the gradients of the outputs.
- Gradients checked against **finite differences** and other stochastic gradient estimators like the **score function** method.

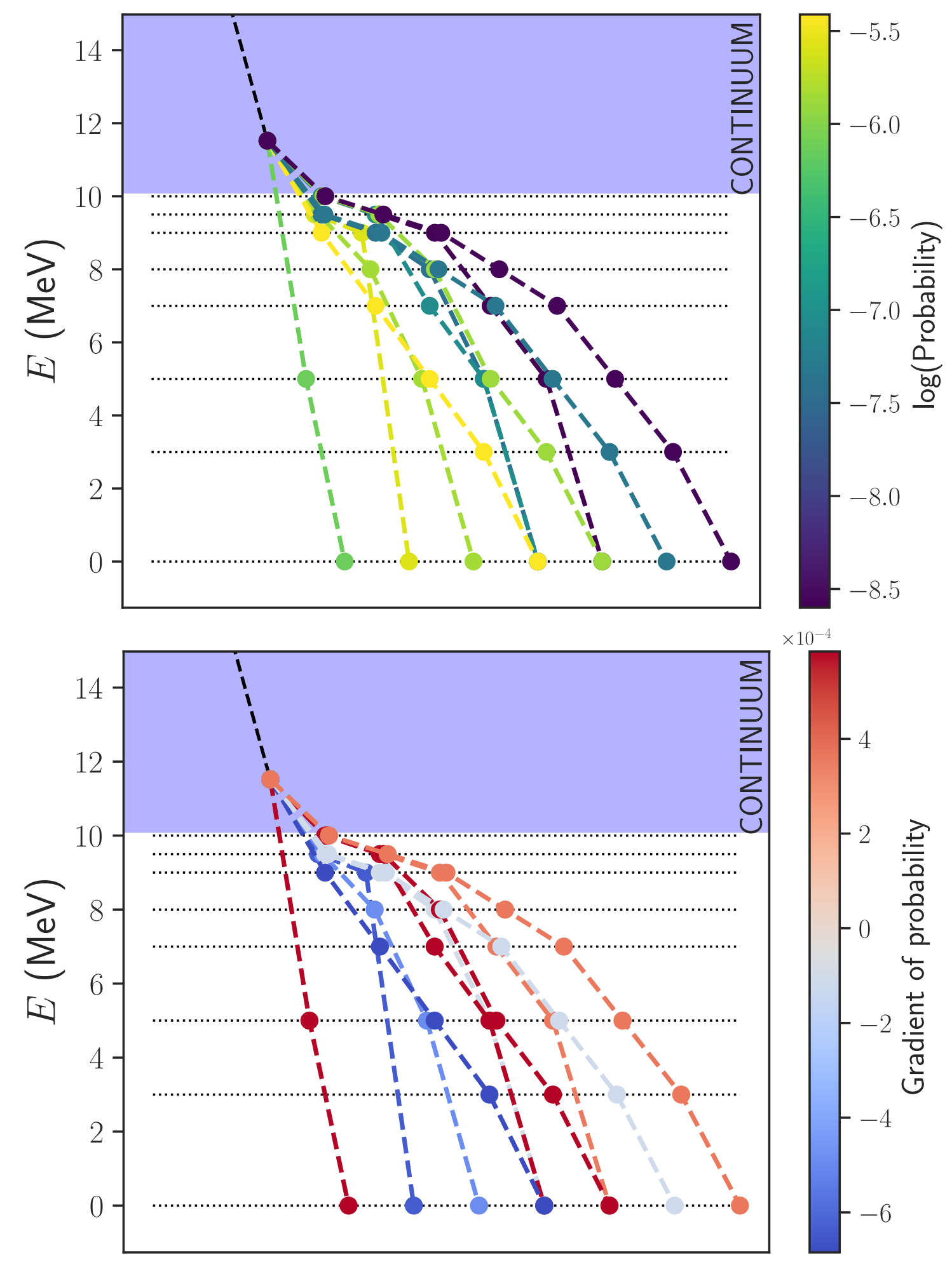

Fig 3. A subsample of the discrete deexcitation "tree" for a given continuum path. **Top:** paths coloured by the path probability. **Bottom:** paths coloured by the gradient of the probability w.r.t. model parameter α, that controls the frequency of the "resonance peaks" in the transition strength function.

- Using a pathwise estimator allows us to take derivatives of and **"move" individual events**.
- We can **estimate the uncertainty** on the expected values of an observable given a set of parameters by **running the simulation only on the central values** and performing an expansion around them.

Model fitting via gradient descent

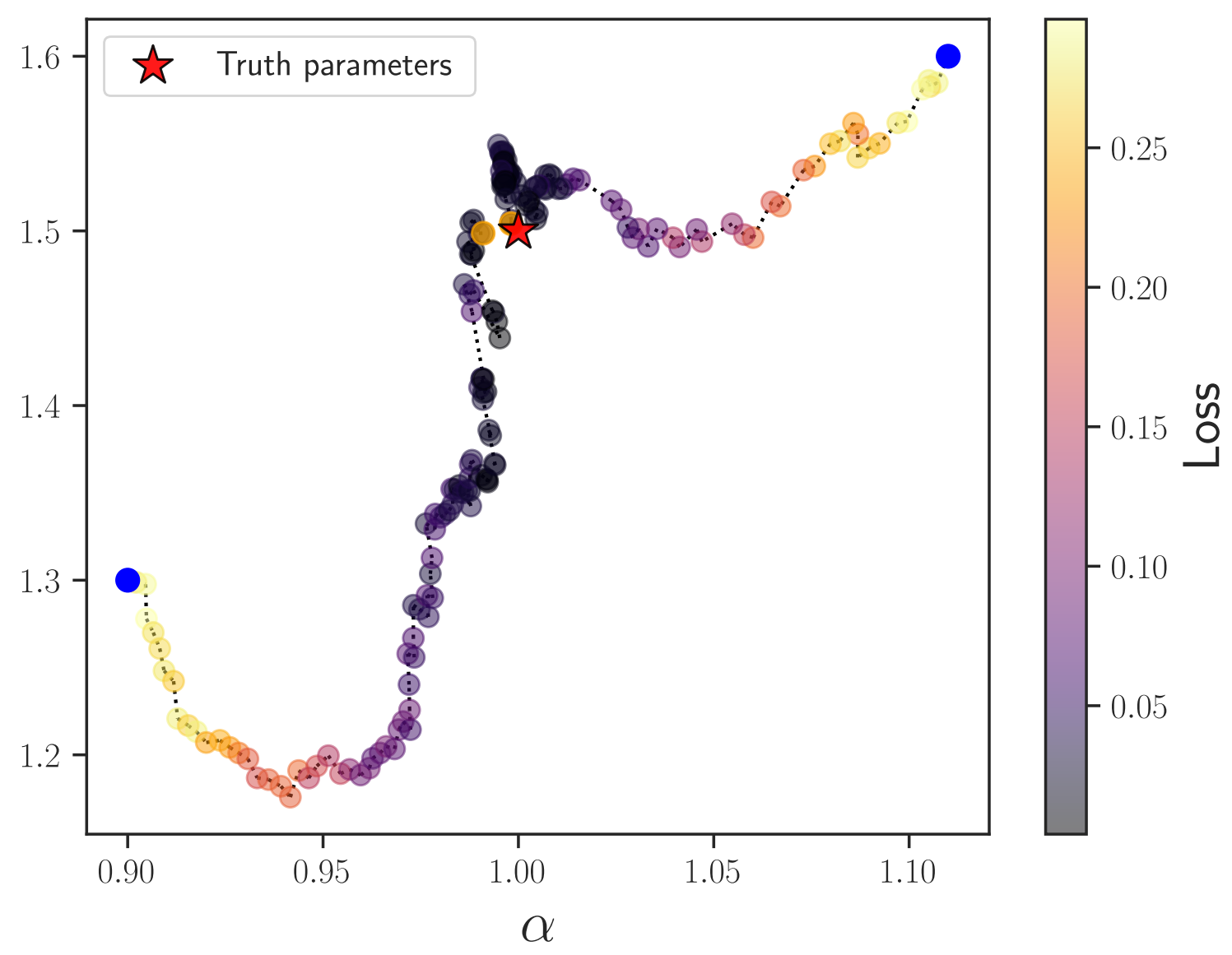

Fig 4. Two gradient descent iterations converging in the α-λ plane, coloured by loss.

- At each point, generate new sample and **compute the loss and its gradient** via Maximum Mean Discrepancy **(MMD)**.
- Perform 10 extra iterations at the end where we **double the sample size to reduce the variance of the estimator.**

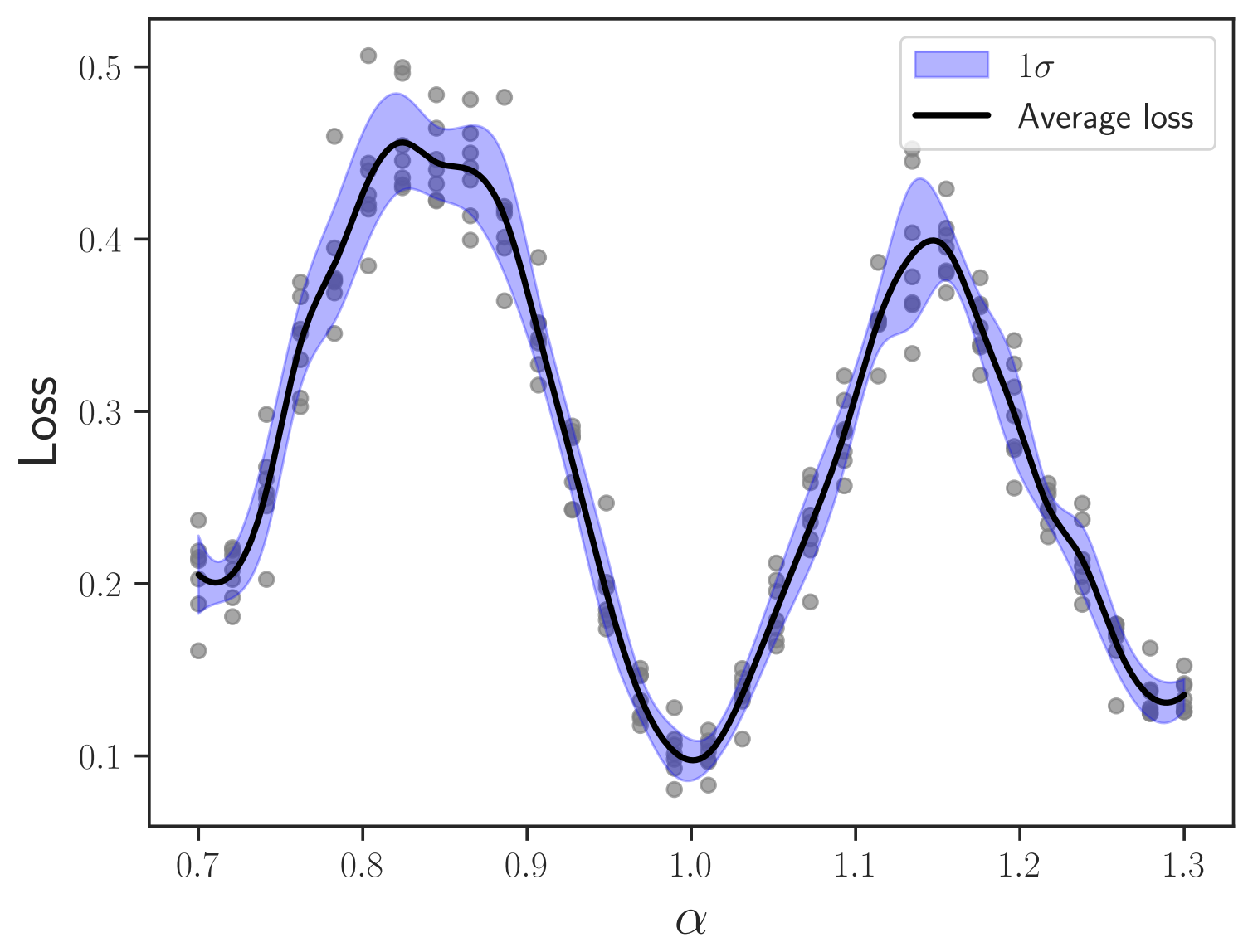

Fig 5. Loss function for different values of the α parameter. Several samples were generated at each value to estimate the loss. While peaked at "thruth" α=1, the variance in the estimator can occasionally produce a gradient in the wrong direction, inducing instability.

- We find that the procedure works well in general but suffers from occasional **instability**, especially for **smaller sample sizes**.
- Computing the probability and gradients for **only a fraction of the total possible discrete paths** yields an estimate accurate enough for gradient descent, **promising scalability**.

Differentiable simulation

Autodiff frameworks like **JAX** [4] give us the tools to build **differentiable simulators** by computing exact gradients of the outputs with respect to the model parameters. **Advantages** include:

- Parameter tuning via gradient descent.
- Efficient estimation of uncertainties.
- Integration with existing Machine Learning tools.

We implement a differentiable toy version of the Hauser-Feshbach model where we focus on the dominant mode of deexcitation: γ-ray emission.

Sampling the continuum

Starting with an initial nuclear energy E_0 transferred by the neutrino in the interaction, we sample the next excitation energy via the sampling path:

$$\hat{E}_1=g(\hat{\epsilon};E_0,\boldsymbol{\theta});\quad \hat{\epsilon}\sim\mathcal{U}[0,1]$$

where **θ** are the model parameters and **g** is the **inverse CDF** of the differential decay width **modified to account for the probability of the decay to a discrete nuclear state**. We continue sampling energies sequentially until we reach a discrete state.

Estimating the gradients

We estimate the gradients of expected values ("stochastic gradients") of a function **f** of the outputs with respect to the model parameters via a pathwise gradient estimator [5]:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{p(E_i;E_{i-1},\boldsymbol{\theta})}[f(E_i)]=\mathbb{E}_{p(E_i;E_{i-1},\boldsymbol{\theta})}[\nabla_{E_i}f(E_i)\nabla_{\boldsymbol{\theta}}E_i]$$

As taking a derivative of the inverse of a function is hard, we can **decouple the sampling and the gradient computation** obtaining the gradient directly from the CDF:

$$\nabla_{\boldsymbol{\theta}}E_i=-\left(\nabla_{E_i}g^{-1}(E_i;E_{i-1},\boldsymbol{\theta})\right)^{-1}\nabla_{\boldsymbol{\theta}}g^{-1}(E_i;E_{i-1},\boldsymbol{\theta})$$

where we must take care to include the dependence of the previous energy with **θ**.

Gradients of the discrete levels

Once the deexcitation process reaches a discrete level, we no longer have a smooth dependence of the process energies on the model parameters. To avoid the problem of taking derivatives of a discrete variable, **we introduce a novel approach sampling the whole discrete tree**: we compute the probability of each discrete path and its derivative with respect to **θ**. Then, we can estimate the elements of the gradient as

$$[E_i]=\sum_{\text{paths}}p(\text{path})\cdot E_{i,\text{path}}\longrightarrow\begin{cases}\nabla_{[E_i]}f([E_i])=\sum_{\text{paths}}p(\text{path})\cdot\nabla_{E_i}f(E_{i,\text{path}})\\ \nabla_{\boldsymbol{\theta}}[E_i]=\sum_{\text{paths}}\nabla_{\boldsymbol{\theta}}p(\text{path})\cdot E_{i,\text{path}}\end{cases}$$

Future work and conclusions

COMING SOON

- More complete implementation of the Hauser-Feshbach deexcitation model.
- Work on stability and speed of the gradient descent algorithm.
- Event moving with higher order derivatives.

- Inclusion of the neutrino-nucleus interaction physics.
- Integration in a fully differentiable simulation chain, from generation to reconstruction.

FOR THE FUTURE

Conclusions

- We have demonstrated the feasibility of a fully differentiable implementation of the key components of a nuclear deexcitation model.
- Sampling and gradient estimation is fast and robust.
- Model fitting via gradient descent still needs work in stability.
- There is much work to do but we are confident in the scaling of our method!

[1] Low-Energy Physics in Neutrino LArTPCs, Contributed Paper to Snowmass 2021, arXiv:2203.00740 [physics.ins-det].
[2] The inelastic scattering of neutrons, W. Hauser and H. Feshbach, Physical Review 87, 366 (1952).
[3] S. Gardiner, Simulating low-energy neutrino interactions with MARLEY, Comput. Phys. Commun. 269.
[4] JAX: High-Performance array computing, https://jax.readthedocs.io/.
[5] Monte Carlo Gradient Estimation in Machine Learning, S. Mohamed et al., Journal of Machine Learning Research 21 (2020) 1-62.

Neutrino Group

CHIPMUNC
GitHub Repo

γPhys 2023

King's College London
18-20 December 2023