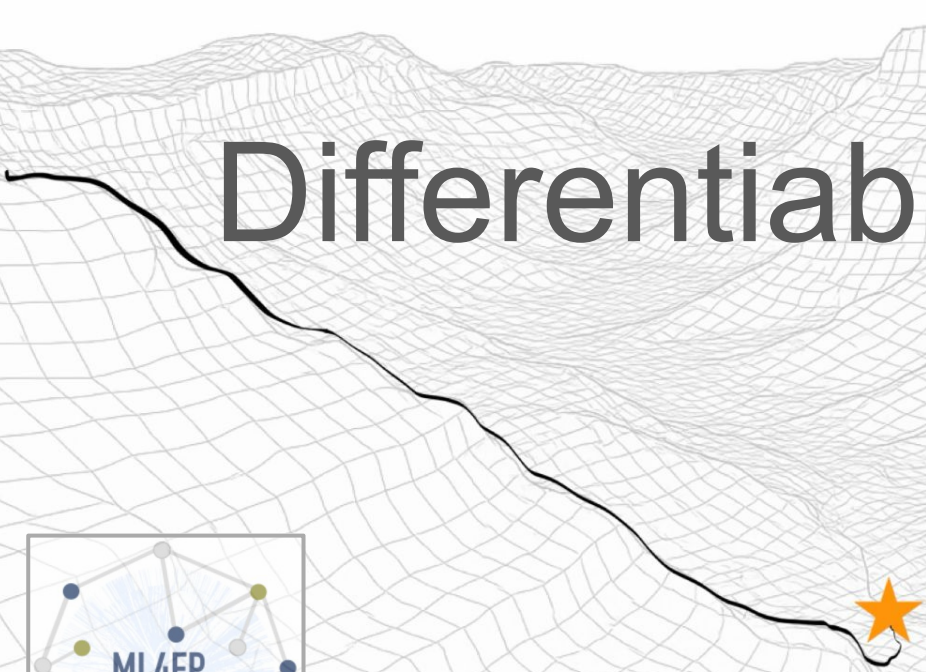
SLAC NATIONAL ACCELERATOR LABORATORY

Differentiable modeling



Yifan Chen

ML4FP @ Georgia Tech

02 June 2026

Outline

Lecture:

- What is differentiable modeling?
- What can it do in HEP?
- Making differentiable modeling
- Applications in HEP
- Tips for real differentiable modeling

Tutorial:

- Fit parameters with differentiable modeling
- Chain a differentiable physics model with neural network
- Tricks for sharp edges

What is differentiable modeling?

Model with gradients

$$y = f(x; \theta)$$

output input model parameters

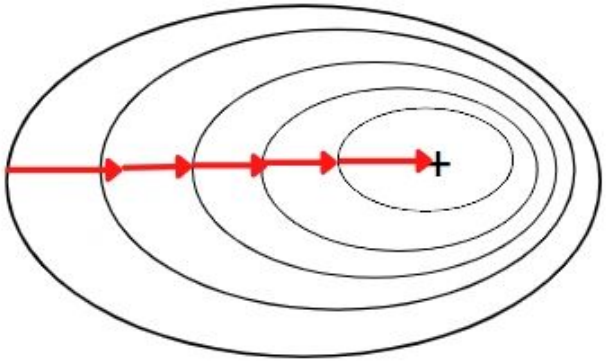
Gradients $\dfrac{\partial y}{\partial x}$ $\dfrac{\partial y}{\partial \theta}$

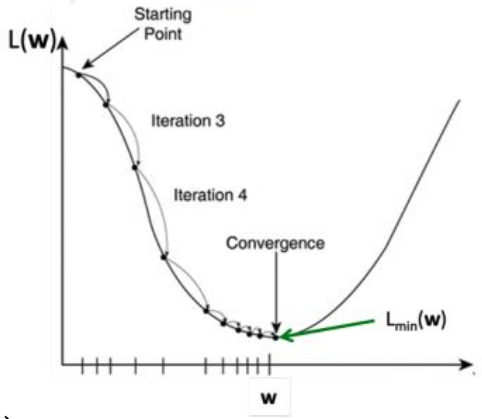
A lot of ML models are differentiable

```
# Multi-layer perceptron
mlp = MLP(n_hidden=n_hidden, hidden_dim=hidden_dim)

# Optimizer
optimizer = torch.optim.Adam(mlp.parameters(), lr=lr)

# Mean squared error
loss_fn = torch.nn.MSELoss()

losses = []

for _ in range(n_epochs):

    # Shuffle data
    idxs = np.random.permutation(len(norm_x))

    # Make predictions
    out = mlp(norm_x[idxs])

    # Calculate loss
    loss = loss_fn(out, norm_y[idxs])

    # Zero out gradients
    optimizer.zero_grad()

    # Compute gradients
    loss.backward()

    # Update parameters
    optimizer.step()
```

$$\nabla_w L(w^{(t)})$$

$$w^{(t+1)} = w^{(t)} - \eta \cdot \nabla_w L(w^{(t)})$$ For gradient descent

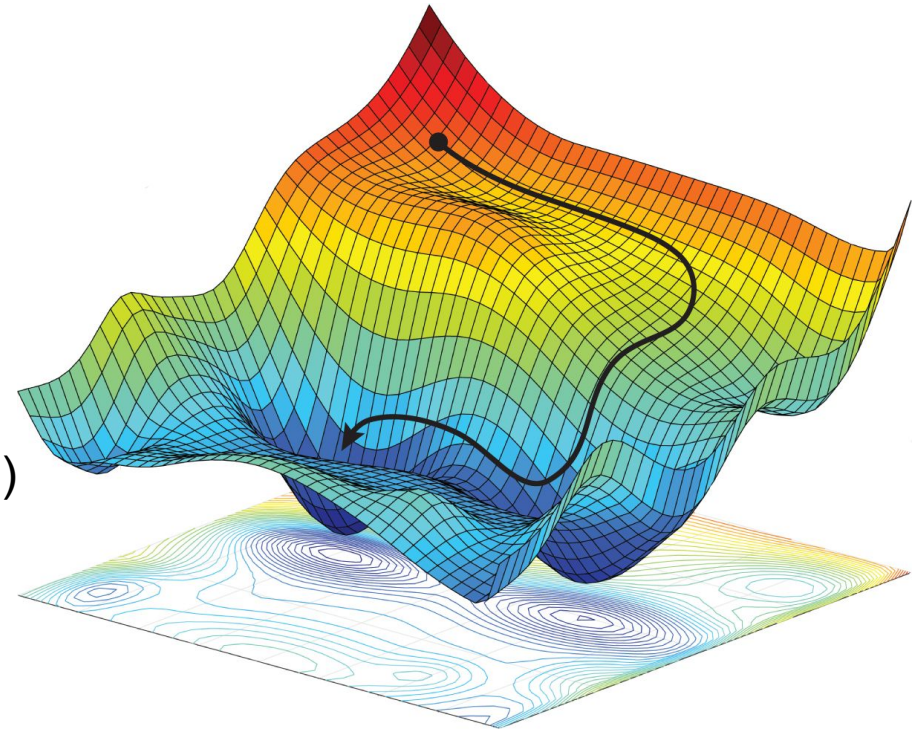




Gradients

For optimizing the models

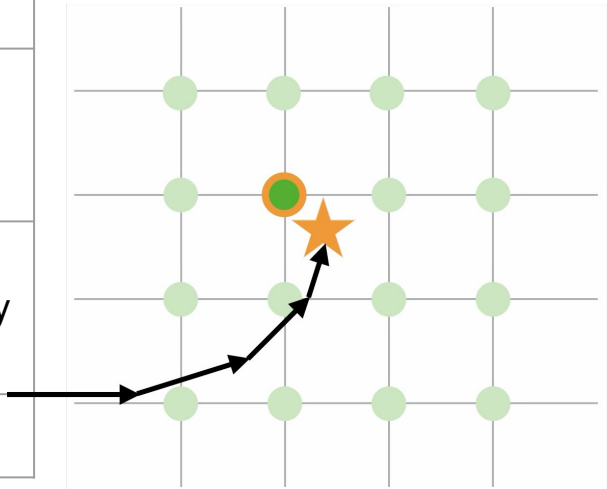
- Scalable wrt. parameters (dimensionality)
- Efficient with principled direction (gradients)
- Gradients encode local geometry (second order describes curvatures)
- Provide relevance of parameters



Compared to gradient-free methods

For optimizing the models

	Gradient-based	**Gradient-free**
Information for optimization	Loss + gradient direction and magnitude for all parameters	Loss at the evaluation points
Scalability	Insensitive to number of parameters, scale with resolution	Can explode with the parameter dimensionality and search resolution
Requirement	Differentiable pipeline	Model evaluation

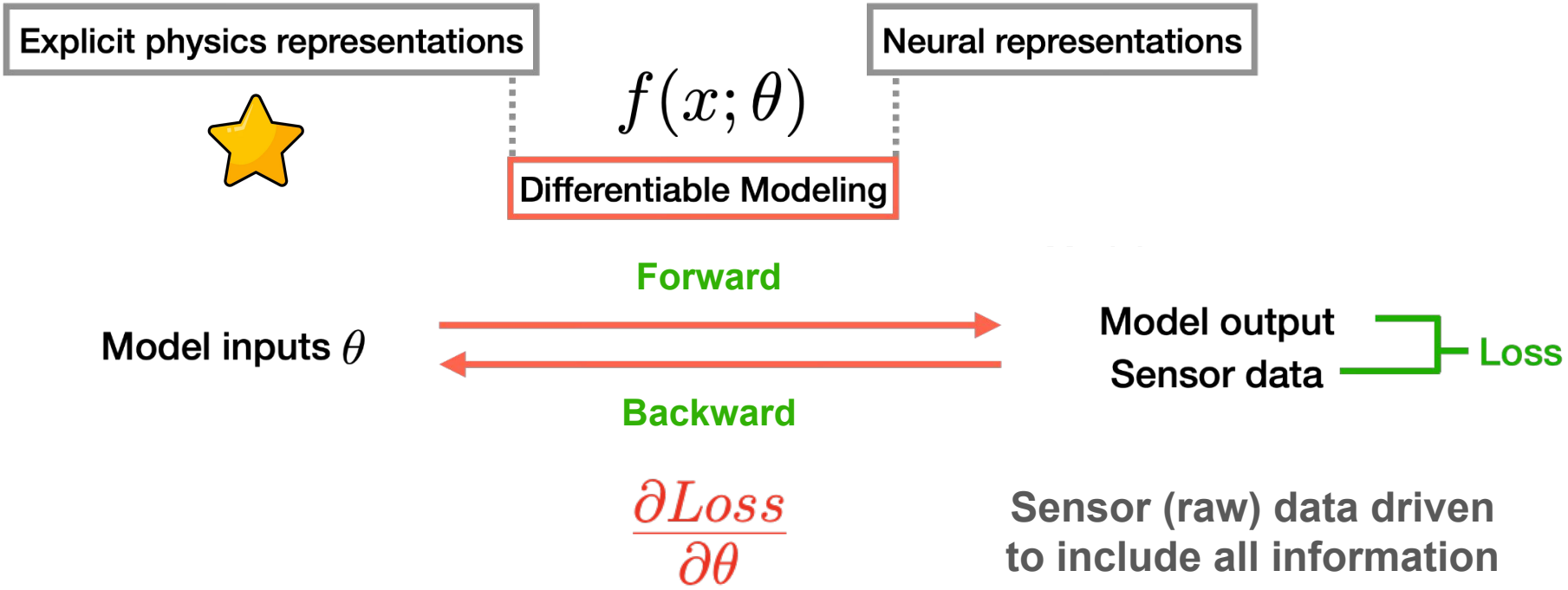


Examples of gradient-free optimization methods:

Grid search, random search, MCMC, population-based evolutionary genetic algorithms, Nelder-Mead method (simplex method), Bayesian optimization, reinforcement learning

Differentiable modeling for inverse problems

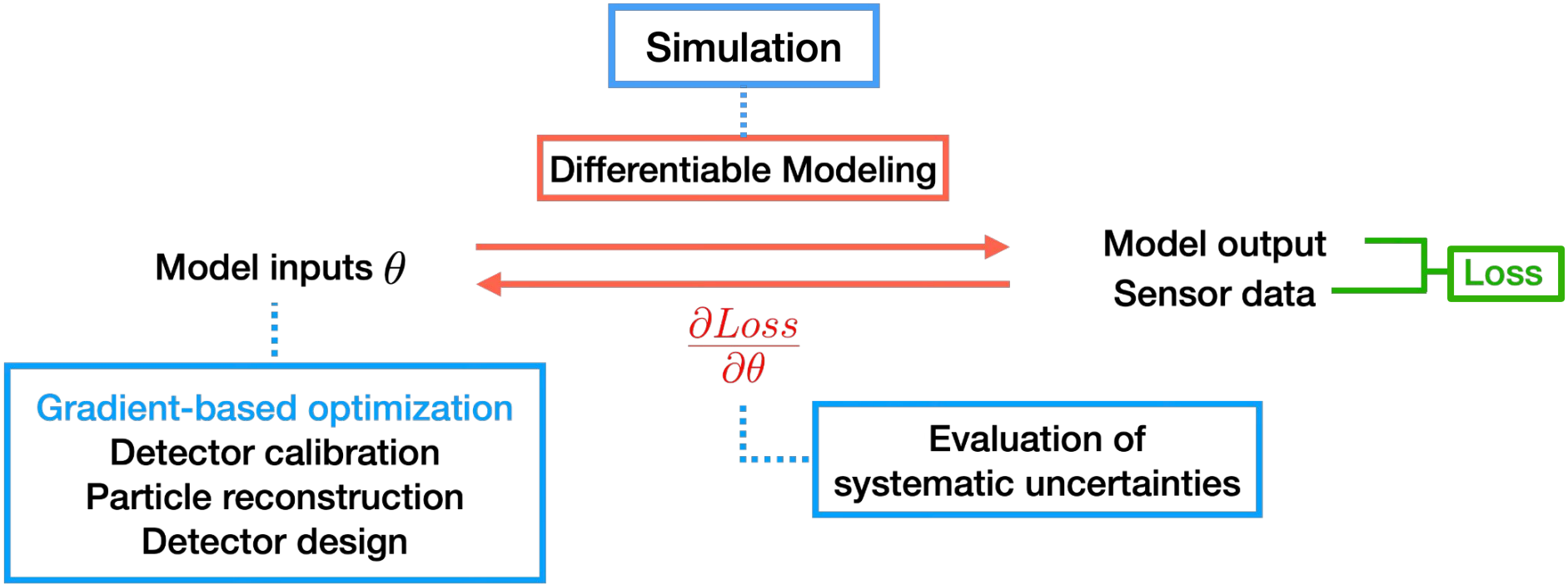
Gradient-based optimization to improve physics knowledge



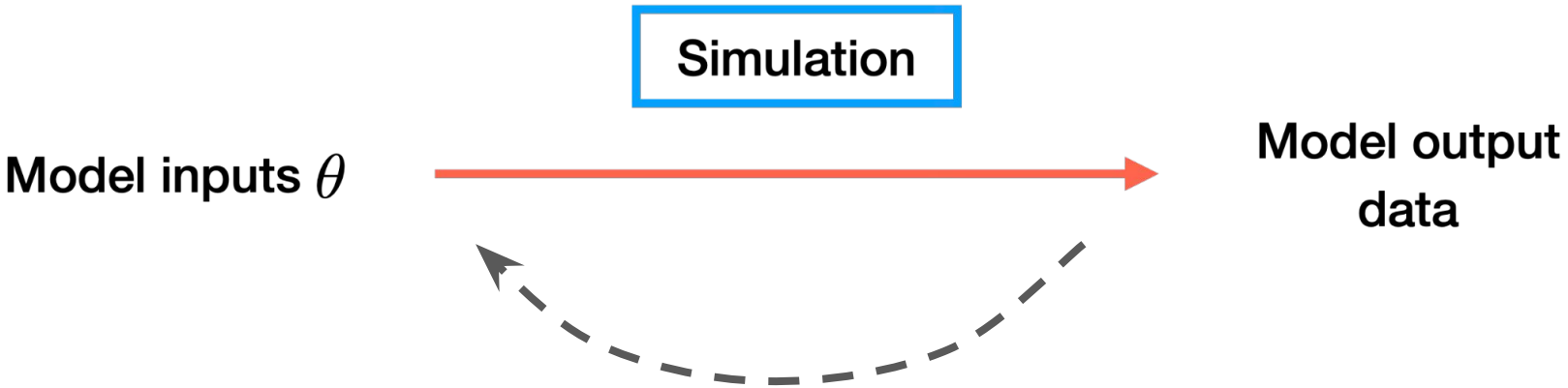
What can differentiable modeling do in HEP?

Differentiable modeling in HEP

In-situ models, no workaround



Examples of modeling optimization in HEP



Examples of workaround in HEP: Generator tuning
GENIE spline (neutrino/electron interaction), Professor polynomial (PYTHIA)

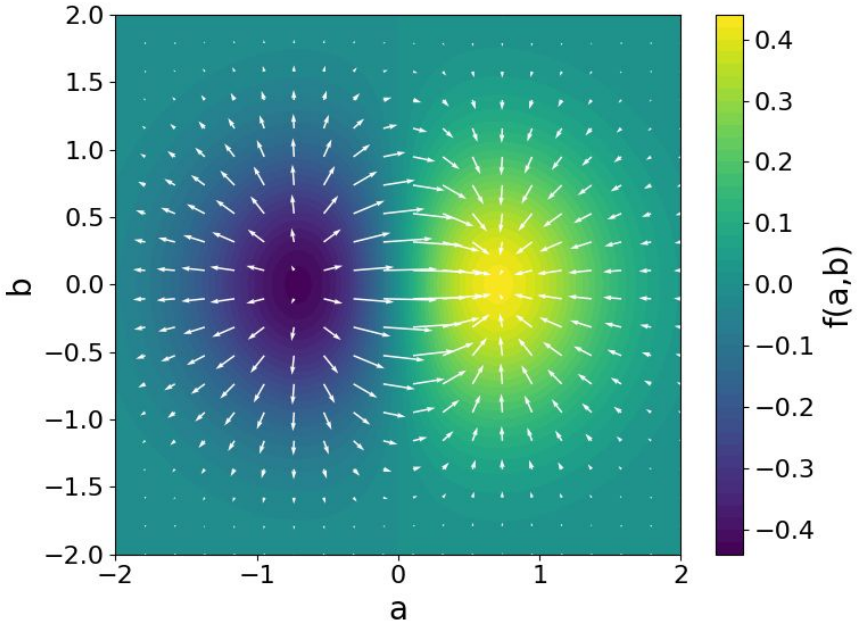
Making differentiable modeling!

Three ways to achieve gradients

Numerical differentiation

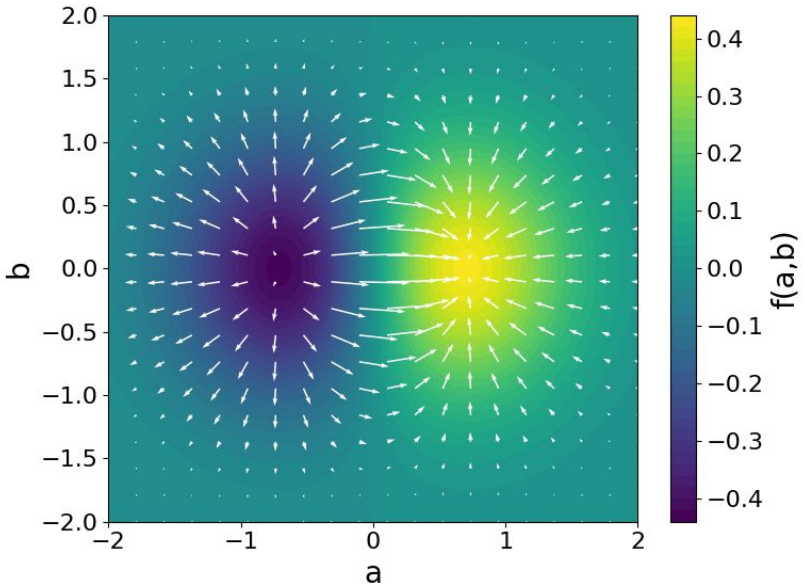
Symbolic differentiation

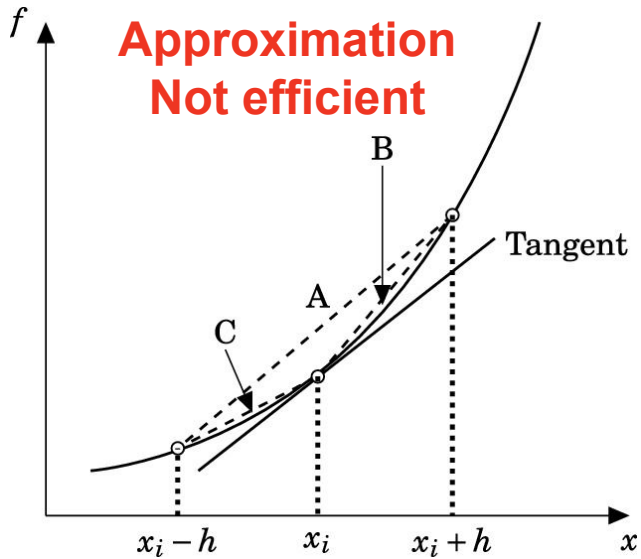
Automatic differentiation



Numerical differentiation

$$\frac{\partial f(a,b)}{\partial a} \approx \lim_{h \to 0} \frac{f(a+h,b) - f(a,b)}{h}$$





Symbolic differentiation

$$f(a,b) = a \cdot e^{-(a^2+b^2)}$$

$$\frac{\partial f}{\partial a} = e^{-(a^2+b^2)}(1-2a^2)$$

$$\frac{\partial f}{\partial b} = -2ab\, e^{-(a^2+b^2)}$$

- Your magic brain
- Pen and paper
- Tools such as Mathematica, SymPy

Exact gradients
Requires closed form expressions
(No control flows, in most physics simulation)

Automatic differentiation (AD)

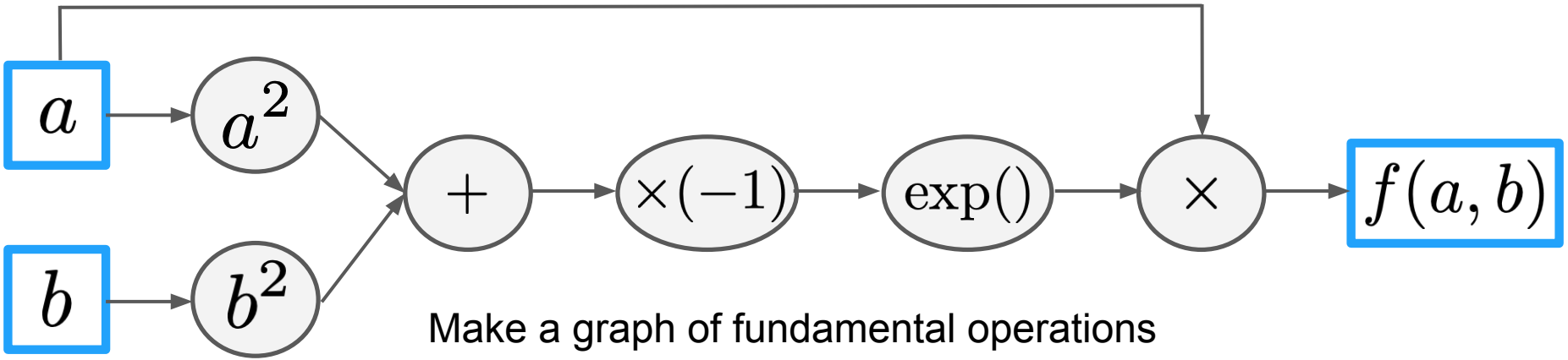
$$f(a,b) = a \cdot e^{-(a^2+b^2)}$$

Applicable to arbitrary computer programs*

Exact gradients

Scales well

(forward + backward) ~ 2 x forward

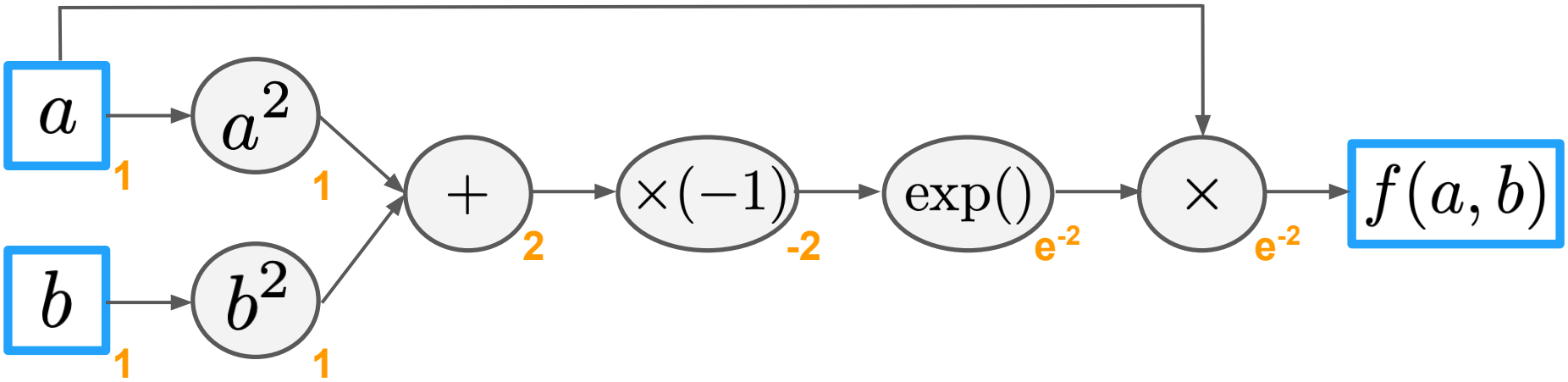


Make a graph of fundamental operations

AD: Primals

$$f(a,b) = a \cdot e^{-(a^2+b^2)}$$

A computational graph made of fundamental operations
Primals are the set of intermediate function values

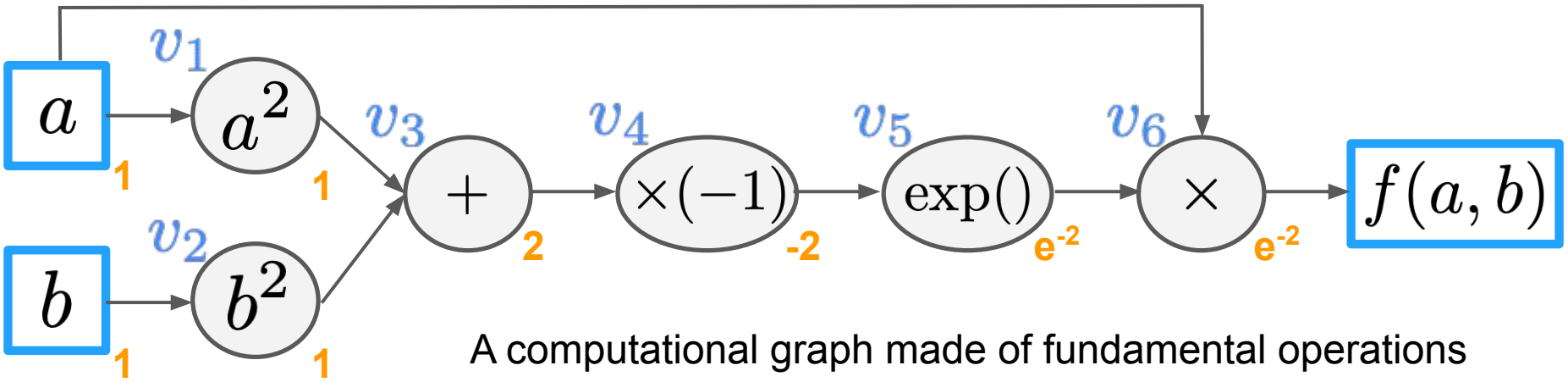


AD: Chain Rule!

$$f(a,b) = a \cdot e^{-(a^2+b^2)}$$

$$\frac{\partial f(a,b)}{\partial a} = \frac{\partial v_6}{\partial a} + \frac{\partial v_6}{\partial v_5}\frac{\partial v_5}{\partial v_4}\frac{\partial v_4}{\partial v_3}\frac{\partial v_3}{\partial v_1}\frac{\partial v_1}{\partial a}$$

$$\frac{\partial f(a,b)}{\partial b} = \frac{\partial v_6}{\partial v_5}\frac{\partial v_5}{\partial v_4}\frac{\partial v_4}{\partial v_3}\frac{\partial v_3}{\partial v_2}\frac{\partial v_2}{\partial b}$$



A computational graph made of fundamental operations

No need of close form expressions

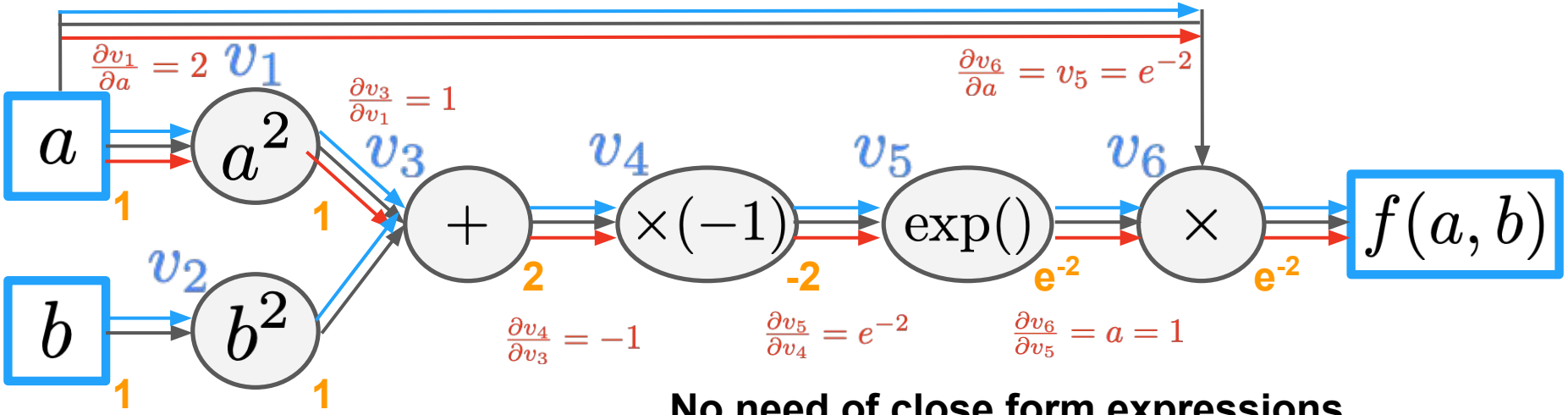
AD: Forward mode

$$f(a,b) = a \cdot e^{-(a^2+b^2)}$$

$$\frac{\partial f(a,b)}{\partial a} = \frac{\partial v_6}{\partial a} + \frac{\partial v_6}{\partial v_5}\frac{\partial v_5}{\partial v_4}\frac{\partial v_4}{\partial v_3}\frac{\partial v_3}{\partial v_1}\frac{\partial v_1}{\partial a}$$

Not to be confused with differentiable forward model

Compute primals and derivatives
in a single forward pass per variable



No need of close form expressions.
Take derivative of the elementary operations

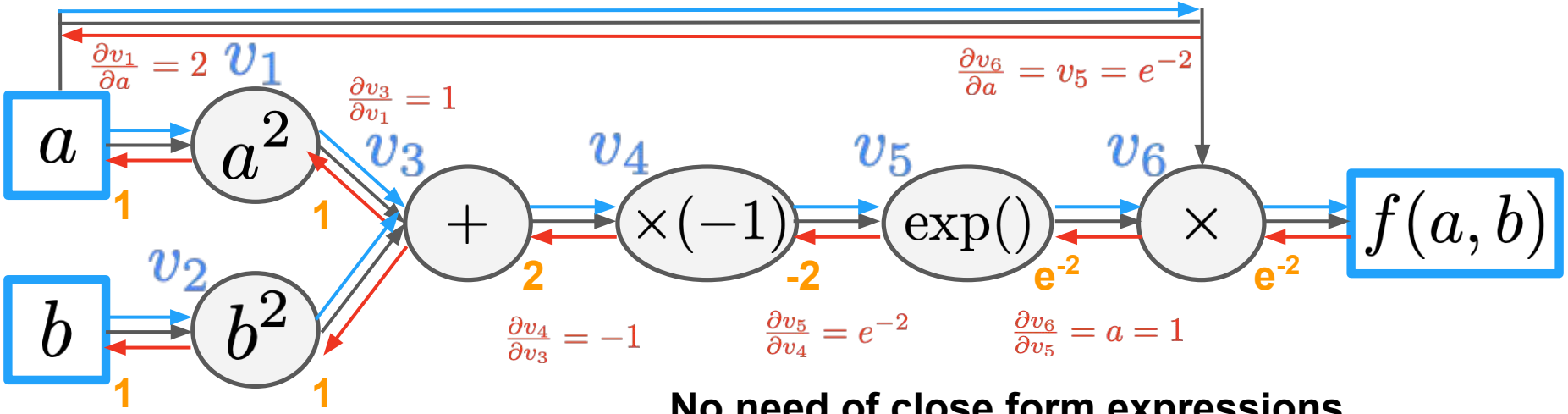
AD: Reverse mode

$$f(a,b) = a \cdot e^{-(a^2+b^2)}$$

$$\frac{\partial f(a,b)}{\partial a} = \frac{\partial v_6}{\partial a} + \frac{\partial v_6}{\partial v_5}\frac{\partial v_5}{\partial v_4}\frac{\partial v_4}{\partial v_3}\frac{\partial v_3}{\partial v_1}\frac{\partial v_1}{\partial a}$$

Not to be confused with differentiable backward model

Compute primals → and derivatives ←
per output



No need of close form expressions.
Take derivative of the elementary operations

AD: Forward mode vs. Reverse mode

$$\mathbf{f}(\theta) : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

Forward mode

$$\frac{d\mathbf{f}(\theta)}{d\theta} = \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} & \cdots & \frac{\partial f_1}{\partial \theta_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial \theta_1} & \cdots & \frac{\partial f_M}{\partial \theta_N} \end{pmatrix}$$

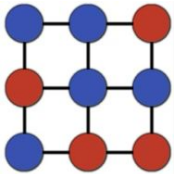
Column of Jacobian matrix
Efficient with large M and small N

Reverse mode

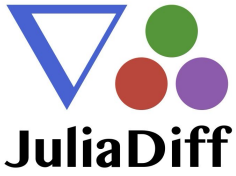
$$\frac{d\mathbf{f}(\theta)}{d\theta} = \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} & \cdots & \frac{\partial f_1}{\partial \theta_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial \theta_1} & \cdots & \frac{\partial f_M}{\partial \theta_N} \end{pmatrix}$$

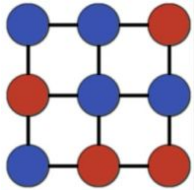
Row of Jacobian matrix
Efficient with small M and large N

AD tools (Incomplete list)



ADOL-C





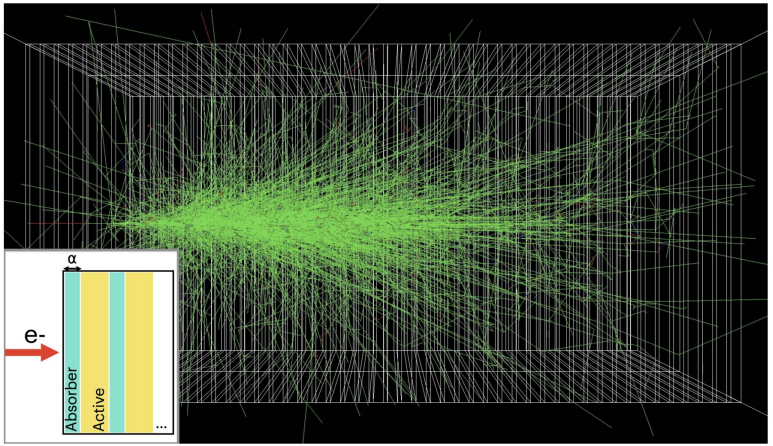
Applications of differentiable modeling in HEP

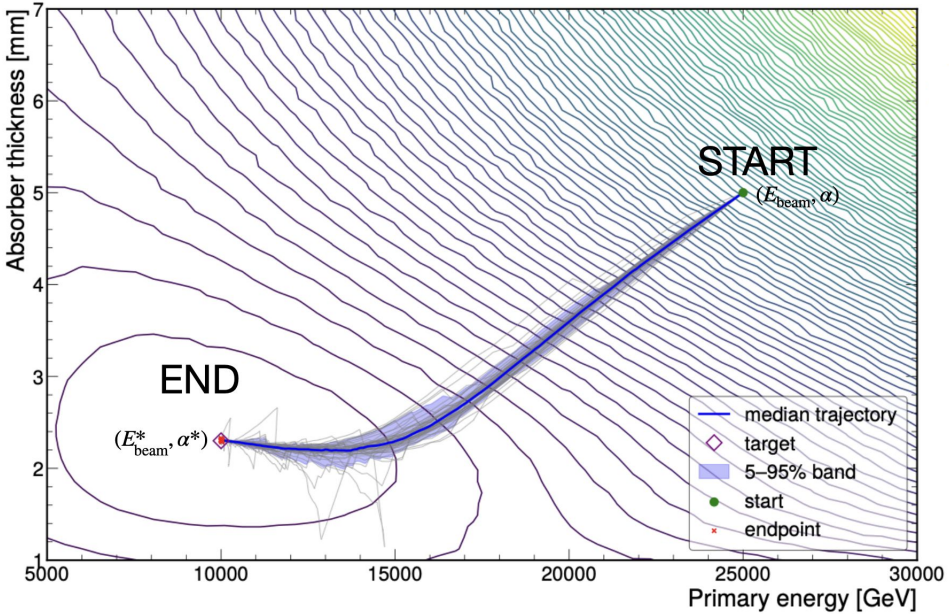
Differentiable Geant4

Detector design study with sampling calorimeter
Focusing on EM showers

arXiv:2605.06779
Comput.Phys.Commun. 309 (2025)
arXiv:2407.02966
arXiv:2308.16680

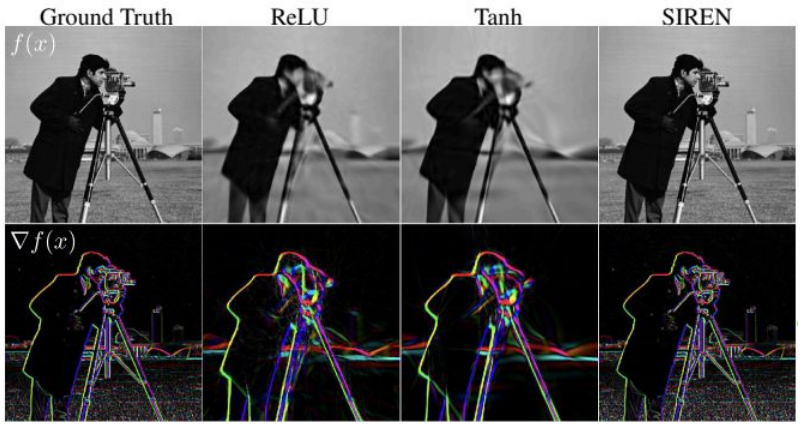
CoDiPack



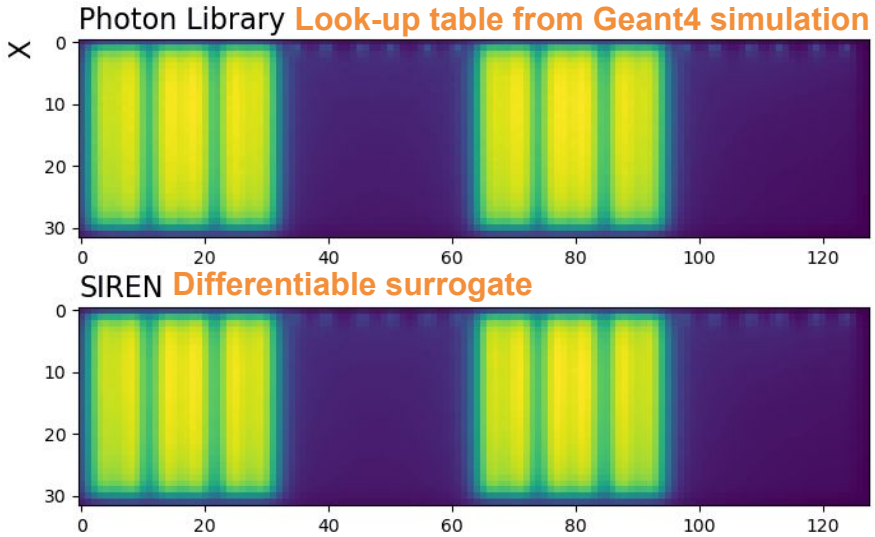


Differentiable photon transport

- **Differentiable neural surrogate** for inclusive modeling
- Focus on physics processes
- SIREN designed to learn accurate gradient field
- Conventional photon library: discrete, sampled, not scalable
- SIREN: O(100-1000) times fewer parameters
- Optimizable directly on data

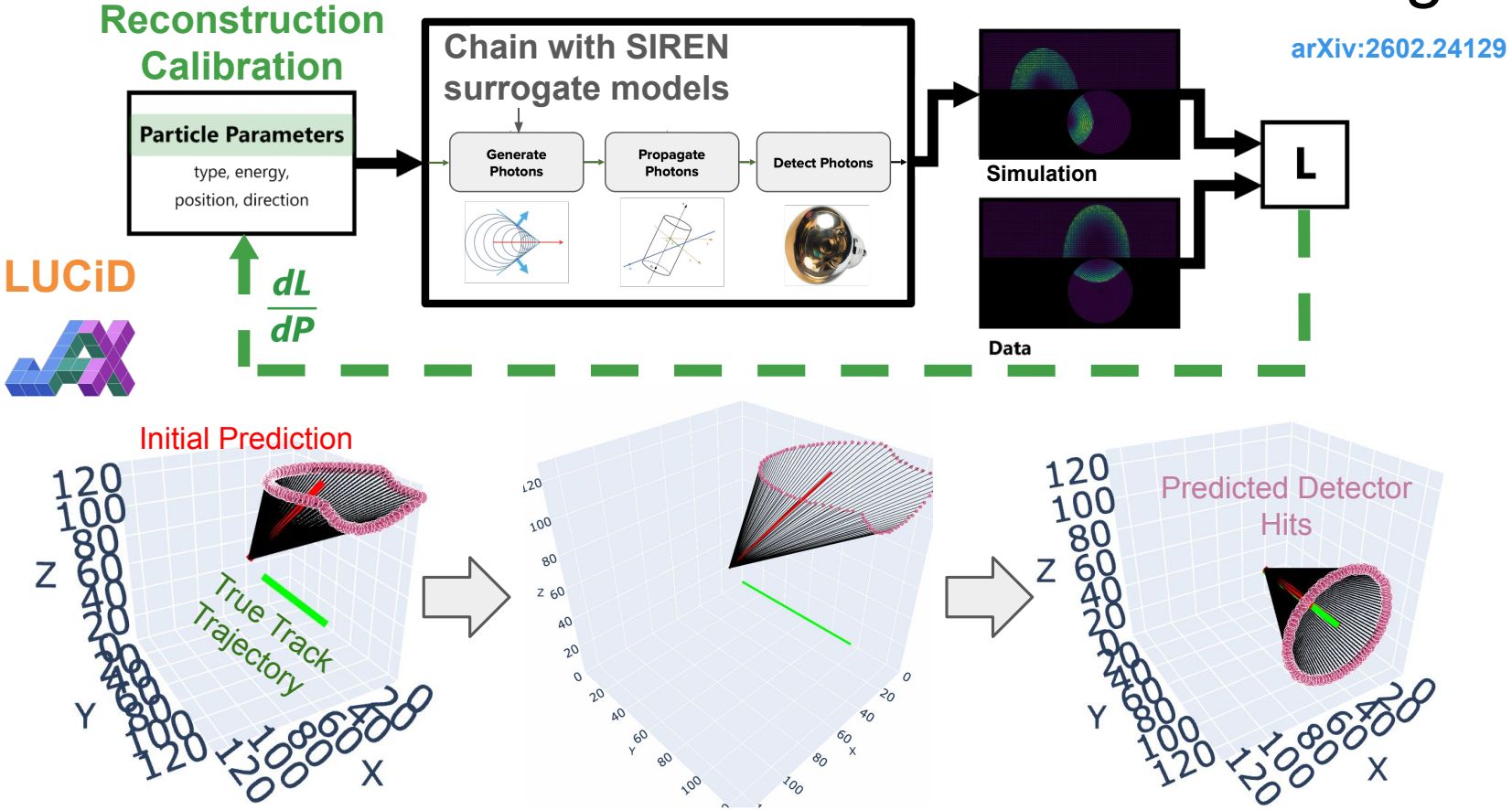


arXiv:2006.09661



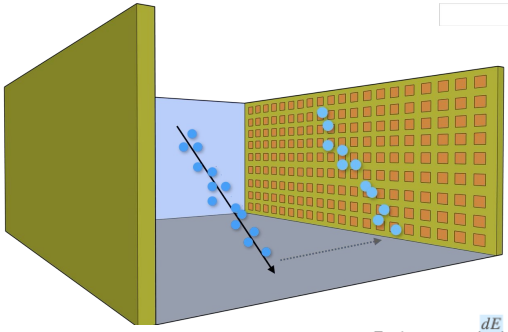
arXiv:2211.01505

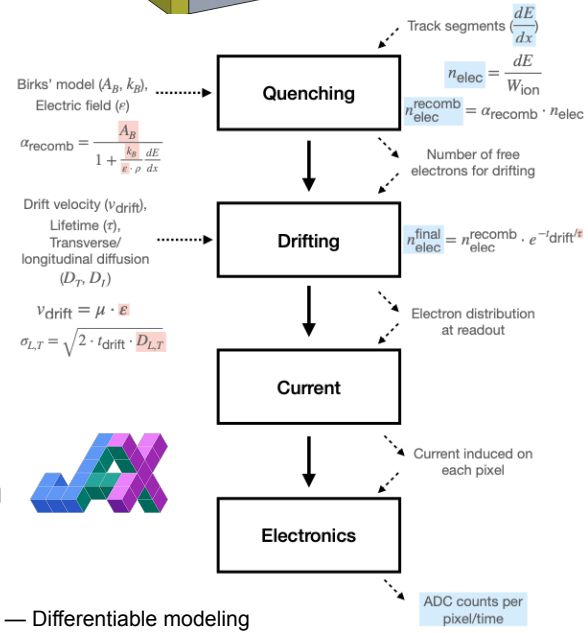
Differentiable water/ice Cherenkov modeling



Differentiable LArTPC modeling

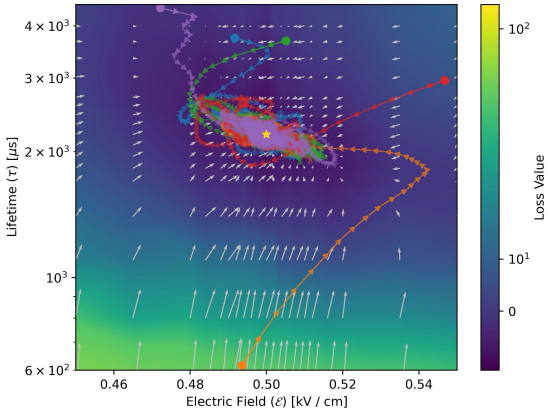
LADS

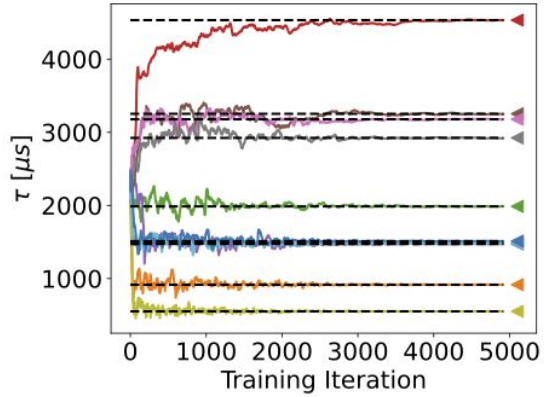




PyTorch

Mach. Learn.: Sci. Technol. 5 025012



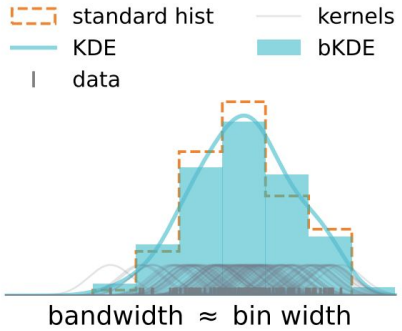


- Detector calibration
- Data application
- Developing fine tuning for particle reconstruction, and use gradients to train reconstruction to be robust against detector uncertainties
- Uncertainty propagation

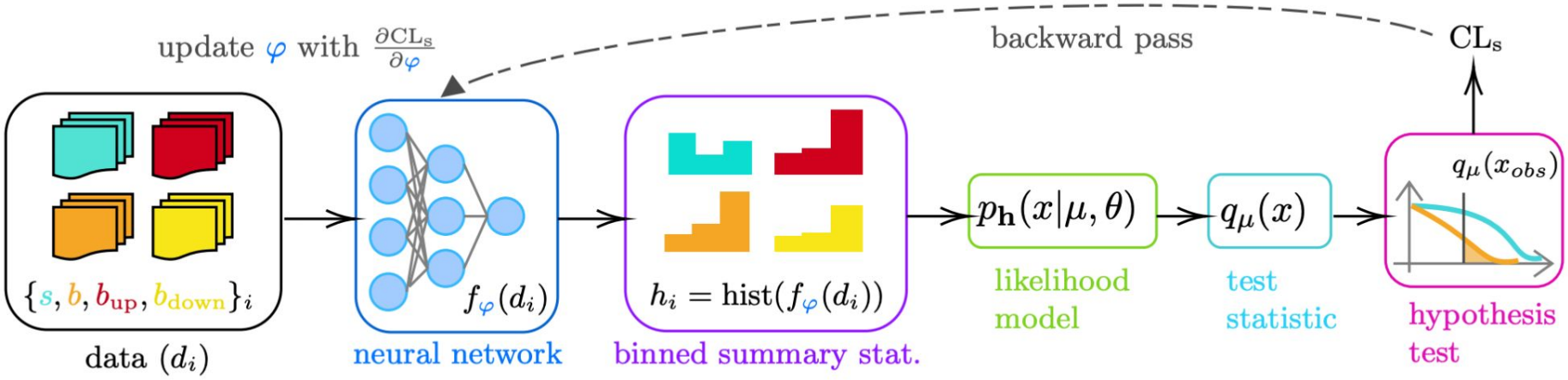
Differentiable analysis pipeline

neos

- Not differentiable physics modeling
- Optimize the analysis chain on the ultimate physics goal!
- Robust against uncertainties



J.Phys.Conf.Ser. 2438 (2023)



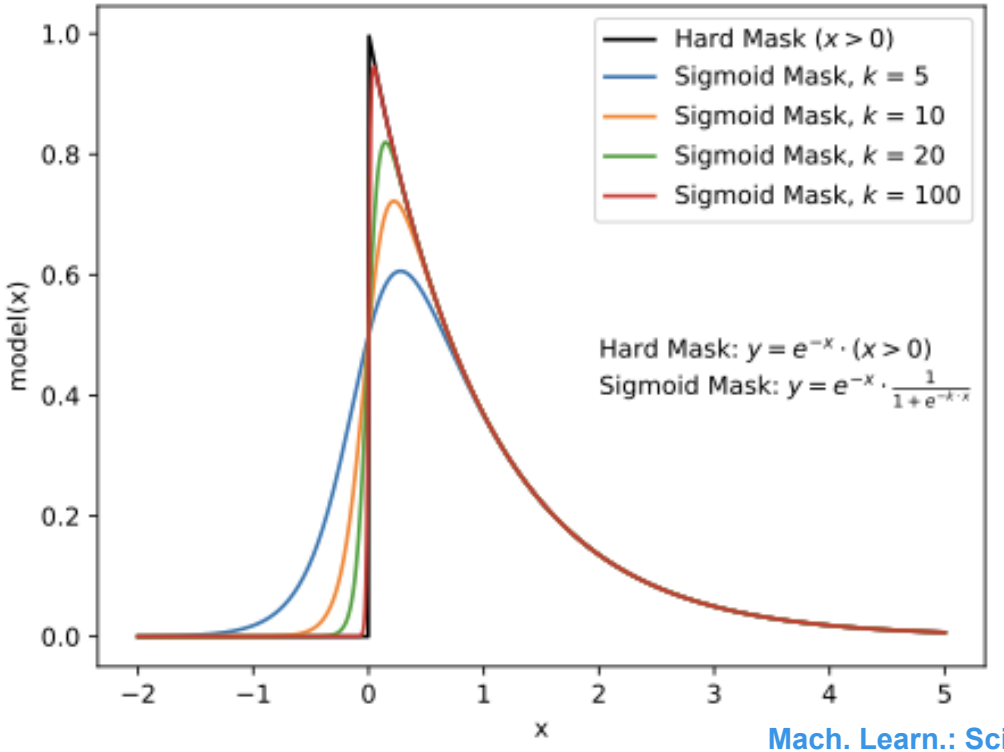
Tips for real differentiable modeling

Non-smoothness

E.g. Threshold, digitization, geometry boundary

<u>Example solution:</u> Replace with smooth functions

LADS



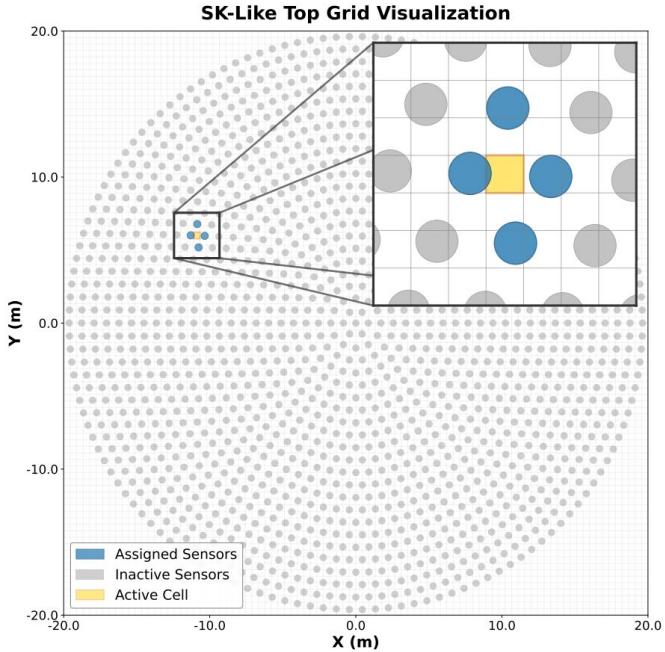
Mach. Learn.: Sci. Technol. 5 025012

Discrete categorical choices

E.g. Readout channel, interaction mode, detector material

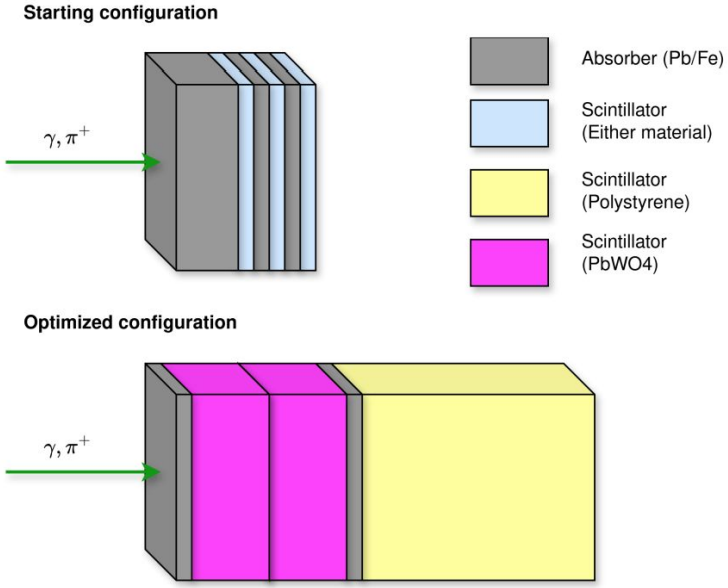
Example solution: Represent with probability, reparameterization trick

LUCiD



arXiv:2602.24129

End-to-end surrogate with diffusion models

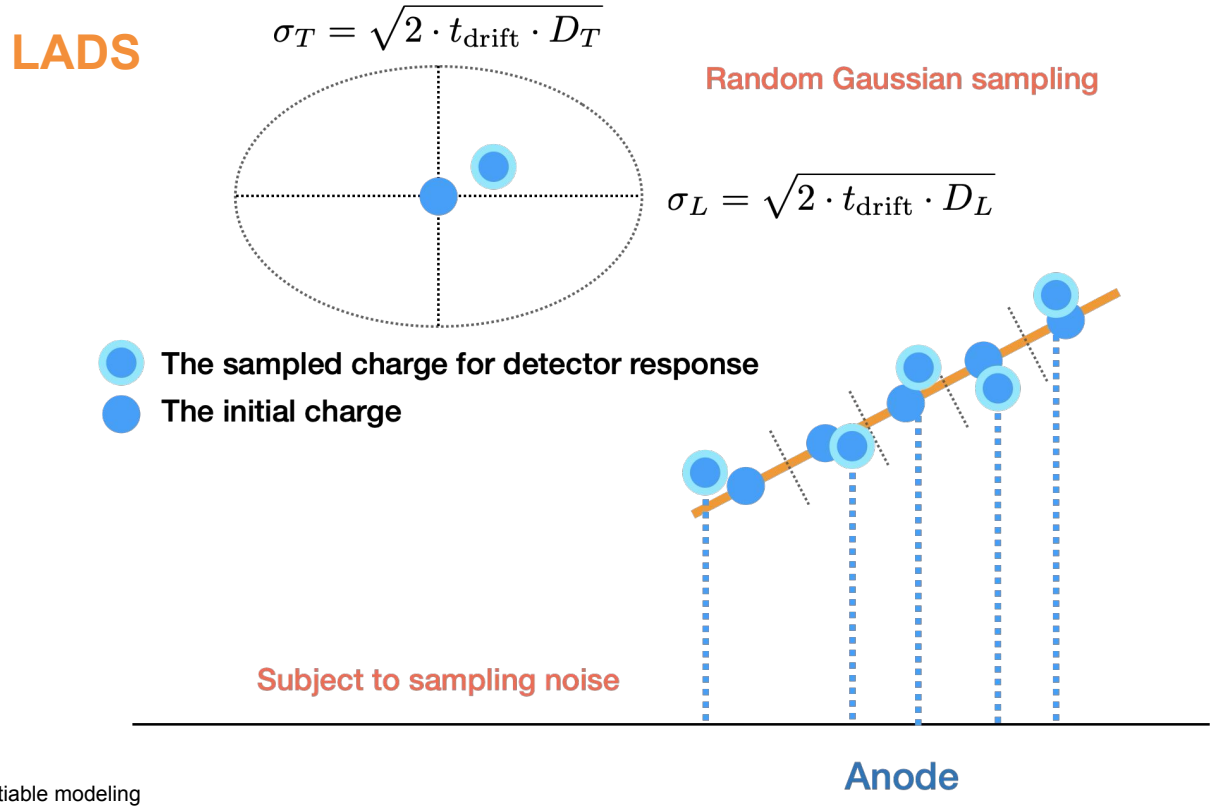


Particles 2025, *8*(2), 47

Sampling randomness

E.g. Particle energy deposition, charge diffusion

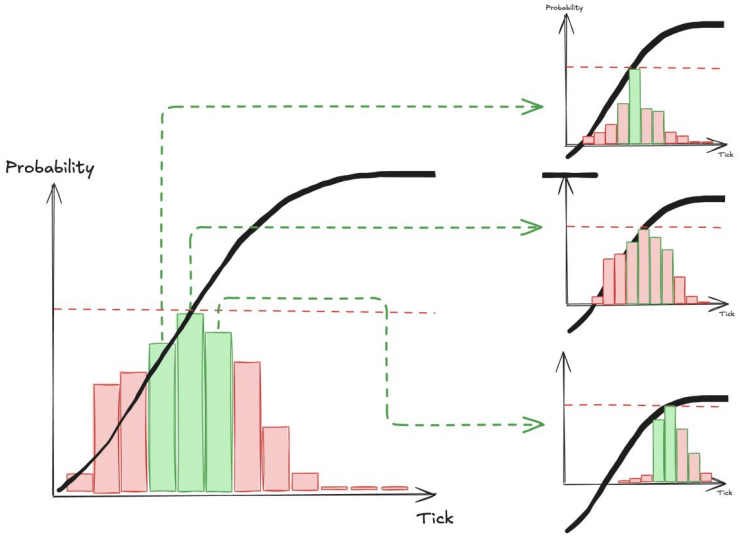
Example solution: Propagating distributions, reparameterization trick



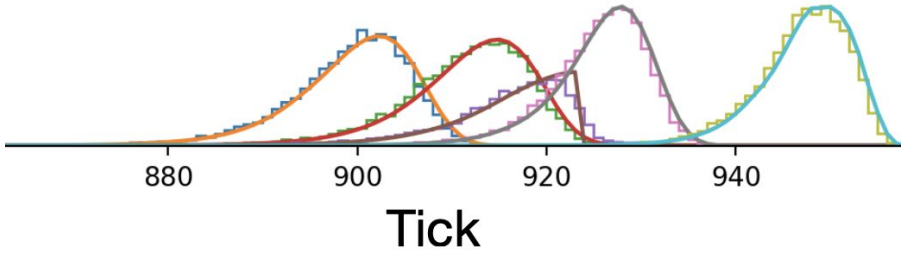
Sampling randomness

E.g. electronic noise

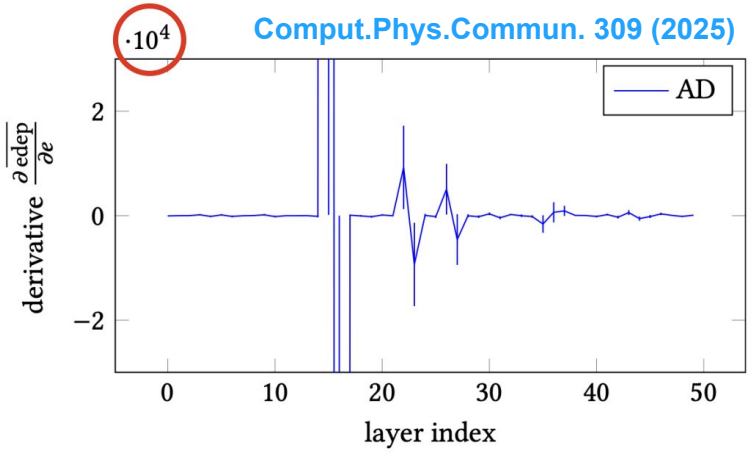
<u>Example solution:</u> Predicting probabilistic distributions in simulation

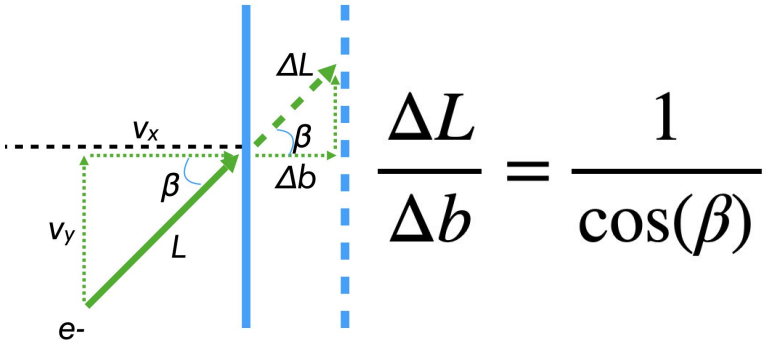


LADS



Exploding/vanishing gradients





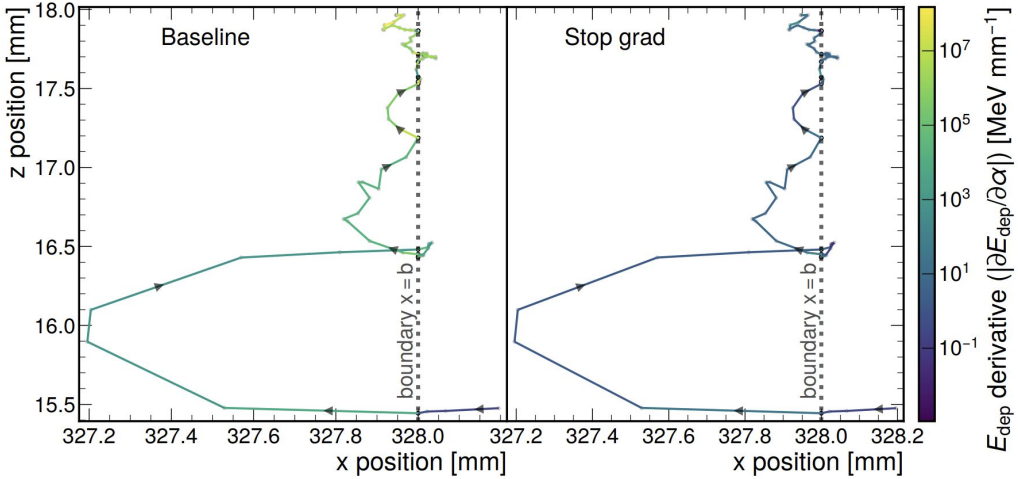
$$\frac{\Delta L}{\Delta b} = \frac{1}{\cos(\beta)}$$

E.g. EM shower crossing material boundaries

1. Discontinuities from boundaries
2. Stochasticity (e.g. multiple Coulomb scattering)
3. Accumulation over many steps

Example solution: Detach the problematic gradients

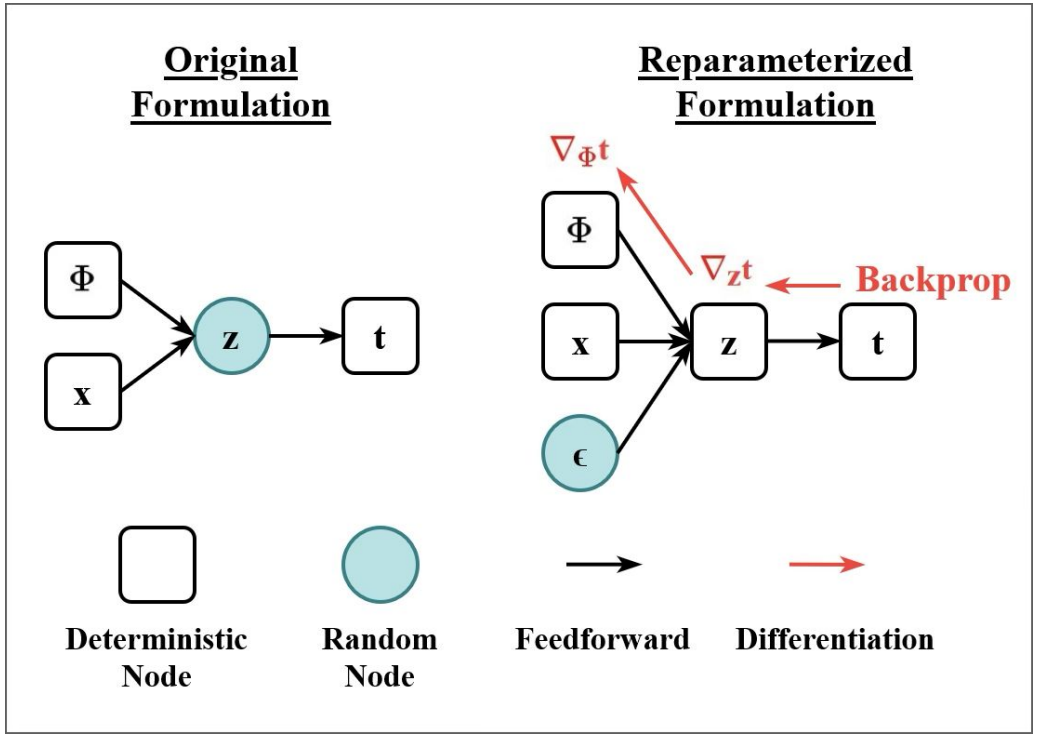
arXiv:2605.06779



Reparameterization trick

Remedy to many problems

Discreteness, stochasticity, exploding/vanishing gradients

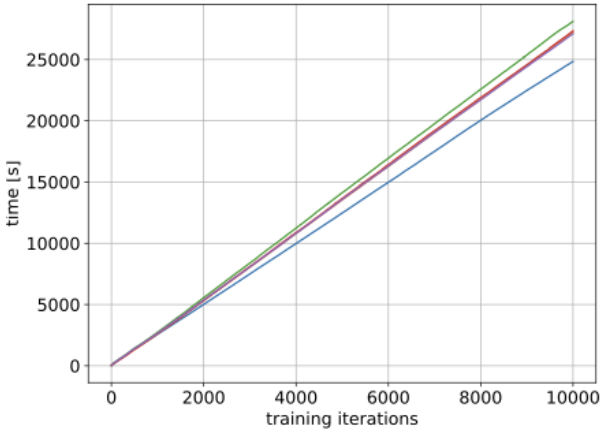


Simulation efficiency (Run speed & memory usage)

Simulation efficiency is crucial for real experimental applications!

- How many iterations you can fit?
- How much data you can include? (optimization loss landscape is approximation to the “true” based on the data we have)

~2.6s per 500 cm



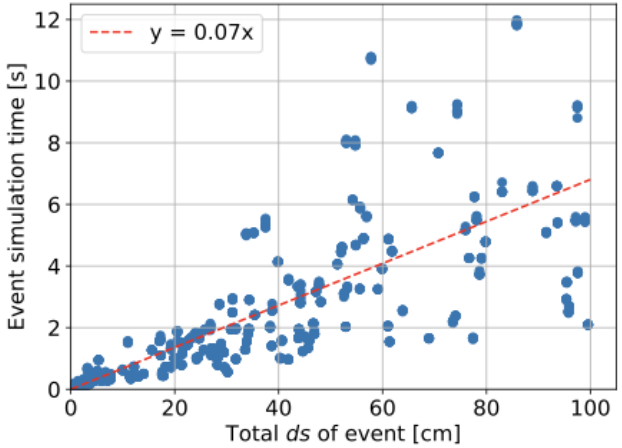
Non differentiable version

~5.6s per 500 cm

LADS

PyTorch

~35s per 500 cm



Mach. Learn.: Sci. Technol. 5 025012

Conclusions

- Differentiable modeling is a promising way to solve inverse problem
- Automatic differentiation is the backbone of differentiable modeling
- Many development of differentiable modeling in HEP
- Plan thoroughly for your differentiable modeling projects