LUMOS: Democratizing SciML Workflows with L0-Regularized Learning for Unified Feature and Parameter Adaptation

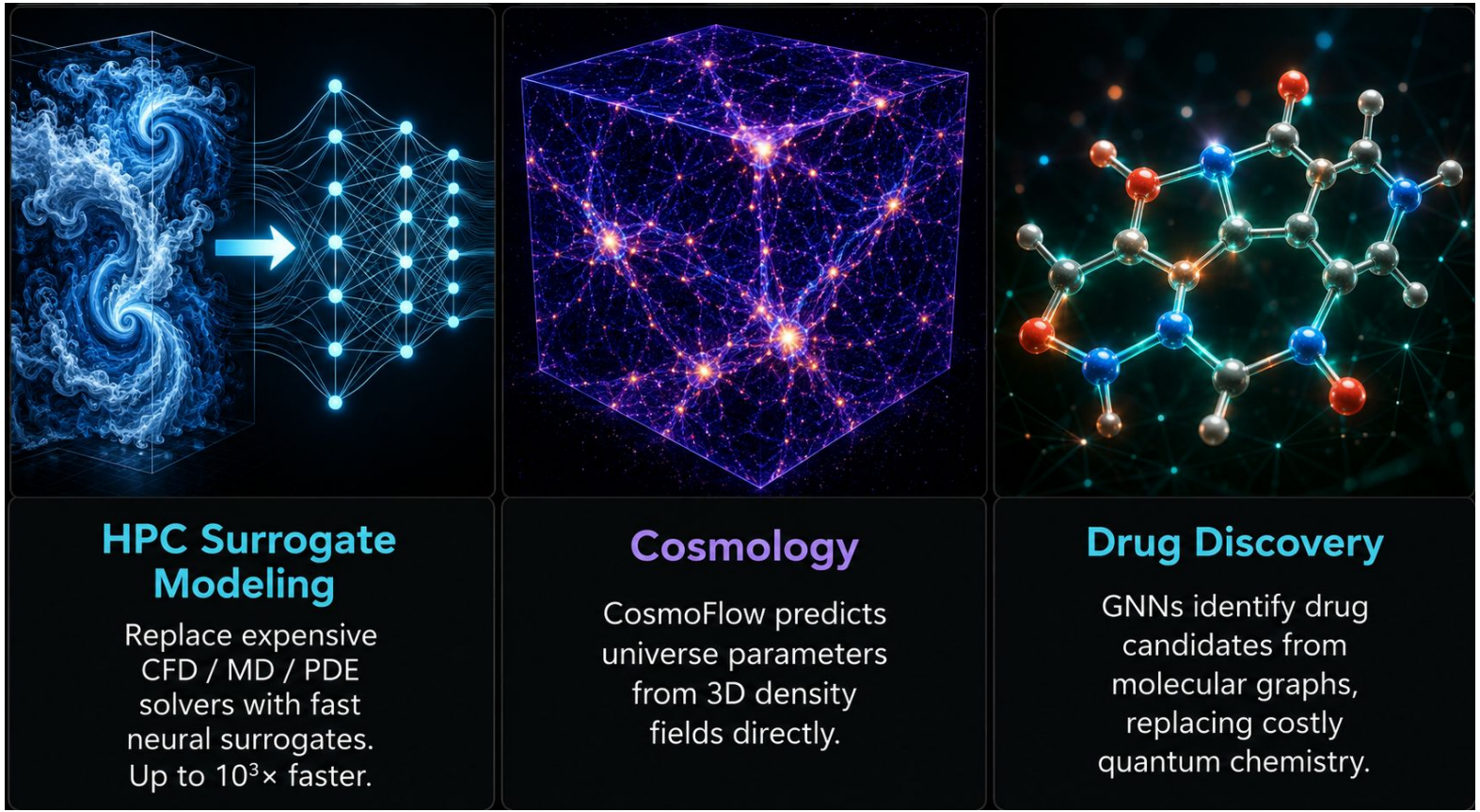
Shouwei Gao[1](Presenter), Xu Zheng[2], Dongsheng Luo[2], Sheng Di[3], Wenqian Dong[1]
[1]Oregon State University [2]Florida International University [3]Argonne National Laboratory

Picomp-Lab

ArXiv

GitHub Code

Scientific Machine Learning: Transforming How We Do Science



HPC Surrogate Modeling

Replace expensive CFD / MD / PDE solvers with fast neural surrogates. Up to $10^3\times$ faster.

Cosmology

CosmoFlow predicts universe parameters from 3D density fields directly.

Drug Discovery

GNNs identify drug candidates from molecular graphs, replacing costly quantum chemistry.

Why SciML Model Design is Hard

- SciML has enabled breakthroughs across many scientific domains

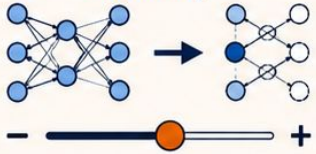
Climate science:
extreme weather prediction

Genomics:
gene expression analysis

Drug discovery:
candidate molecule screening

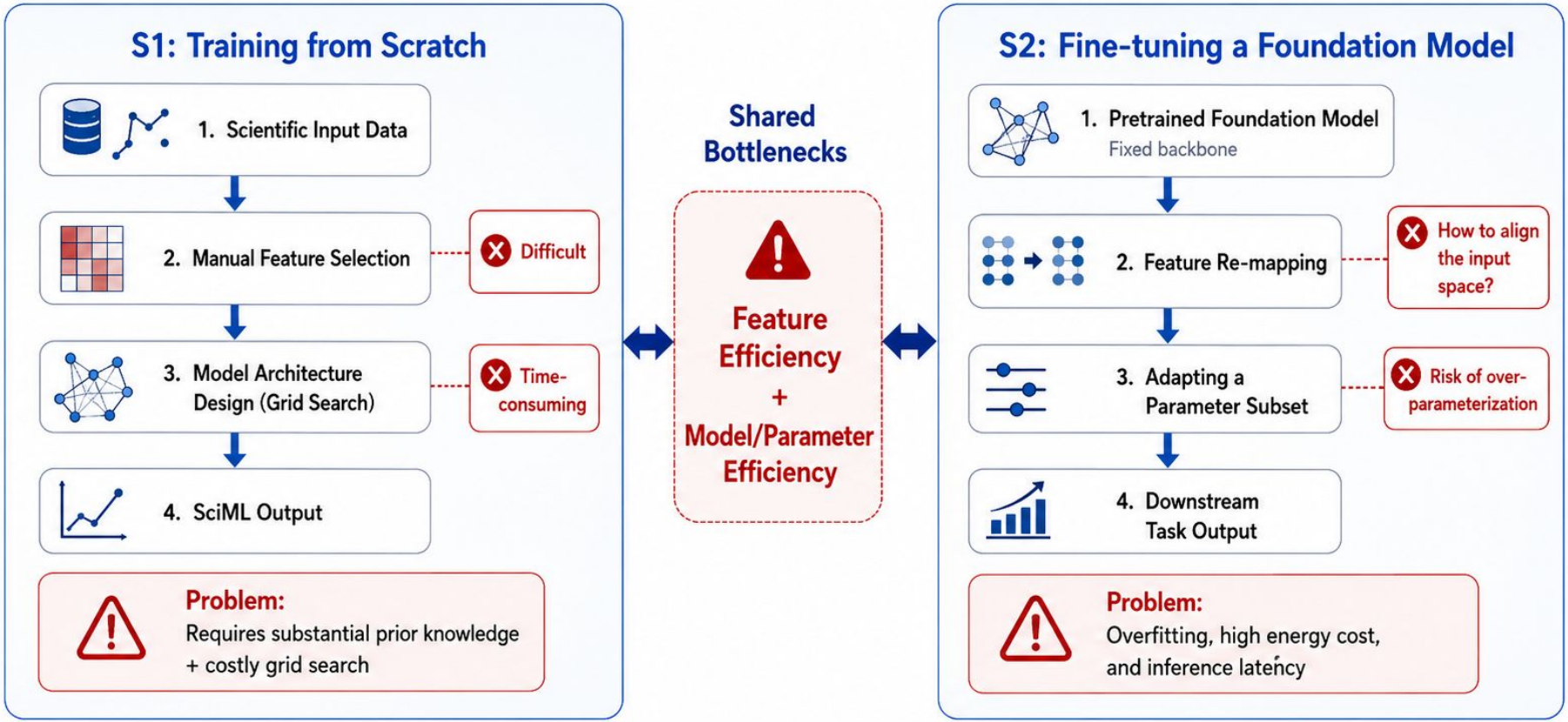
Combined with HPC, SciML can deliver $10^3\times$ acceleration

Two Underexplored Bottlenecks

Challenge	Description	Key Question
1 Feature Efficiency 	Selecting which input features carry the most information while removing irrelevant or redundant ones.	Which input features matter most?
2 Parameter Efficiency 	Choosing the appropriate model size/complexity to balance accuracy with computational cost and generalization.	How large should the model be?

> *Building effective SciML models requires substantial prior knowledge — determining input features and model size often relies on costly grid search.*

Bottlenecks in SciML Workflows

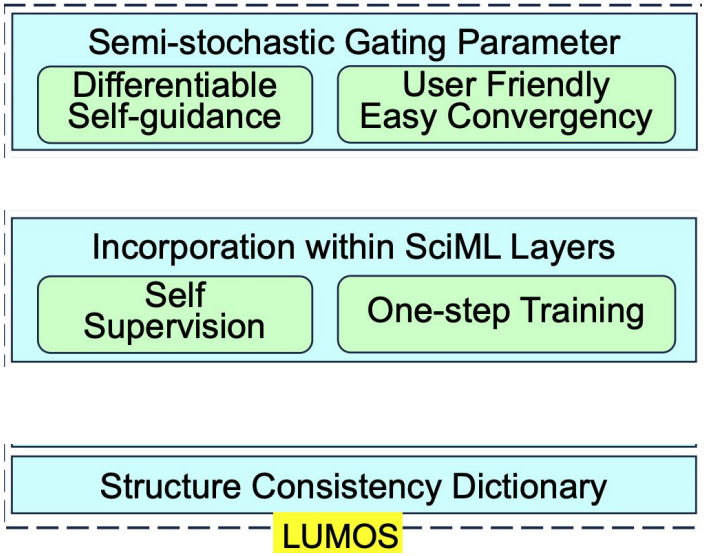


LUMOS: A Unified End-to-End Framework

Challenge 1: Efficient Pruning: Need for suitable, simple pruning algorithms to reduce model complexity without sacrificing performance.

Challenge 2: Universality: Solutions must be adaptable across a wide range of SciML layers (FC, Conv, GNN, Attention).

Challenge 3: Structural Integrity: Ensuring structural pruning leads to actual, physical model compression and hardware speedup.

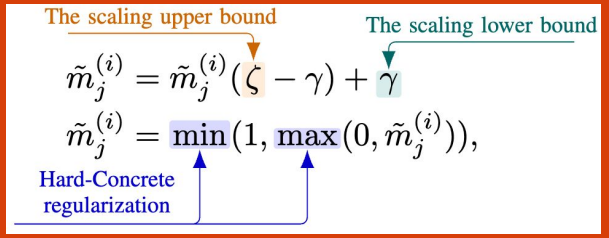


(I) Semi-Stochastic Gating & L0 Regularization

Semi-Stochastic Gating Parameters:

$$m_j^{(i)} = g(\pi_i^{(l)}) = \begin{cases} 1 & \pi_i^{(l)} > t_u, \\ \text{Bern}(\pi_i^{(l)}) & t_u \geq \pi_i^{(l)} > t_l, \\ 0 & t_l \geq \pi_i^{(l)}. \end{cases}$$

Hard-concrete relaxation:



Reparametrization for Differentiable Optimization:

$$\tilde{m}_j^{(i)} = \sigma\left(\frac{\log\frac{\epsilon}{1-\epsilon} + \alpha_j^{(i)}}{\tau}\right)$$

Accuracy loss + Complexity loss:

$$\mathcal{L}_T = \mathcal{L}_A + \lambda\mathcal{L}_C$$

(II) Gates Work Across All Major SciML Layers

(1) FC Layer:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{X}^{(l)} \cdot \mathbf{m}^{(l)} \odot \mathbf{W}^{(l)} + \mathbf{b}^{(l)}\right).$$

Gating parameters ($\mathbb{R}^{1\times n}$); Input feature $\mathbb{R}^{m\times n}$; Layer weights $\mathbb{R}^{n\times k}$

(2) Convolutional layer:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathrm{conv}(\mathbf{X}^{(l)}, \mathbf{m}^{(l)} \odot \mathbf{W}^{(l)}) + \mathbf{b}^{(l)}\right)$$

Convolution operation

(3) GNN Layer:

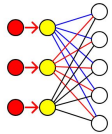
$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{W}^{(l)}((1+\varepsilon)\cdot \mathbf{m}^{(l)} \odot \mathbf{X}^{(l)} + \mathbf{m}^{(l)} \odot prop(emb(\mathbf{E}), \mathbf{X}^{(l)})))$$

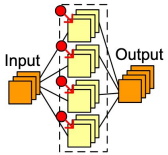
Learnable parameter; Aggregates neighbor information; Embeds edge attributes

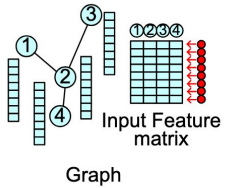
(4) Attention Layer

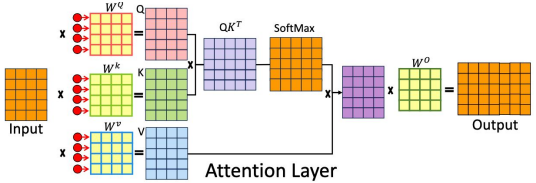
$$\mathbf{Q}^{(l)} = \mathbf{X}^{(l)} \cdot \mathbf{m}_Q^{(l)} \odot \mathbf{W}_Q^{(l)}$$
$$\mathbf{K}^{(l)} = \mathbf{X}^{(l)} \cdot \mathbf{m}_K^{(l)} \odot \mathbf{W}_K^{(l)}$$
$$\mathbf{V}^{(l)} = \mathbf{X}^{(l)} \cdot \mathbf{m}_V^{(l)} \odot \mathbf{W}_V^{(l)}$$
$$\mathbf{H}^{(l+1)} = \mathrm{Attention}(\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)})\mathbf{W}_O^{(l)}$$

softmax $\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}$







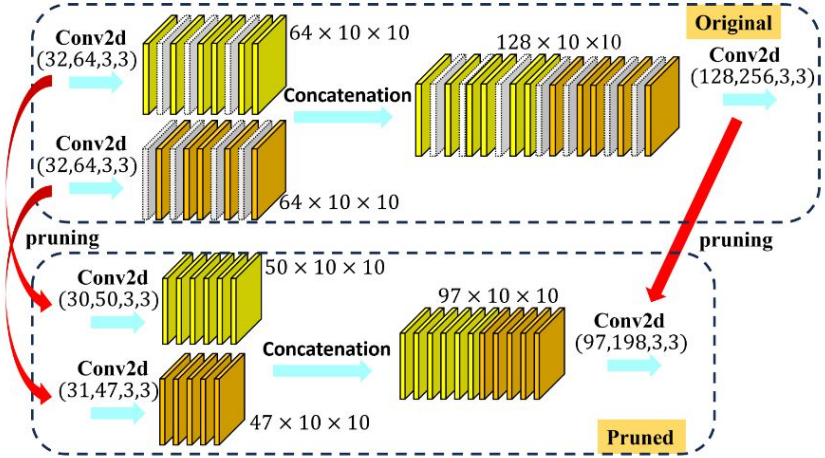


Gate; FC layer's input features; Kernel; Graph node's feature

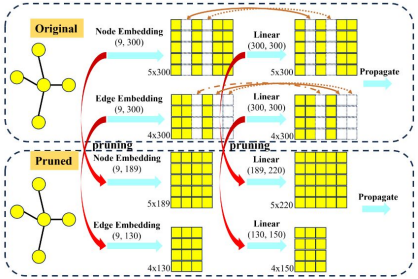
(III) Structural Consistency: Keeping the Pruned Model Executable

After pruning, LUMOS must ensure:

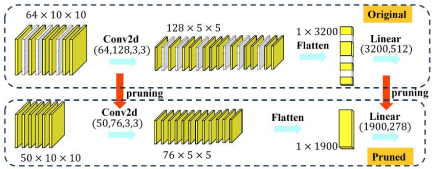
- Tensor dimensions still match;
- Layer dependencies remain valid;
- Outputs are still executable;
- Residual and concatenation structures are preserved.



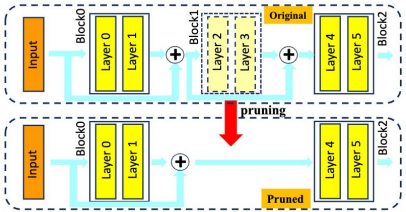
Concatenation



Graph layers



Flatten



Layer removal

Experimental Setup

Test platform: Cluster with 8 A100 GPUs

TABLE II: Summary of evaluated scientific workloads

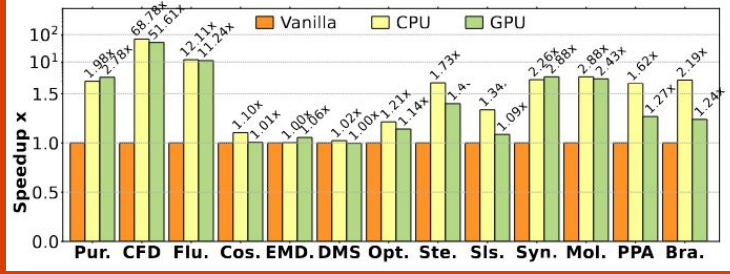
Workloads	Domain	Network	Task
CFD [59]	Fluid Simulation	FCs	Surrogate:`Compute_Flux()`
Fluid [59]	Fluid Simulation	FCs	Surrogate:`Compute_Force()`
Puremd [59]	Molecular	FCs	Surrogate:`Add_dBond_to_Forces()`
CosmFlow [60]	Cosmology	CONVs+FCs	Regression: The universe parameters
EM-DN [61]	Material Sciences	Unet	Regression: Denoising
Synthetic [61]	Math formulation	FCs	Regression: Math formulation
Optical [61]	Instrumentation	Enc.&Dec.	Detection: Optical equipment damage
DMS [61]	Material Sciences	CONVs+FCs	Classification: Image
STEM-DL [61]	Material Sciences	VGG	Classification: Diffraction patterns
SLSTR [61]	Atmospheric	Unet	Classification: Image(at pixel level)
PPA [62]	Biological Science	GCN	Classification: Taxonomic group
Molhiv [63]	Chemical Science	GIN	Classification: Molecular properties
BrainTumor [64]	Medical	ViT	Classification: Tumor detection

Metrics include:

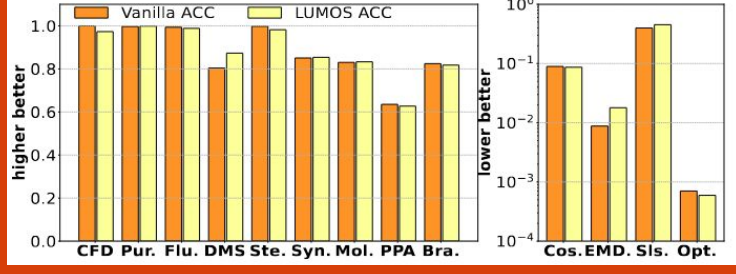
- Inference speedup;
- Accuracy;
- Parameter reduction;
- FLOPs reduction;
- Memory usage;
- Energy consumption;
- Scalability.

[1] Thiyagalingam, Jeyan, et al. "Scientific machine learning benchmarks." Nature Reviews Physics 4.6 (2022): 413-420.

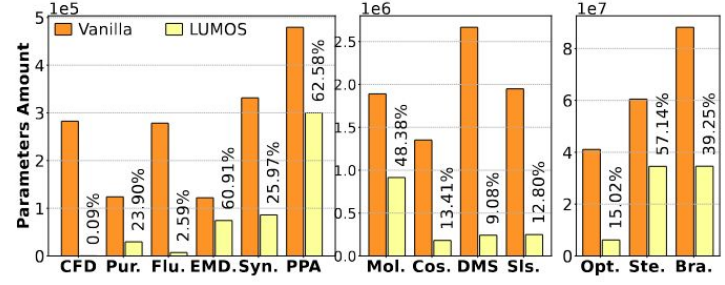
Main Results: Speedup, Accuracy & Compression



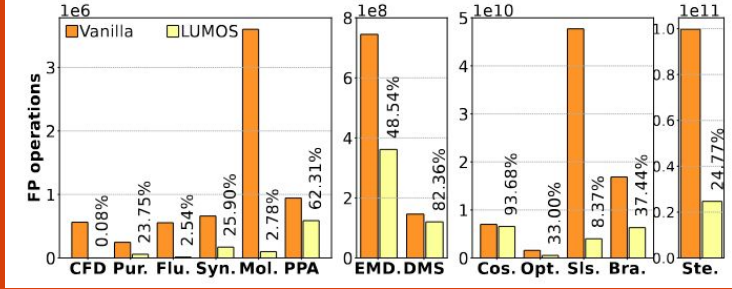
Average 6.4x GPU Speedup



Accuracy Matches to Vanilla

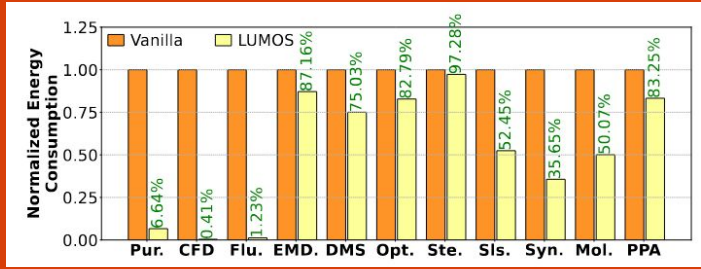


Average 71% Parameter Reduction

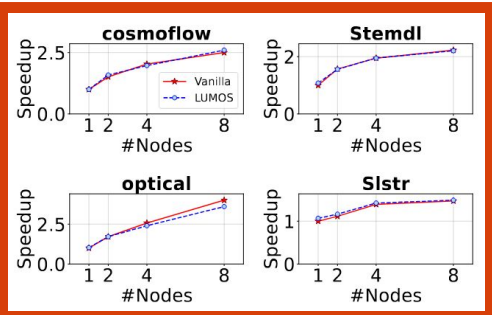


Average 69% Floating Point Reduction

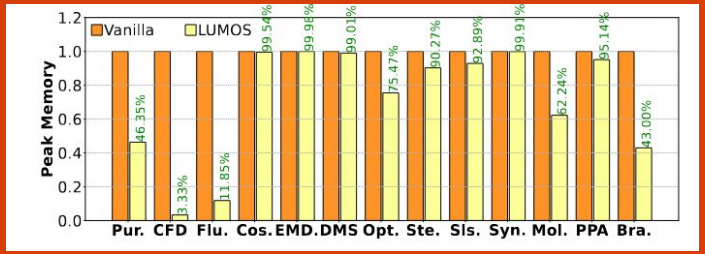
Energy, Memory & Scalability



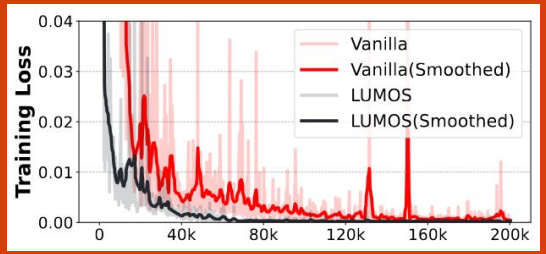
Average 50.7% Power Reduction



Scale up to 8 V100 GPUs



Average 23% Peak Memory Reduction



Training process

Conclusion

LUMOS democratizes SciML model design by reducing manual feature and architecture tuning. It helps scientists build models that are:

Smaller | **Faster** | **Energy-efficient** | **Easier to deploy** | **Still accurate**

Take away:

LUMOS makes SciML workflows more automatic, efficient, and practical for real-world scientific applications.

Thanks
Any questions?

Let's discuss any doubts or concerns.

If something comes up later, contact **gaosho@oregonstate.edu**

5. LUMOS vs. SOTA Pruning Methods & Feature Interpretability

TABLE III: Comparison with SOTA pruning methods

Metrics	OTO	L1	NEURAL	DepGraph	LUMOS
One-Shot	Yes	No	No	Yes	Yes
Compression	Yes	Yes	No	Yes	Yes
Adaptiveness	No	No	No	No	Yes
Puremd(%↑)	72.73%	76.3%	80.10%	76.35%	76.10%
Puremd(R^2↑)	0.9430	0.9969	0.9318	0.9937	0.9988
CFD(%↑)	99.10%	99.12%	99.10%	99.12%	99.10%
CFD(R^2↑)	0.5933	0.9136	0.9785	0.9747	0.9732
EMD.(%↑)	30.10%	39.30%	39.50%	39.33%	39.09%
EMD.(MSE↓)	9.13e-3	0.0401	0.021	0.041	9.51e-3
DMS.(%↑)	91.00%	91.02%	91.09%	91.22%	90.92%
DMS.(ACC↑)	67.35	75.94	72.06	71.11%	87.33
Bra.(%↑)	60.34%	59.98%	61.09%	-	60.75%
Bra.(ACC↑)	78.41	80.94	79.06	-	81.77

Note: '↑' indicates that higher is better; '↓' indicates that lower is better; '-' indicates that the latest version can't support this application.