1D Transport Using Neural Nets, SN, and MC

M&C 2019
Portland, Oregon
August 25-29, 2019

August 27, 2019

Mike Pozulp

THE UNIVERSITY OF CALIFORNIA DAVIS

LLNL-PRES-787894
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

This talk has three parts

1) Motivation

2) Methods

3) Results

   - The transport equation can be solved using a neural network (NN). The runtime is competitive with SN and MC for a simple test problem.

Porting code is hard when everything changes

Algorithm			
Language			
Library			
Compiler			
Runtime			
ISA			
Hardware			

Porting code is hard when everything changes

	Trinity Phase 1		
Algorithm	MC		
Language	C++		
Library	MPI		
Compiler	Intel		
Runtime	OpenMP		
ISA	X86/AVX2		
Hardware	Haswell		

Porting code is hard when everything changes

	Trinity Phase 1	Trinity Phase 2	
Algorithm	MC	MC	
Language	C++	C++	
Library	MPI	MPI	
Compiler	Intel	Intel	
Runtime	OpenMP	OpenMP	
ISA	X86/AVX2	X86/AVX512	
Hardware	Haswell	KNL	

Porting code is hard when everything changes

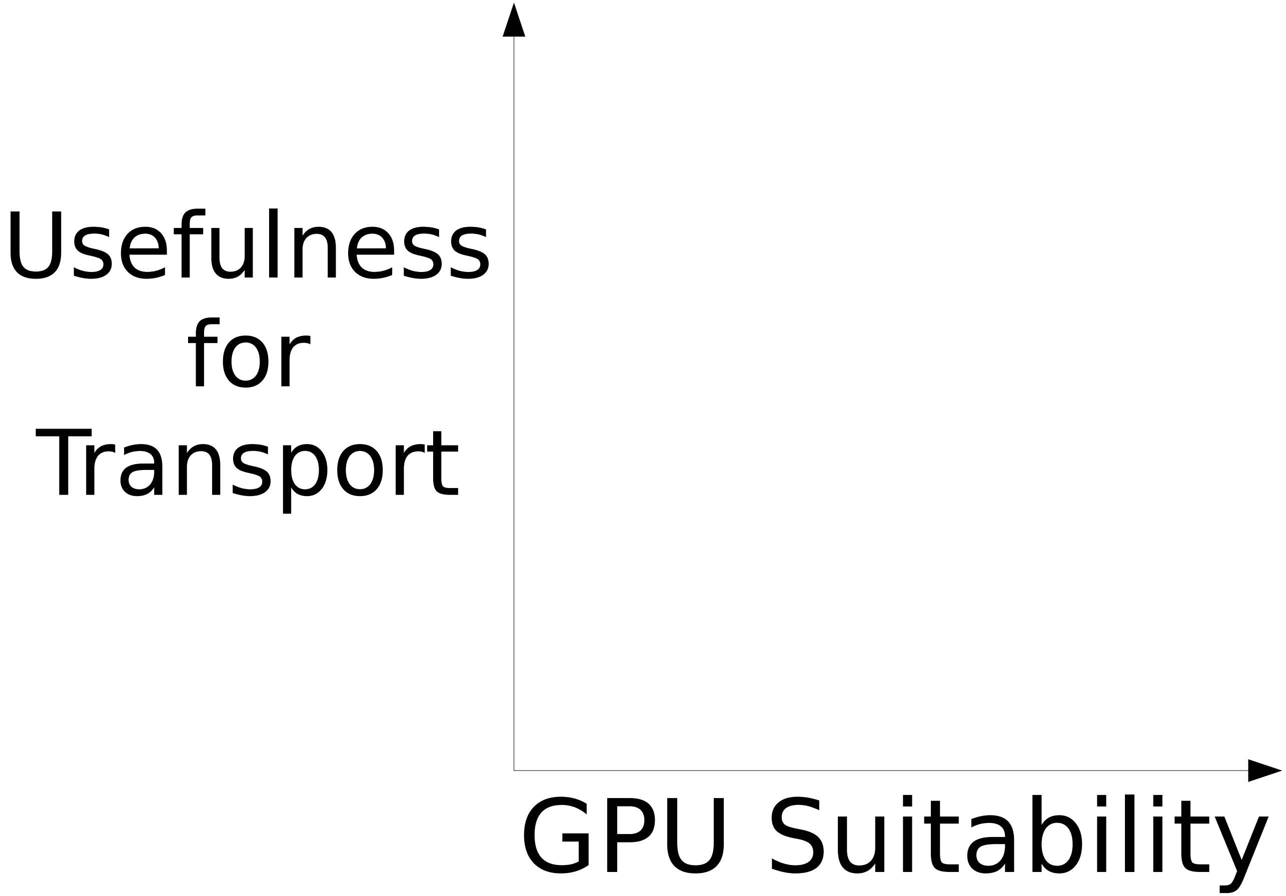
	Trinity Phase 1	Trinity Phase 2	Sierra
Algorithm	MC	MC	MC
Language	C++	C++	CUDA
Library	MPI	MPI	Torch
Compiler	Intel	Intel	NVCC
Runtime	OpenMP	OpenMP	CUDA-RT
ISA	X86/AVX2	X86/AVX512	PTX
Hardware	Haswell	KNL	Volta

Porting code is hard when everything changes

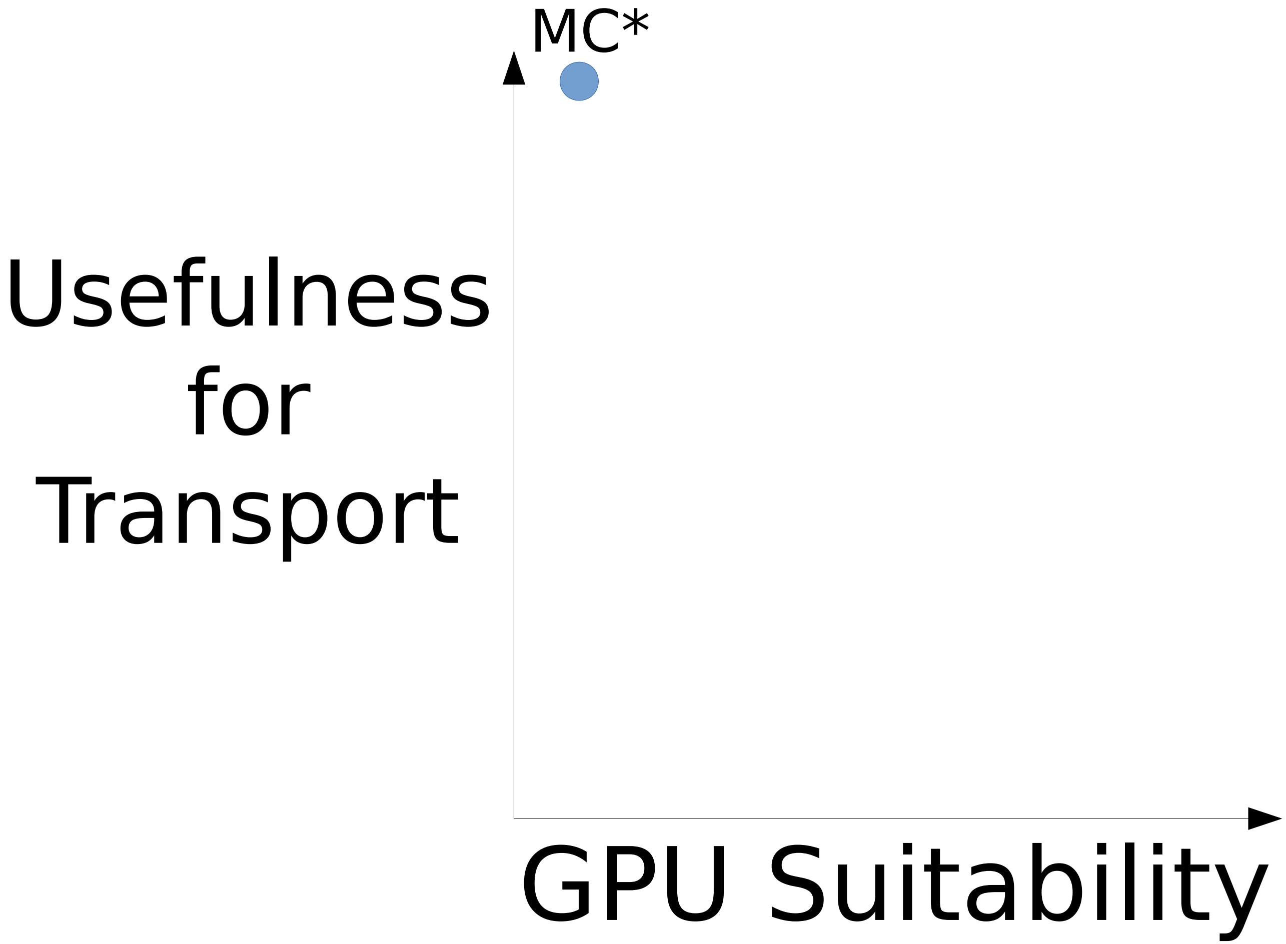
	Trinity Phase 1	Trinity Phase 2	Sierra
Algorithm	MC	MC	~~MC~~ NN
Language	C++	C++	CUDA
Library	MPI	MPI	Torch
Compiler	Intel	Intel	NVCC
Runtime	OpenMP	OpenMP	CUDA-RT
ISA	X86/AVX2	X86/AVX512	PTX
Hardware	Haswell	KNL	Volta

We can port from both sides

We can port from both sides

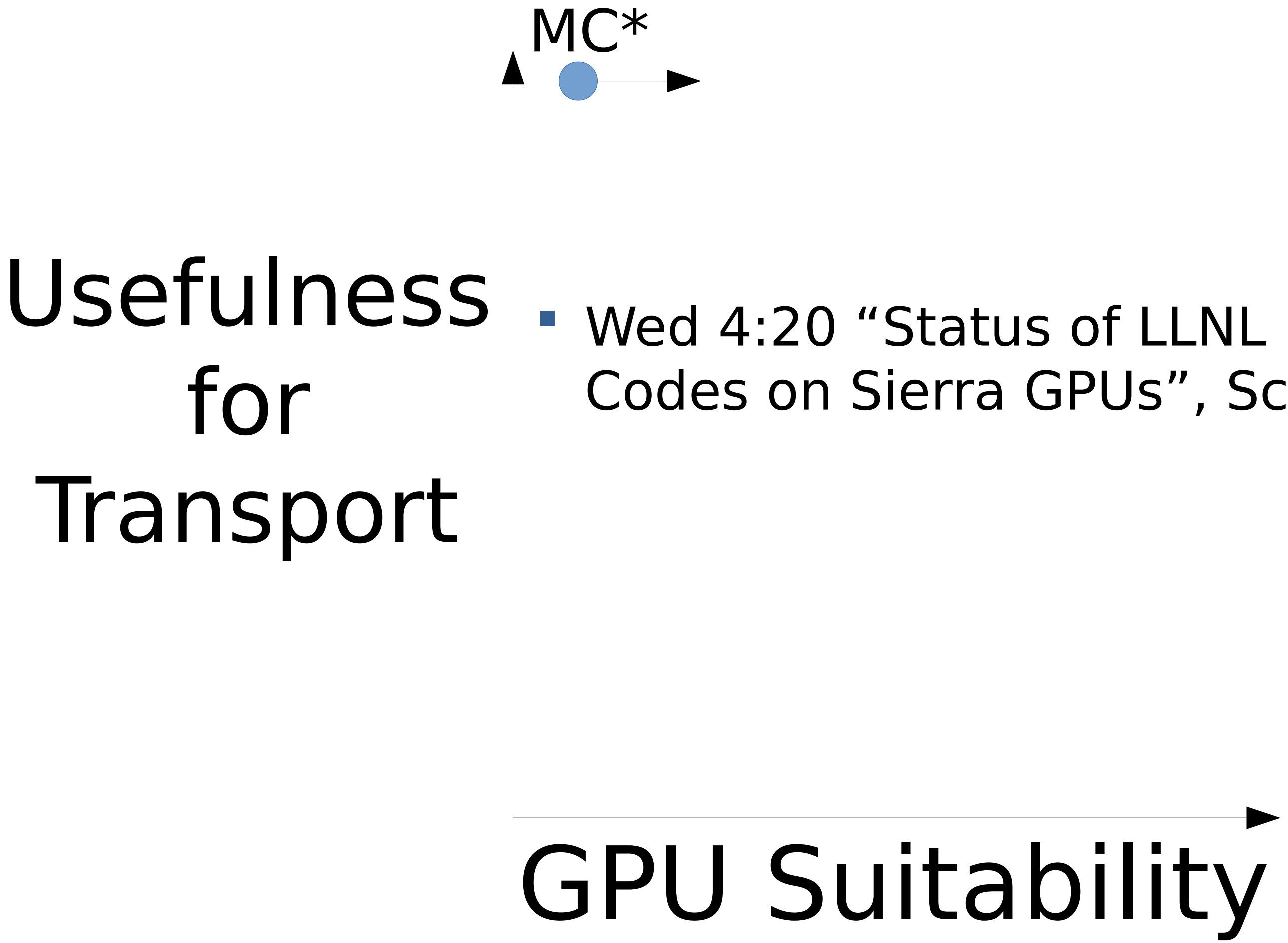


We can port from both sides



**History-based* Monte Carlo

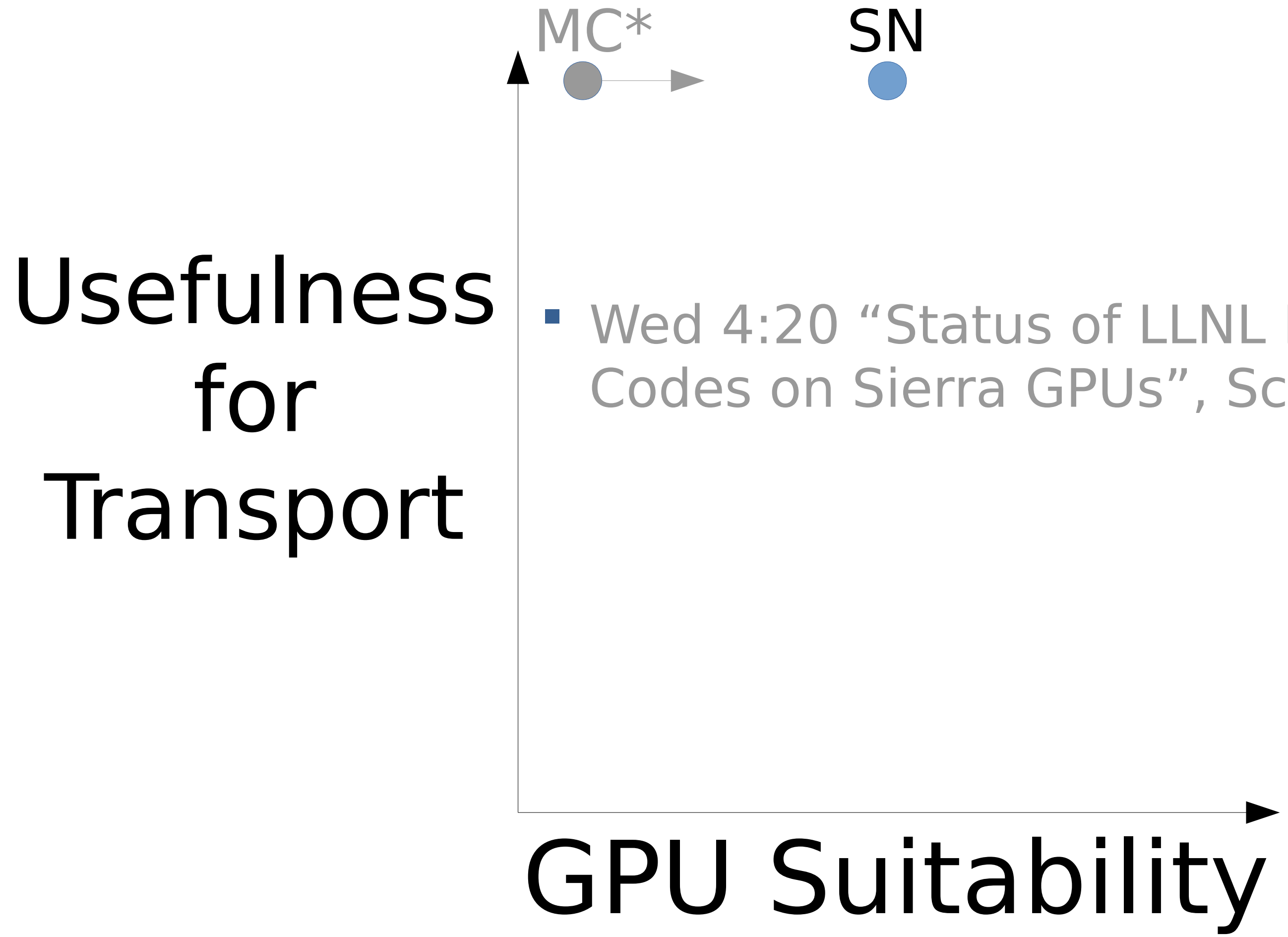
We can port from both sides



- Wed 4:20 “Status of LLNL Monte Carlo Transport Codes on Sierra GPUs”, Scott McKinley et al

***History-based* Monte Carlo

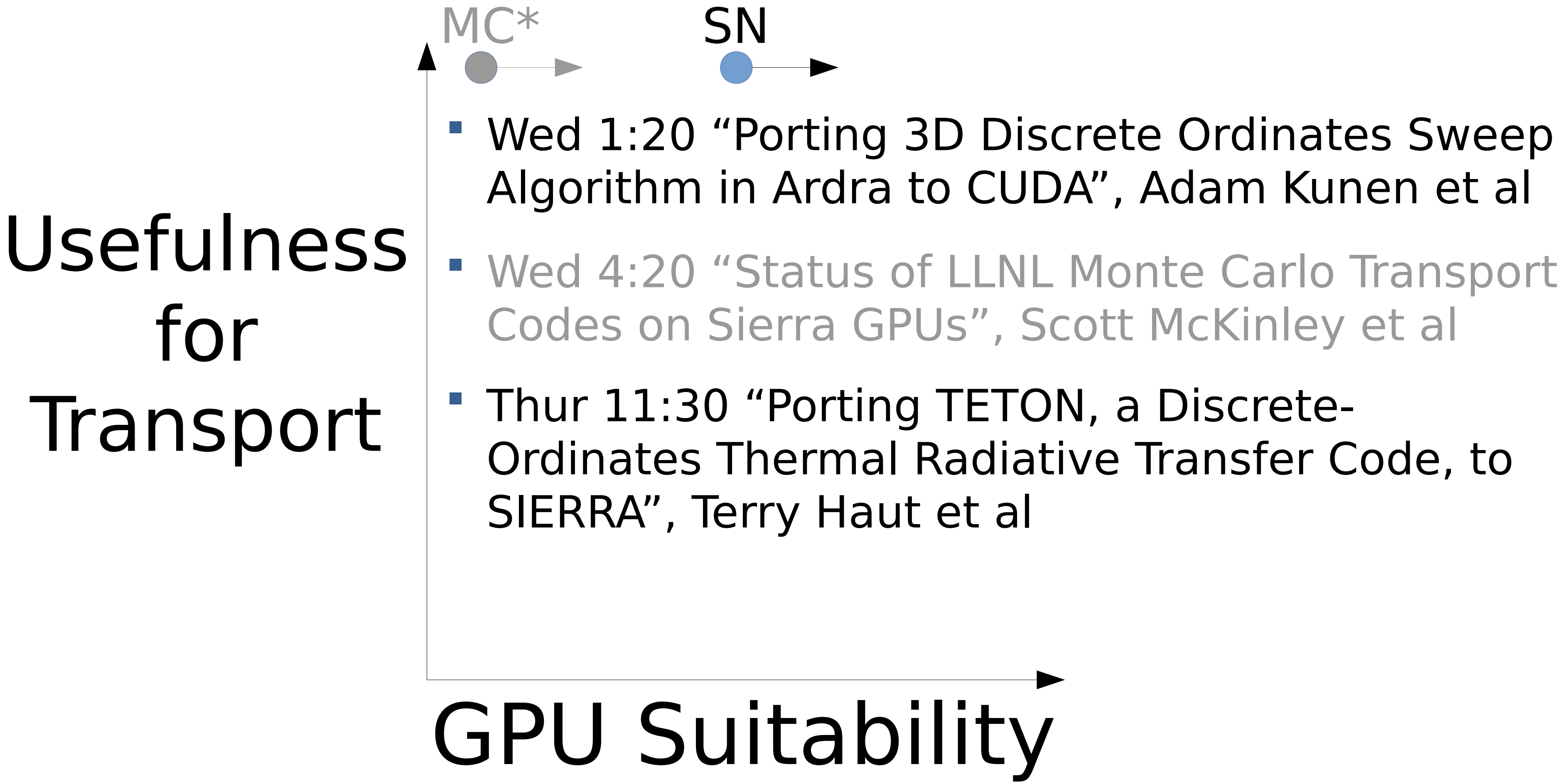
We can port from both sides



- Wed 4:20 “Status of LLNL Monte Carlo Transport Codes on Sierra GPUs”, Scott McKinley et al

**History-based* Monte Carlo

We can port from both sides



MC*

SN

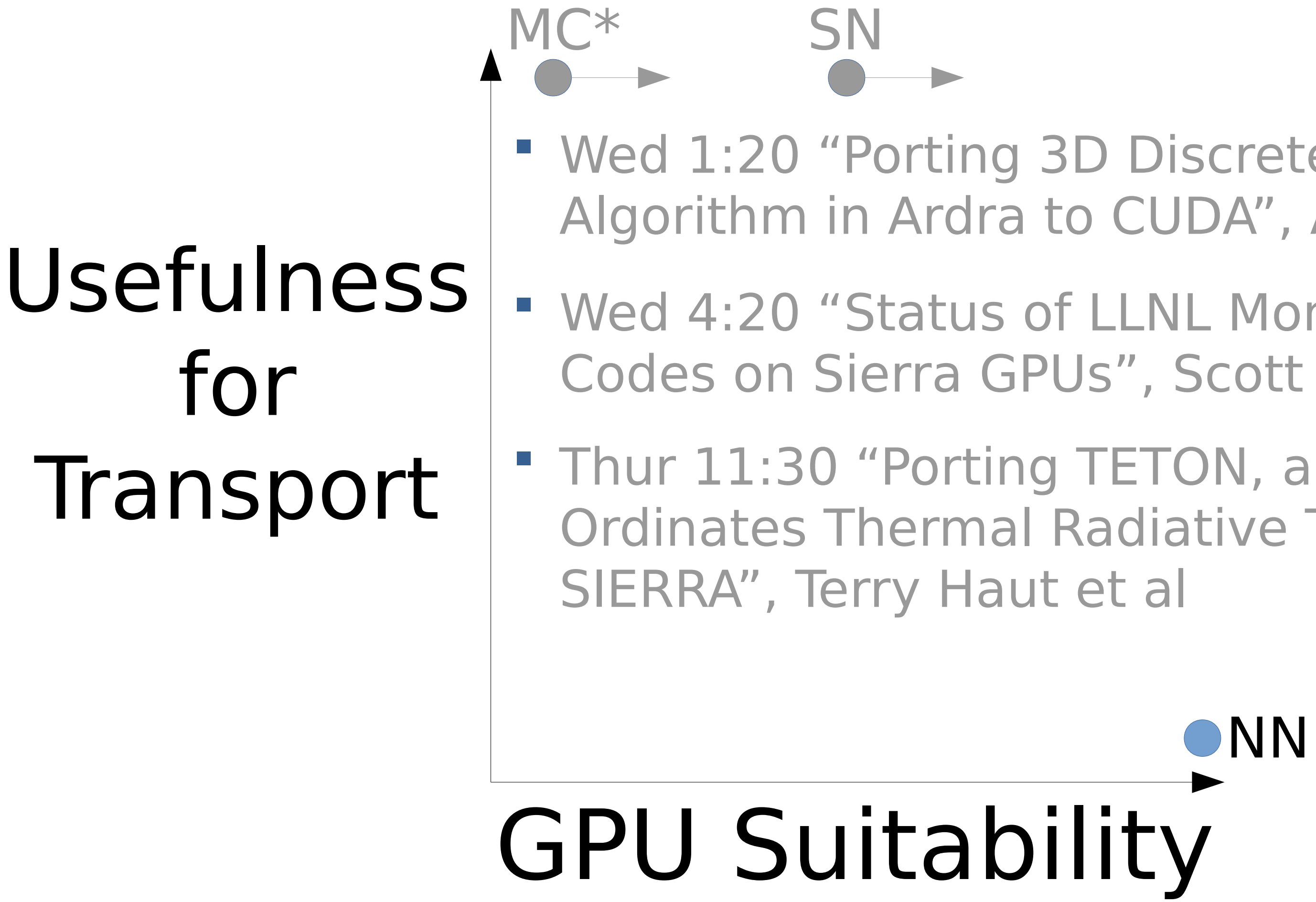
Usefulness for Transport

- Wed 1:20 “Porting 3D Discrete Ordinates Sweep Algorithm in Ardra to CUDA”, Adam Kunen et al
- Wed 4:20 “Status of LLNL Monte Carlo Transport Codes on Sierra GPUs”, Scott McKinley et al
- Thur 11:30 “Porting TETON, a Discrete-Ordinates Thermal Radiative Transfer Code, to SIERRA”, Terry Haut et al

GPU Suitability

*_History-based_ Monte Carlo

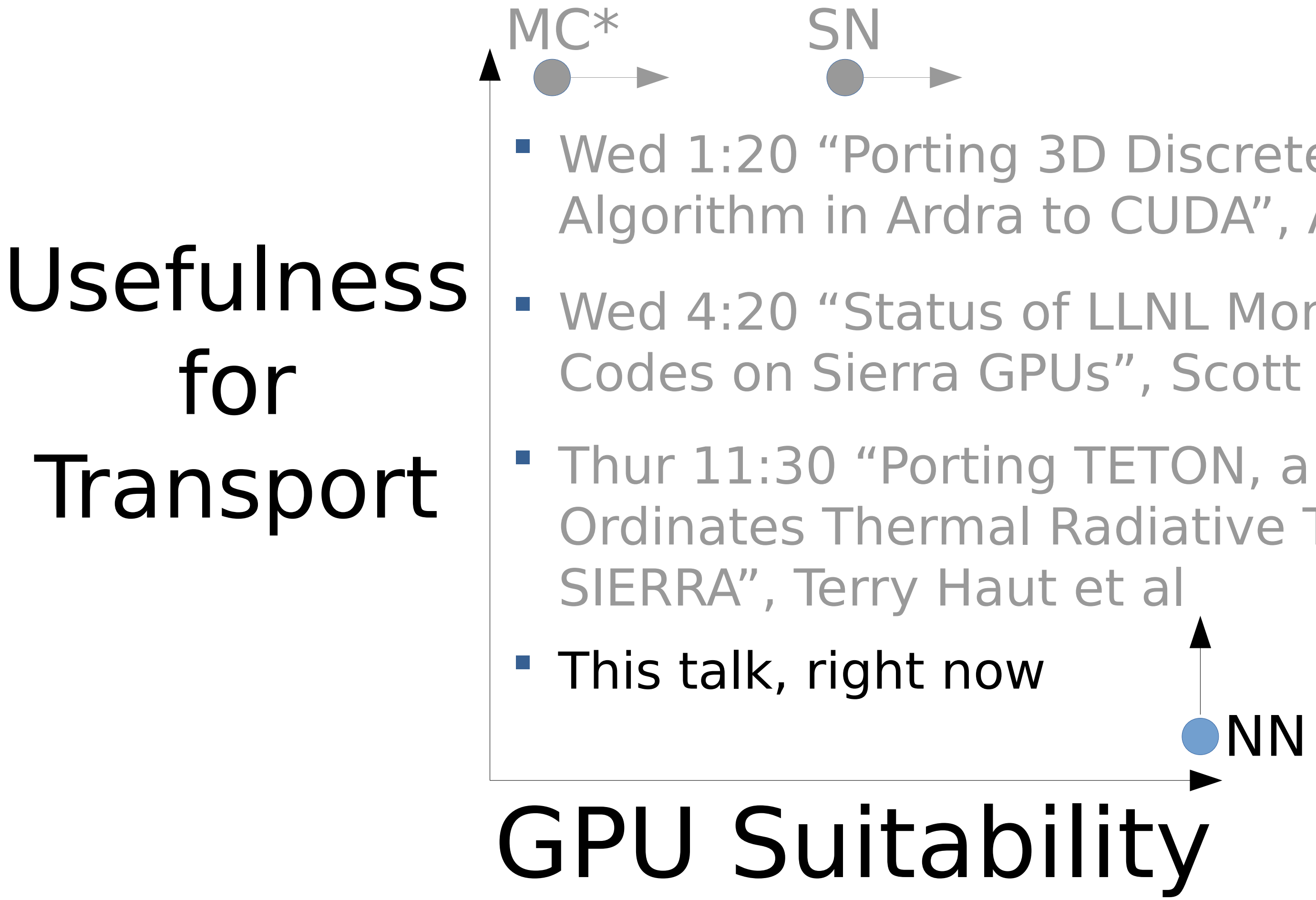
We can port from both sides



- Wed 1:20 “Porting 3D Discrete Ordinates Sweep Algorithm in Ardra to CUDA”, Adam Kunen et al
- Wed 4:20 “Status of LLNL Monte Carlo Transport Codes on Sierra GPUs”, Scott McKinley et al
- Thur 11:30 “Porting TETON, a Discrete-Ordinates Thermal Radiative Transfer Code, to SIERRA”, Terry Haut et al

***History-based* Monte Carlo

We can port from both sides



- Wed 1:20 “Porting 3D Discrete Ordinates Sweep Algorithm in Ardra to CUDA”, Adam Kunen et al
- Wed 4:20 “Status of LLNL Monte Carlo Transport Codes on Sierra GPUs”, Scott McKinley et al
- Thur 11:30 “Porting TETON, a Discrete-Ordinates Thermal Radiative Transfer Code, to SIERRA”, Terry Haut et al
- This talk, right now

**History-based* Monte Carlo

1D is the easiest place to start

- Slab geometry neutron transport

$$\mu \frac{\partial \psi(z,\mu)}{\partial z} + \Sigma_t(z)\psi(z,\mu) = 2\pi \int_{-1}^{1} d\mu' \Sigma_s(z,\mu_0)\psi(z,\mu') + \frac{1}{2}\left[\nu \Sigma_f(z)\phi(z) + Q_{ext}(z)\right]$$

1D is the easiest place to start

- Slab geometry neutron transport

$$\mu\frac{\partial\psi(z,\mu)}{\partial z}+\Sigma_t(z)\psi(z,\mu)=2\pi\int_{-1}^{1}d\mu'\Sigma_s(z,\mu_0)\psi(z,\mu')+\frac{1}{2}\left[\nu\Sigma_f(z)\phi(z)+Q_{ext}(z)\right]$$

- Discrete ordinates

$$\frac{\mu_m}{l_j}\left(\psi_{m,j+\frac{1}{2}}^{(l+1)}-\psi_{m,j-\frac{1}{2}}^{(l+1)}\right)+\Sigma_{t,j}\psi_{m,j}^{(l+1)}=\frac{1}{2}\hat{Q}_{m,j}^{(l)}$$

1D is the easiest place to start

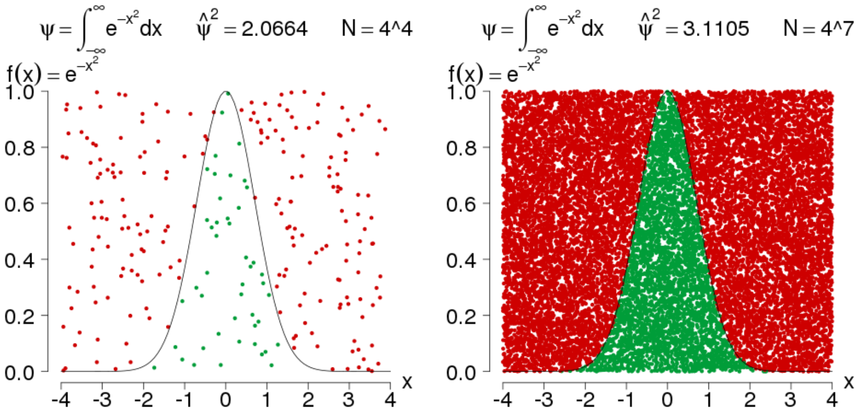
- Slab geometry neutron transport

$$\mu\frac{\partial\psi(z,\mu)}{\partial z}+\Sigma_t(z)\psi(z,\mu)=2\pi\int_{-1}^{1}d\mu'\Sigma_s(z,\mu_0)\psi(z,\mu')+\frac{1}{2}\left[\nu\Sigma_f(z)\phi(z)+Q_{ext}(z)\right]$$

- Discrete ordinates

$$\frac{\mu_m}{l_j}\left(\psi^{(l+1)}_{m,j+\frac{1}{2}}-\psi^{(l+1)}_{m,j-\frac{1}{2}}\right)+\Sigma_{t,j}\psi^{(l+1)}_{m,j}=\frac{1}{2}\hat{Q}^{(l)}_{m,j}$$

- Monte Carlo



1D is the easiest place to start

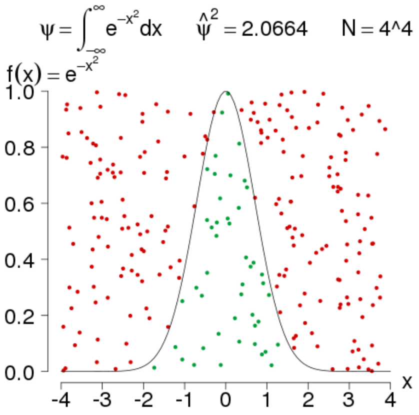
- Slab geometry neutron transport

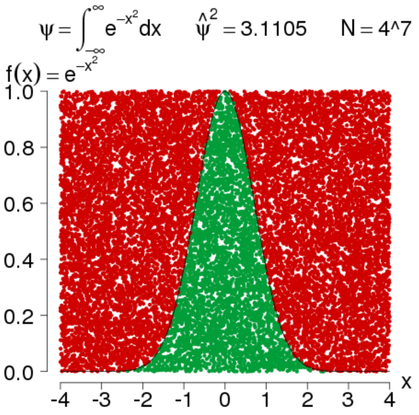
$$\mu\frac{\partial\psi(z,\mu)}{\partial z}+\Sigma_t(z)\psi(z,\mu)=2\pi\int_{-1}^{1}d\mu'\Sigma_s(z,\mu_0)\psi(z,\mu')+\frac{1}{2}\left[\nu\Sigma_f(z)\phi(z)+Q_{ext}(z)\right]$$

- Discrete ordinates

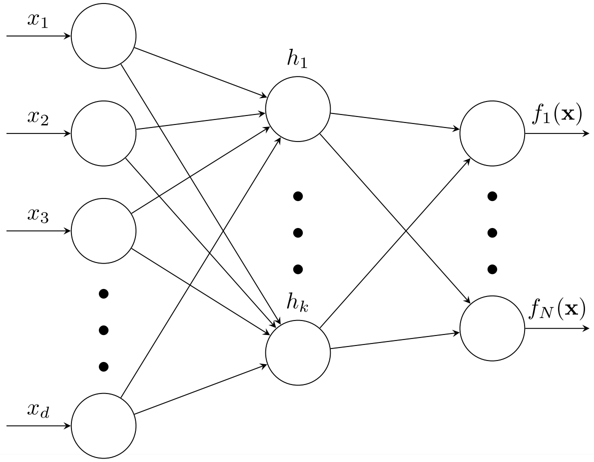
$$\frac{\mu_m}{l_j}\left(\psi^{(l+1)}_{m,j+\frac{1}{2}}-\psi^{(l+1)}_{m,j-\frac{1}{2}}\right)+\Sigma_{t,j}\psi^{(l+1)}_{m,j}=\frac{1}{2}\hat{Q}^{(l)}_{m,j}$$

- Monte Carlo





- Neural network

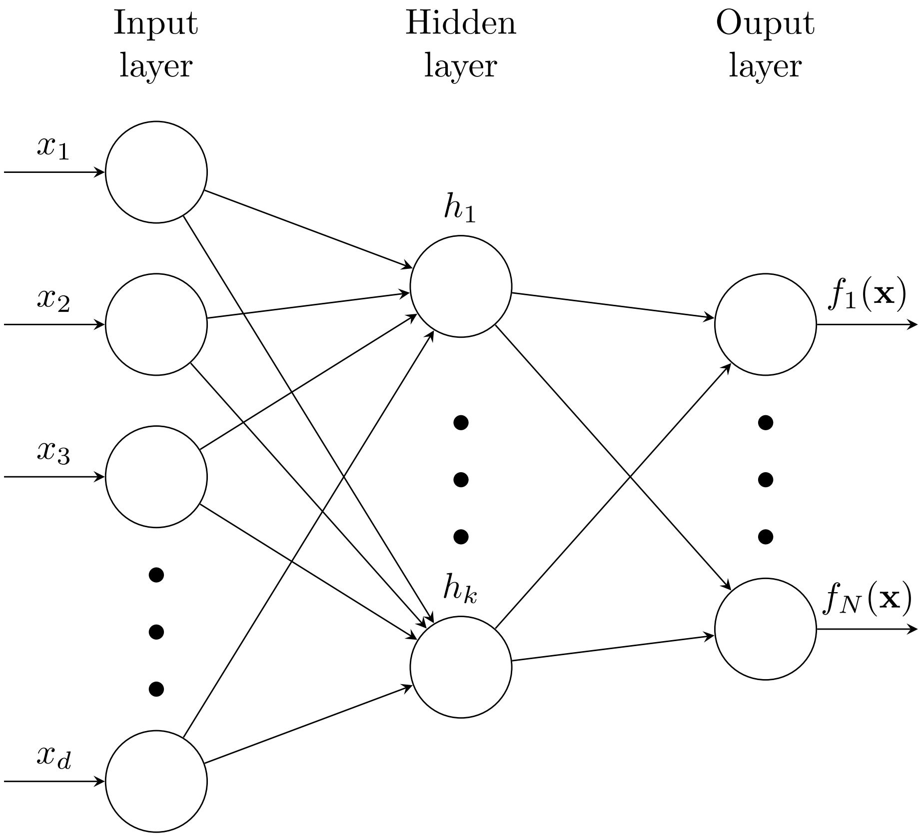


A neural network is a function approximation technique

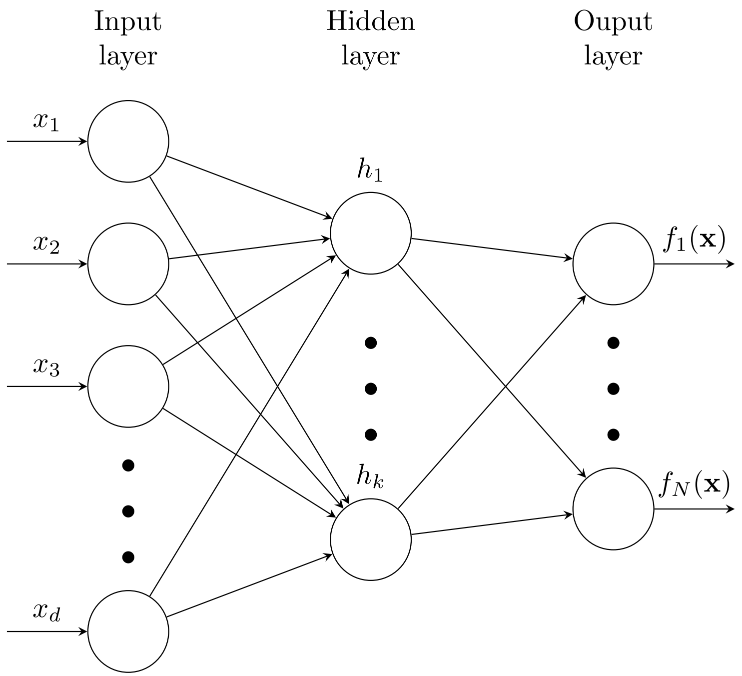
- Input $\mathbf{x} \in \mathbb{R}^d$
- Output $f(\mathbf{x}) \in \mathbb{R}^N$
- Non-linear activation function

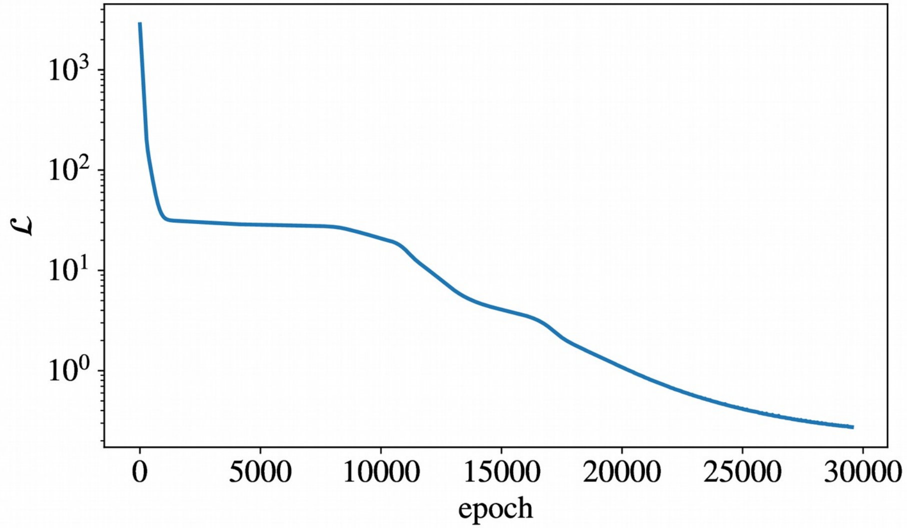
$$h_i = \sigma(\mathbf{w}_i^T \mathbf{x} + b_i)$$

- Weights $\mathbf{w}_i \in \mathbb{R}^n$
- Biases b_i



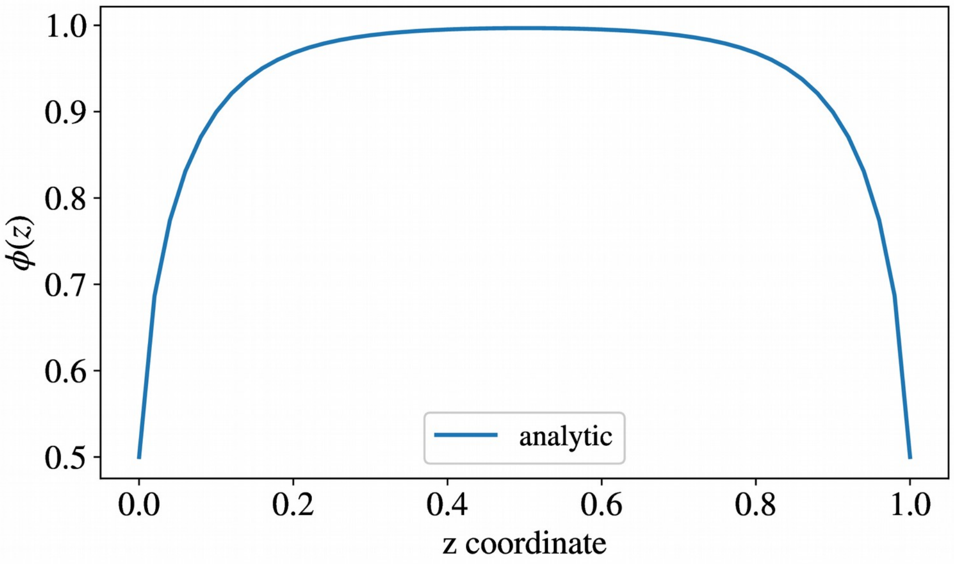
The neural network minimizes a loss function



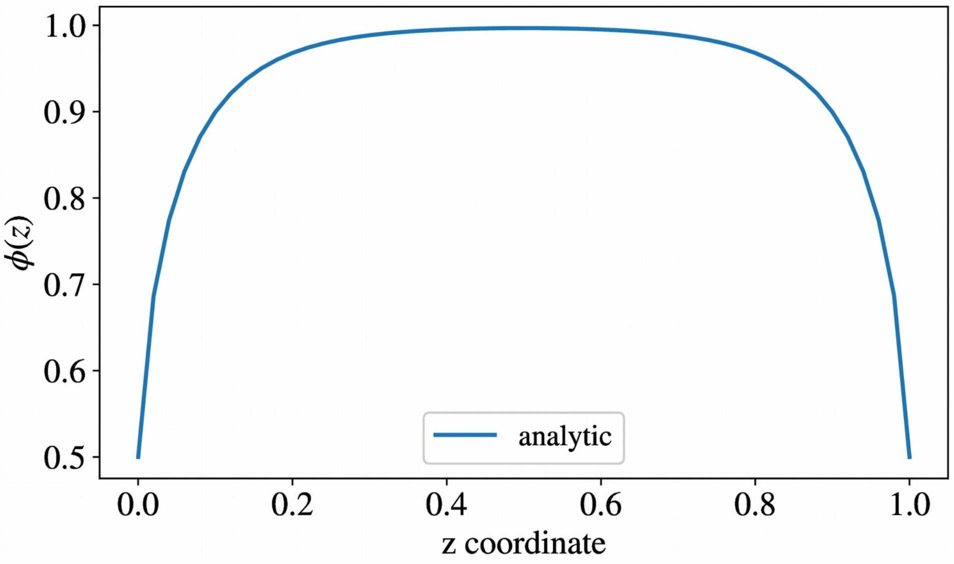


$$\mathcal{L} = \left\| \nabla\hat{\mathbf{\Psi}}\mathrm{diag}(\boldsymbol{\mu}) + \Sigma_t\hat{\mathbf{\Psi}} - \frac{1}{2}(\Sigma_{s0}\hat{\mathbf{\Phi}}_0 1_N^T - 3\Sigma_{s1}\hat{\mathbf{\Phi}}_1\boldsymbol{\mu}^T - \mathbf{Q})\right\|_F^2$$
$$+ \gamma_L||\hat{\mathbf{\Psi}}^{\mu>0}(z=0) - \mathbf{\Psi}_L||_F^2$$
$$+ \gamma_R||\hat{\mathbf{\Psi}}^{\mu<0}(z=z_{max}) - \mathbf{\Psi}_R||_F^2$$

We tested the neural network using a homogeneous medium, uniform source problem

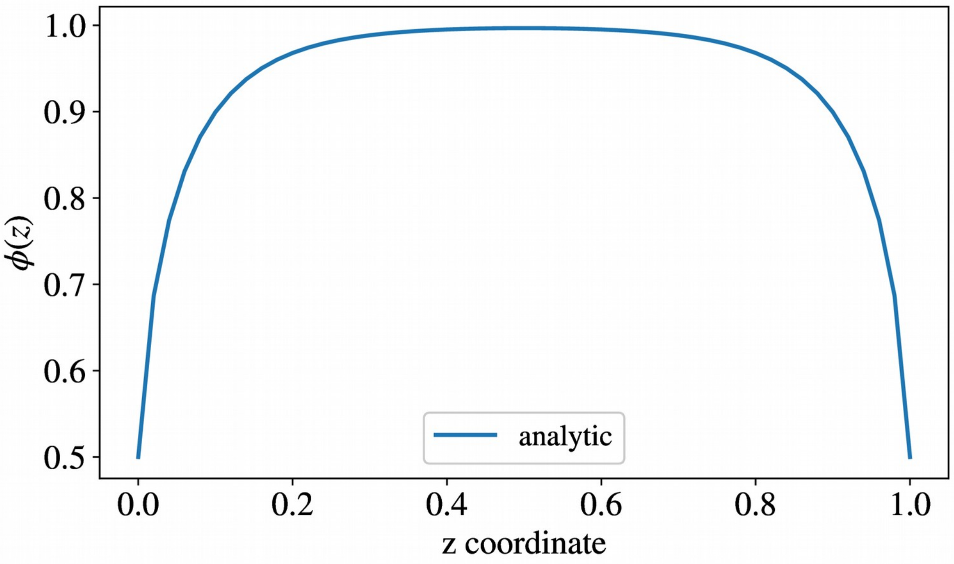


We tested the neural network using a homogeneous medium, uniform source problem



$$\phi(z) = \frac{Q_0}{\Sigma_a} - \frac{Q_0}{2\Sigma_a}\left(e^{-\Sigma_a z} + e^{-\Sigma_a (H-z)}\right) + \frac{Q_0}{2\Sigma_a}\left(z E_1(\Sigma_a z) + (H-z) E_1(\Sigma_a (H-z))\right)$$

We tested the neural network using a homogeneous medium, uniform source problem



Parameter	Value	Description
Σ_t	8	Total cross section (cm^{-1})
Σ_{s0}	0	Total scattering cross section (cm^{-1})
Σ_{s1}	0	Linearly anisotropic cross section (cm^{-1})
Q_0	8	External source magnitude
J_{nn}	50	Number of zones for NN solution
J_{sn}	50	Number of zones for S_N solution
J_{mc}	50	Number of zones for MC solution
NR_{nn}	4	Number of ordinates for NN solution
NR_{sn}	4	Number of ordinates for S_N solution
NP	1e6	Number of particles for MC solution
ϵ_{nn}	1e-6	Convergence criterion value for NN solution
ϵ_{sn}	1e-13	Convergence criterion value for S_N solution
k	5	Number of hidden layer nodes in NN

$$\phi(z) = \frac{Q_0}{\Sigma_a} - \frac{Q_0}{2\Sigma_a}\left(e^{-\Sigma_a z} + e^{-\Sigma_a(H-z)}\right) + \frac{Q_0}{2\Sigma_a}\left(zE_1(\Sigma_a z) + (H-z)E_1(\Sigma_a(H-z))\right)$$

The neural network is correct* and fast**

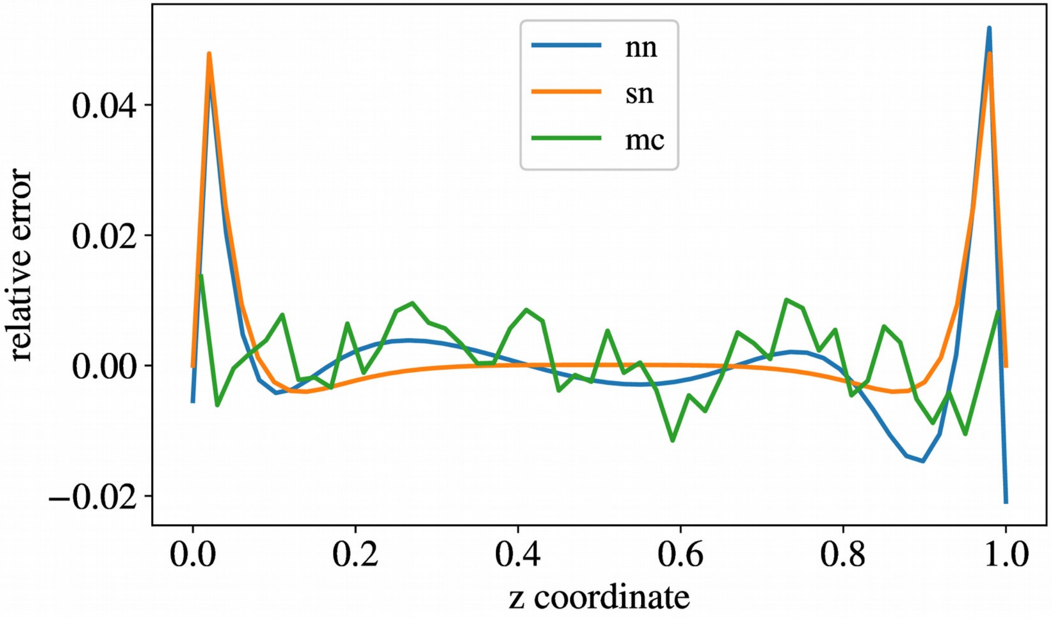
The neural network is correct* and fast**

Max relative error

$$r_m = \max_j \frac{|\phi_j - \hat{\phi}_j|}{\phi_j}$$

$$m \in \{\mathrm{nn},\ \mathrm{sn},\ \mathrm{mc}\}$$

Solution $\begin{cases} \hat{\phi}_j & \text{Model} \\ \phi_j & \text{Analytic} \end{cases}$



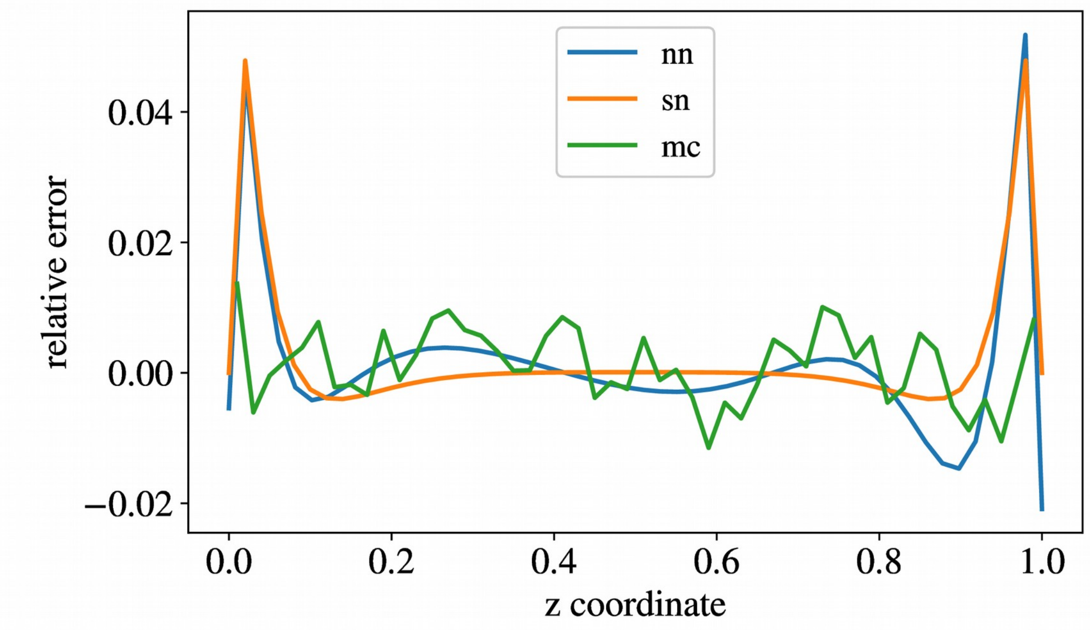
The neural network is correct* and fast**

Max relative error

$$r_m = \max_j \frac{|\phi_j - \hat{\phi}_j|}{\phi_j}$$

$$m \in \{\text{nn, sn, mc}\}$$

Solution
$$\begin{cases} \hat{\phi}_j & \text{Model} \\ \phi_j & \text{Analytic} \end{cases}$$



***Conservation and symmetry are not preserved**

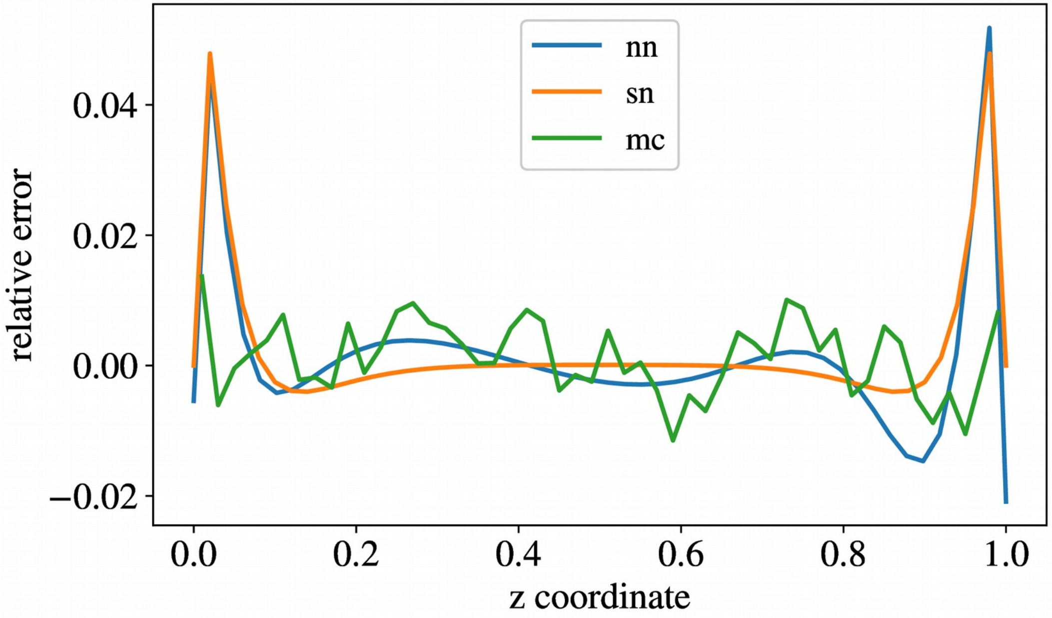
The neural network is correct* and fast**

Max relative error $$r_m = \max_j \frac{|\phi_j - \hat{\phi}_j|}{\phi_j}$$

$$m \in \{\text{nn, sn, mc}\}$$

Solution $$\begin{cases} \hat{\phi}_j & \text{Model} \\ \phi_j & \text{Analytic} \end{cases}$$

***Conservation and symmetry are not preserved**



Algorithm	Max. rel. err.	Runtime (s)
NN Train	-	6.97e+01
NN Pred	0.051807	1.39e-04
SN	0.047869	4.39e-03
MC	0.013654	2.77e+00

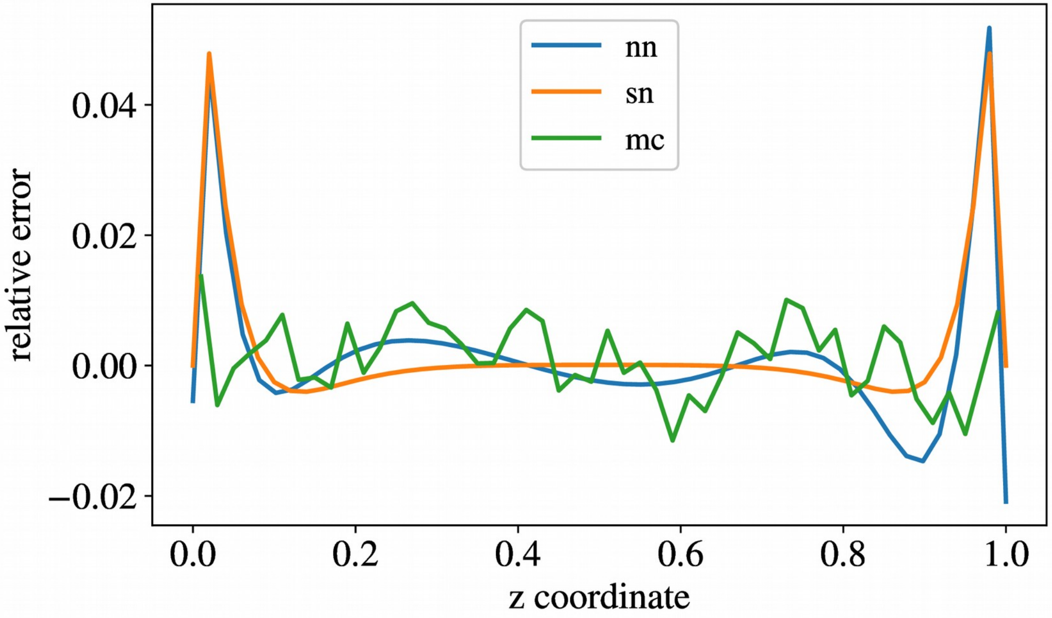
The neural network is correct* and fast**

Max relative error

$$r_m = \max_j \frac{|\phi_j - \hat{\phi}_j|}{\phi_j}$$

$$m \in \{\text{nn, sn, mc}\}$$

Solution $\begin{cases} \hat{\phi}_j & \text{Model} \\ \phi_j & \text{Analytic} \end{cases}$



Algorithm	Max. rel. err.	Runtime (s)
NN Train	-	6.97e+01
NN Pred	0.051807	1.39e-04
SN	0.047869	4.39e-03
MC	0.013654	2.77e+00

***Conservation and symmetry are not preserved**

****If training can be amortized**

Thanks for the help!

- Thanks Kyle Bilton for working with me, Jasmina Vujic for inspiring us, and Patrick Brantley for suggesting the topic, providing initial guidance, and providing feedback on this talk

- References

  - P. S. Brantley. Spatial treatment of the slab-geometry discrete ordinates equations using artificial neural networks. Technical Report UCRL-JC-143205. Lawrence Livermore National Laboratory. Livermore, California, September 2001.
  - Michael M. Pozulp. 1D Transport Using Neural Nets, SN, and MC. LLNL-CONF-772639. M&C 2019. Portland, Oregon (August 25-29, 2019).

- Ongoing research
  - Bob Anderson (LLNL), machine learning for transport
  - Todd Palmer & students (OSU), machine learning and neural nets in 2D

Lawrence Livermore
National Laboratory