Physics-guided Machine Learning:

Opportunities in Combining Physical Knowledge with Data Science for Weather and Climate Sciences

Anuj Karpatne

Assistant Professor, Computer Science

Virginia Tech

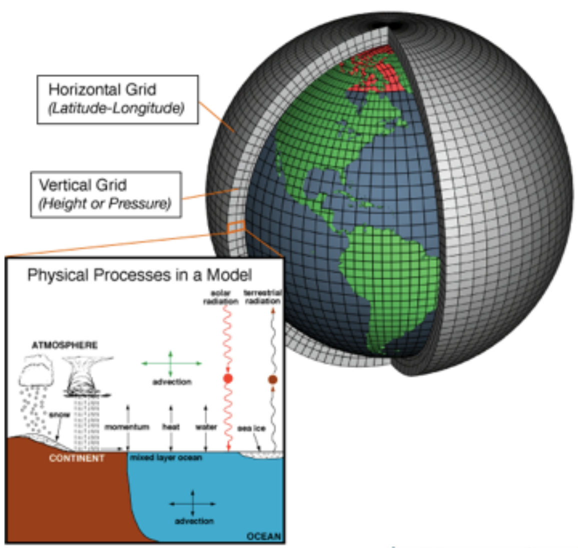
Torgersen Hall 3160Q,

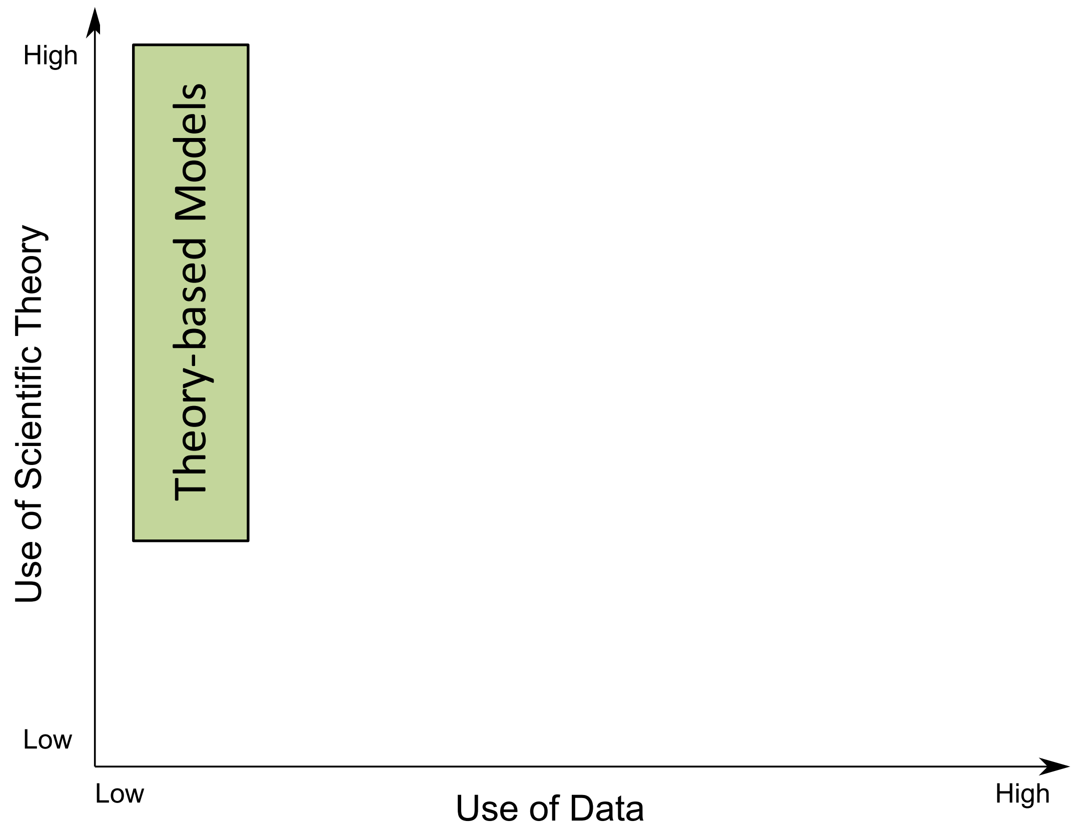
karpatne@vt.edu

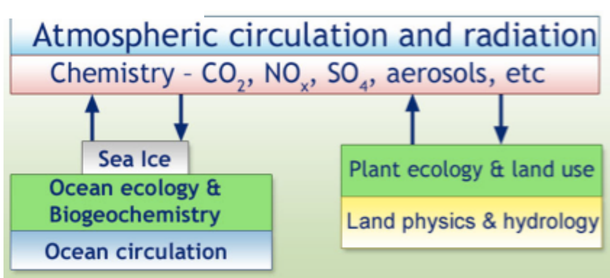
https://people.cs.vt.edu/karpatne/

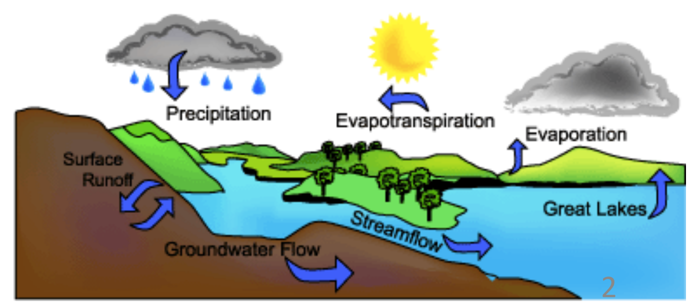
Theory-based vs. Data Science Models

Contain knowledge gaps in describing certain processes (turbulence, groundwater flow)





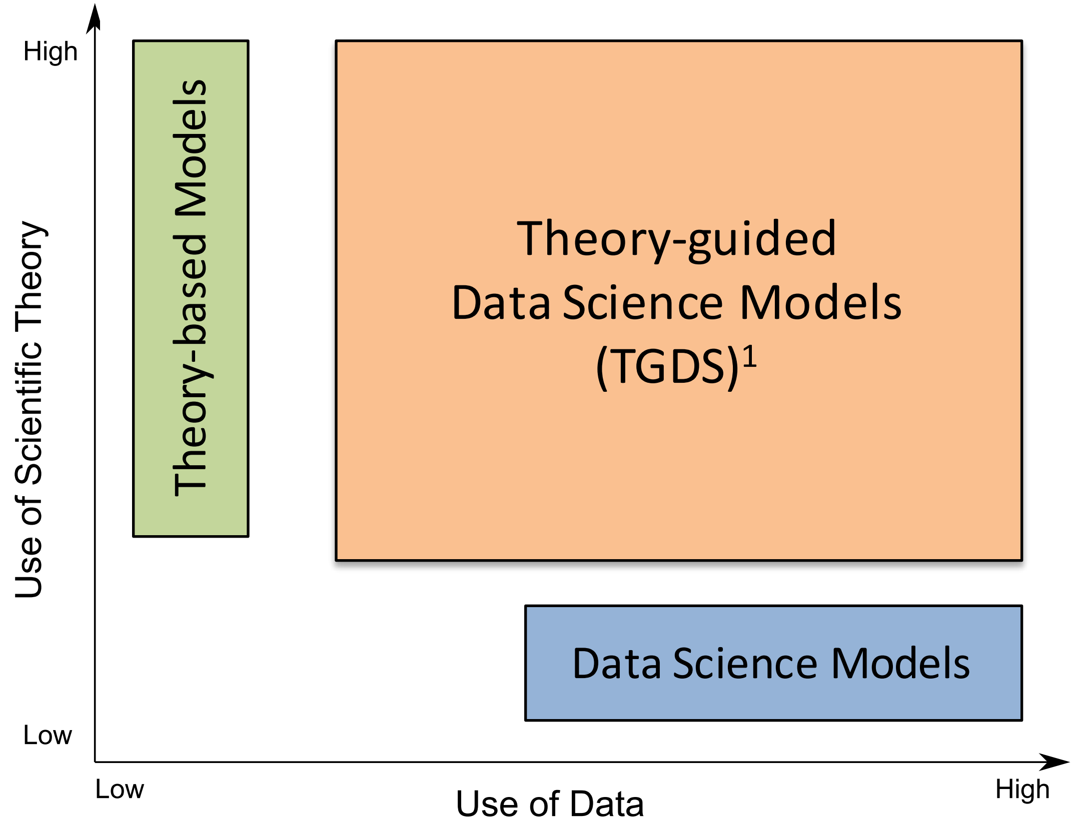




Theory-based vs. Data Science Models

Contain knowledge gaps in describing certain processes (turbulence, groundwater flow)

Take full advantage of data science methods without ignoring the treasure of accumulated knowledge in scientific “theories”



Require large number of representative samples

[1] Karpatne et al. “**Theory-guided data science: A new paradigm for scientific discovery**,” TKDE 2017

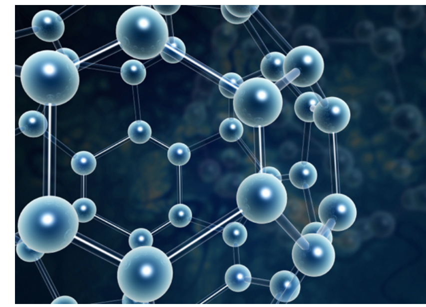
Illustrative Case Study:

Modeling Lake Quality (Temperature)

- Motivation:

Growth and survival of fisheries

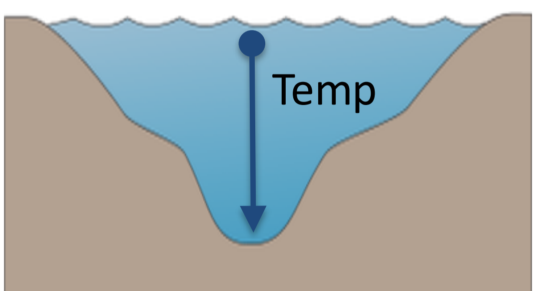
Harmful Algal Blooms



Chemical Constituents: O_2, C, N

- 1-D Model of Temperature:

Target: Temperature of water at every depth in a lake



Input Drivers (observed via meteorology):

Short-wave Radiation, Long-wave Radiation, Air Temperature, Relative Humidity, Wind Speed, Rain, …

Modeling Temperature using Physics-based Models

- Standard Paradigm for Scientific Computing

Input Drivers: solar radiation, air temp., … → x_t

PHY

Hidden States → z_t

Parameters: water clarity, salinity, … → θ

F, G ← **Physics-based Equations**

Target: temperature → y_t

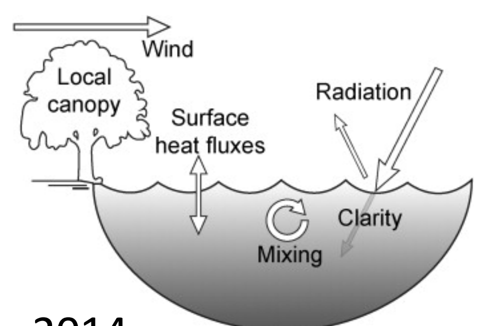
Conservation of mass, momentum, energy

$$\frac{\partial \rho}{\partial t} = -\nabla.(\rho \mathbf{u})$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} = -\nabla.\left(\frac{1}{\rho}(\rho \mathbf{u}) \otimes (\rho \mathbf{u}) + p\mathbf{I}\right) + \rho \mathbf{g}$$

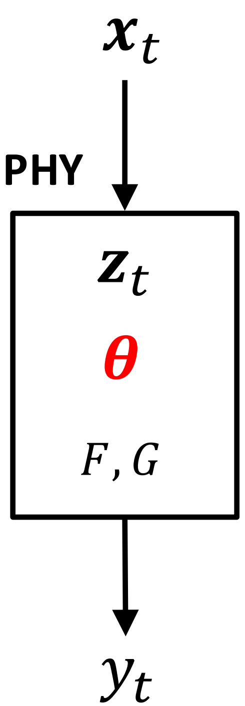
$$\frac{\partial E}{\partial t} = -\nabla.\left(\frac{1}{\rho}(E + p)(\rho \mathbf{u})\right) + \mathbf{u}.\rho \mathbf{g}$$

General Lake Model[1] (GLM)

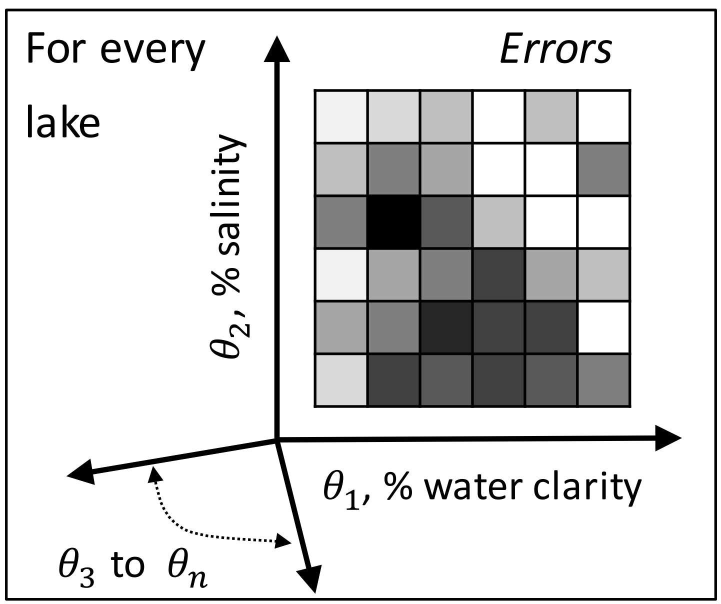


[1]Hipsey et al., 2014

Limitations of Physics-based Models

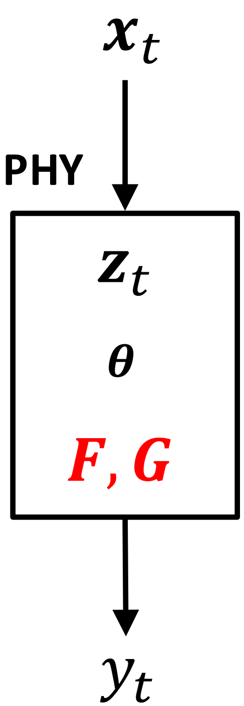


- Unknown parameters ($\boldsymbol{\theta}$) need to be "calibrated"
  - *Computationally Expensive*
  - *Easy to overfit*: large number of parameter choices, small number of samples



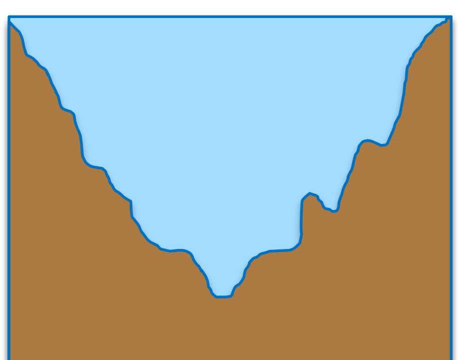
Number of parameter choices: $\sim O(2^n)$

Limitations of Physics-based Models



- Unknown parameters (θ) need to be "calibrated"
  - *Computationally Expensive*
  - *Easy to overfit*: large number of parameter choices, small number of samples

- Incomplete or missing physics (F, G)
  - Physics-based models often use approximate forms to meet "scale-accuracy" trade-off
  - Results in *inherent model bias*

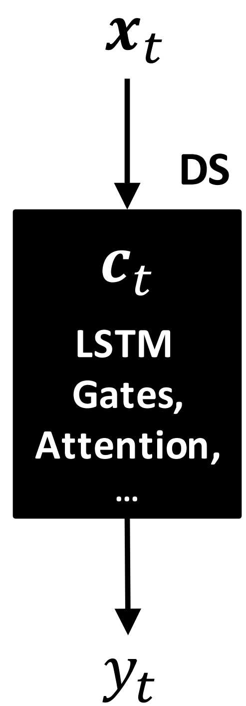


Lake bathymetry often simplified using approximate forms

“Black-box” Data Science Models

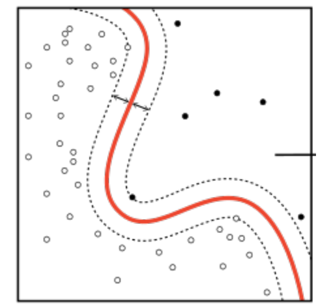
- An alternative to physics-based modeling?

- Requires calibration of model parameters
- Incomplete physics

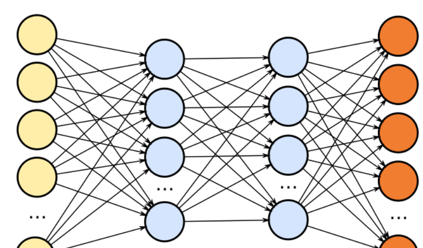


- Require lots of data
- Ignorant of physical knowledge

Choice of model family not governed by physics



Support Vector Machine

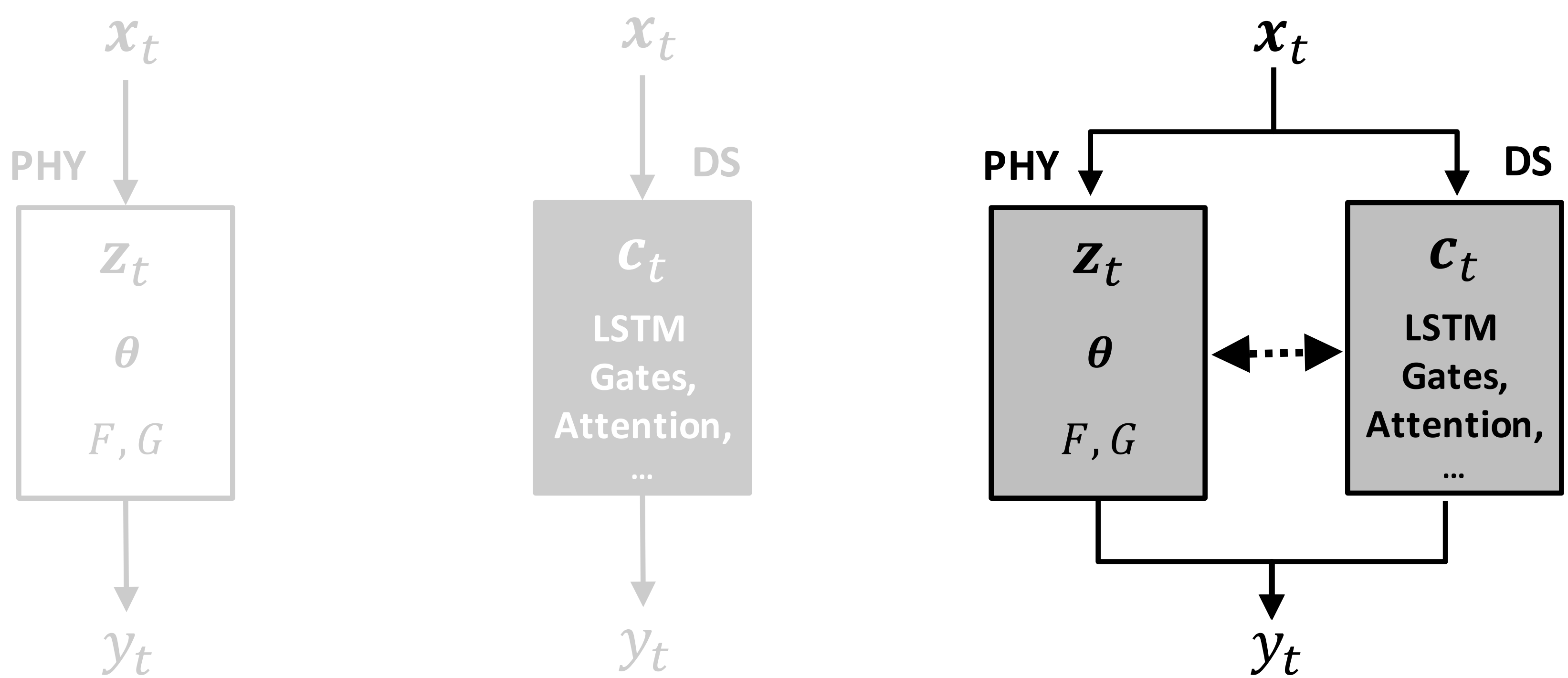


Deep Learning

. . .

Hybrid-Physics-Data Models

- A paradigm shift in data-intensive scientific modeling

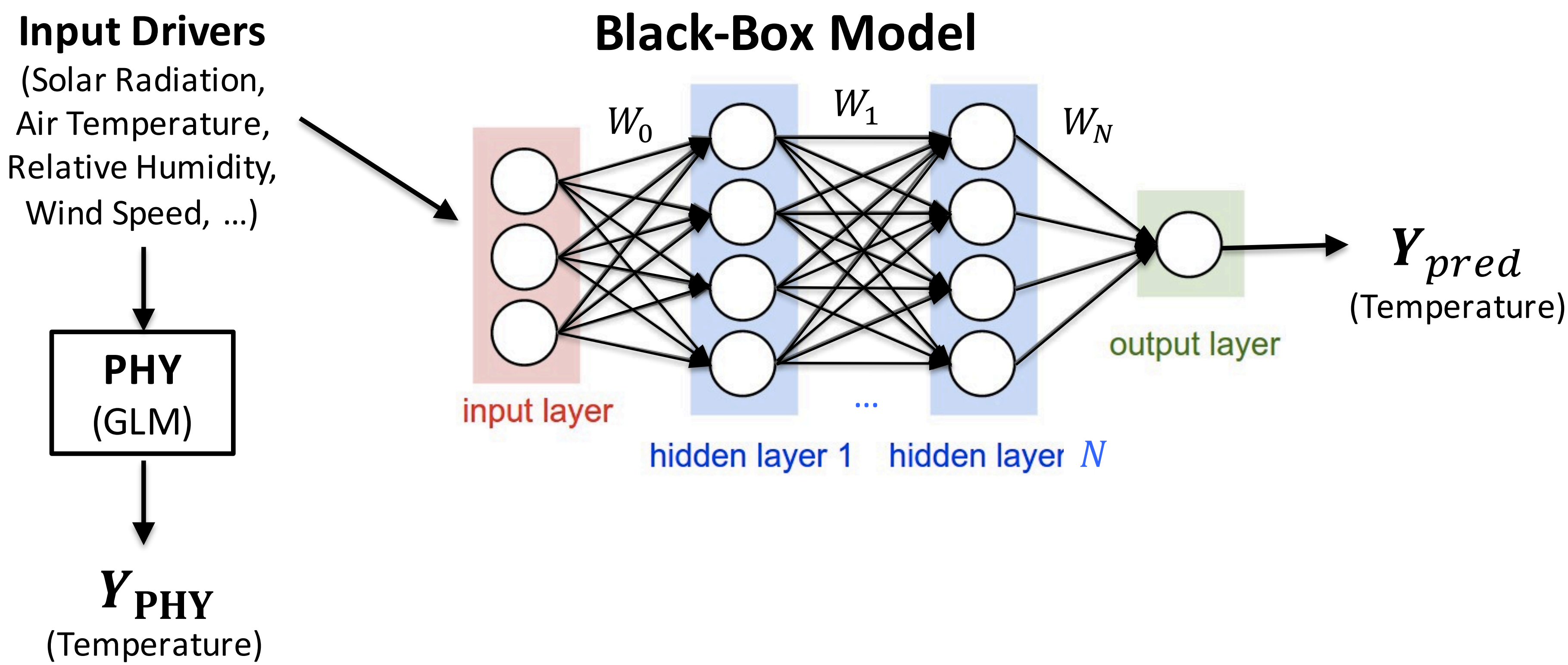


- Requires calibration of model parameters
- Incomplete physics

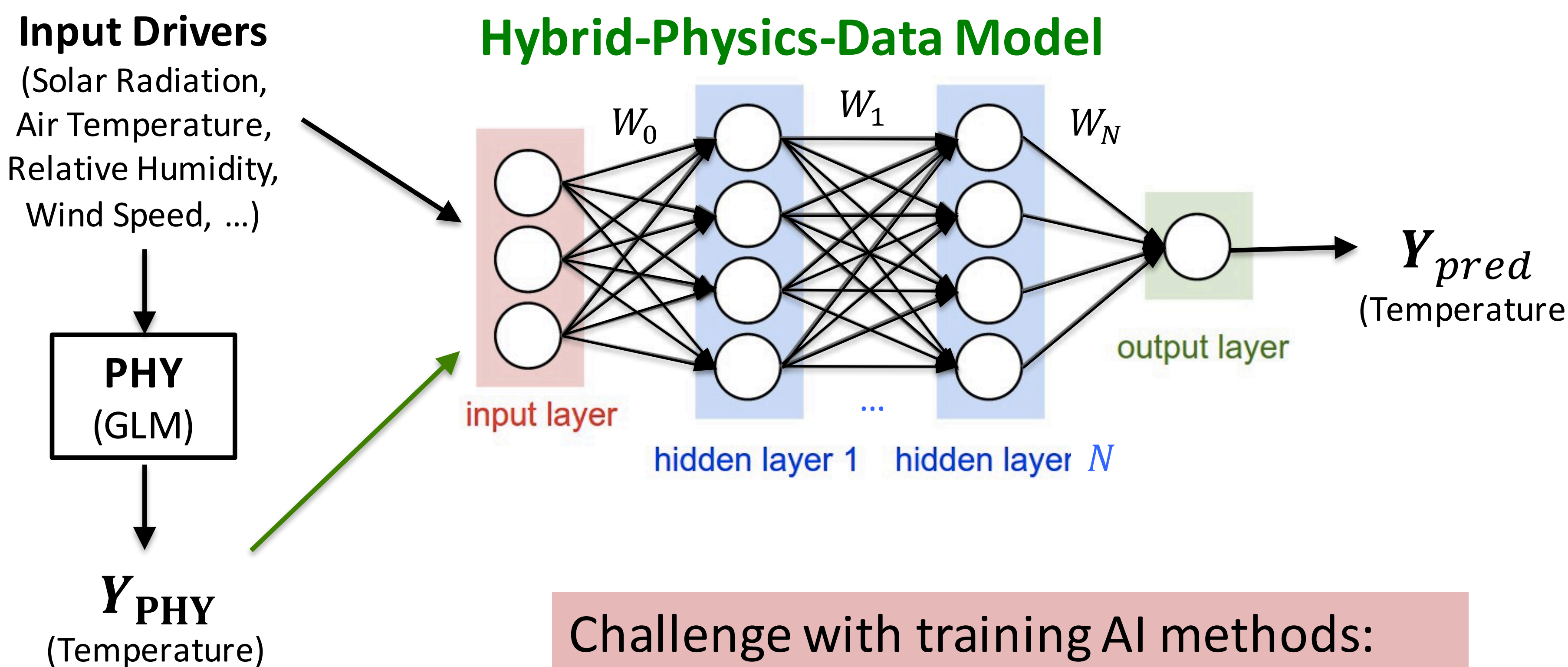
- Require lots of data
- Incoherent with physical knowledge

Overcomes complementary weaknesses of PHY and DS by combining them together

A Generic Framework for Hybrid-Physics-Data (HPD) Modeling:



A Generic Framework for Hybrid-Physics-Data (HPD) Modeling:

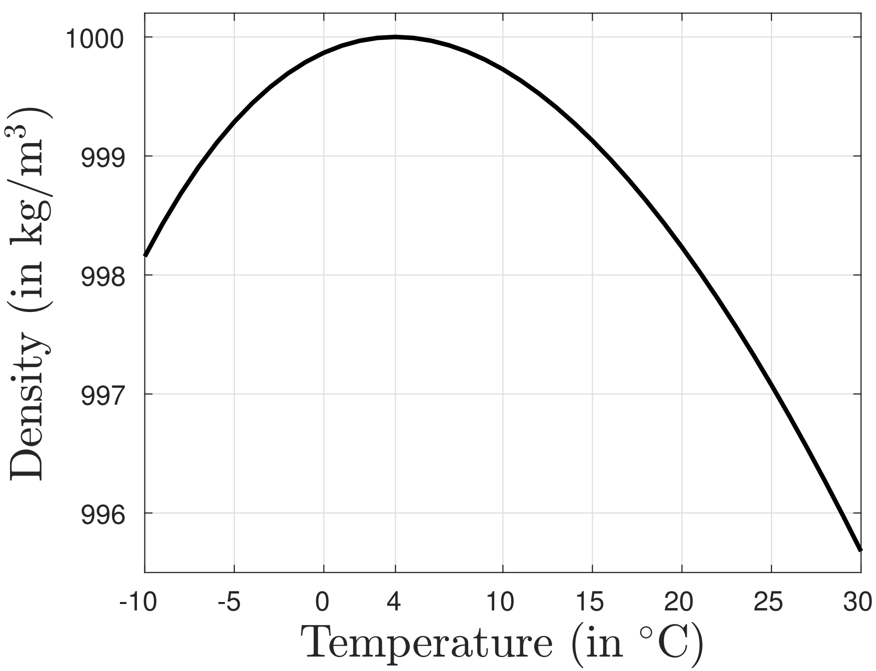


Challenge with training AI methods:

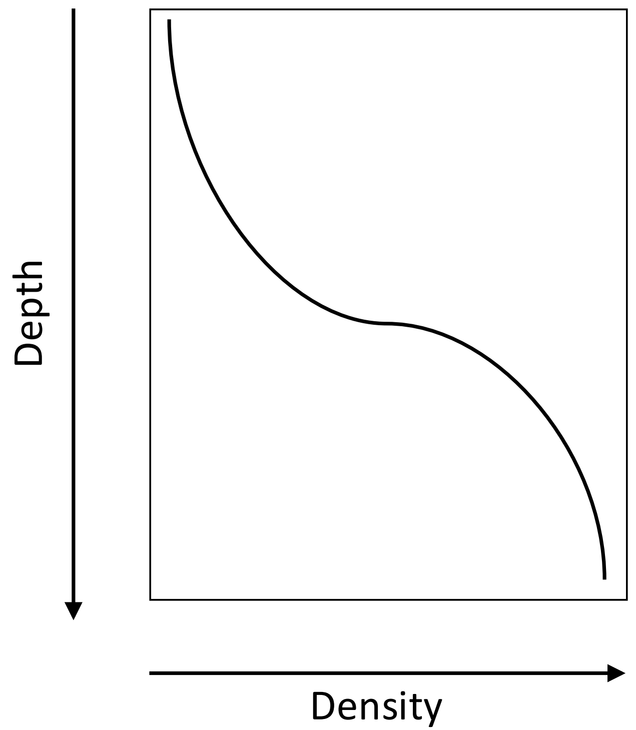
- Y_{pred} may violate **physical relationships** b/w Y and other variables

Physical Relationships of Temperature

Temperature directly related to density of water



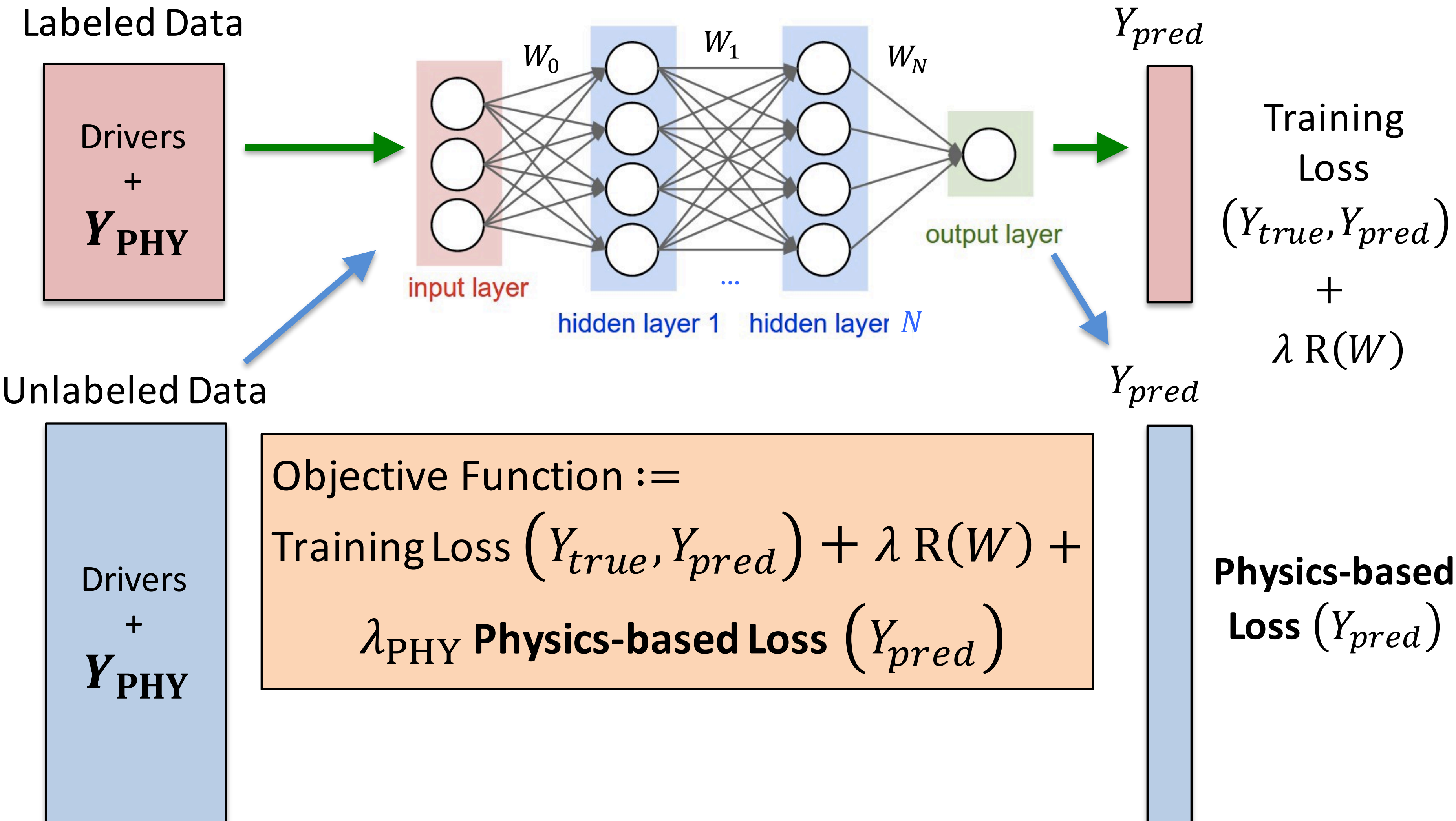
Denser water is at higher depth



Use **physics-based loss functions:**

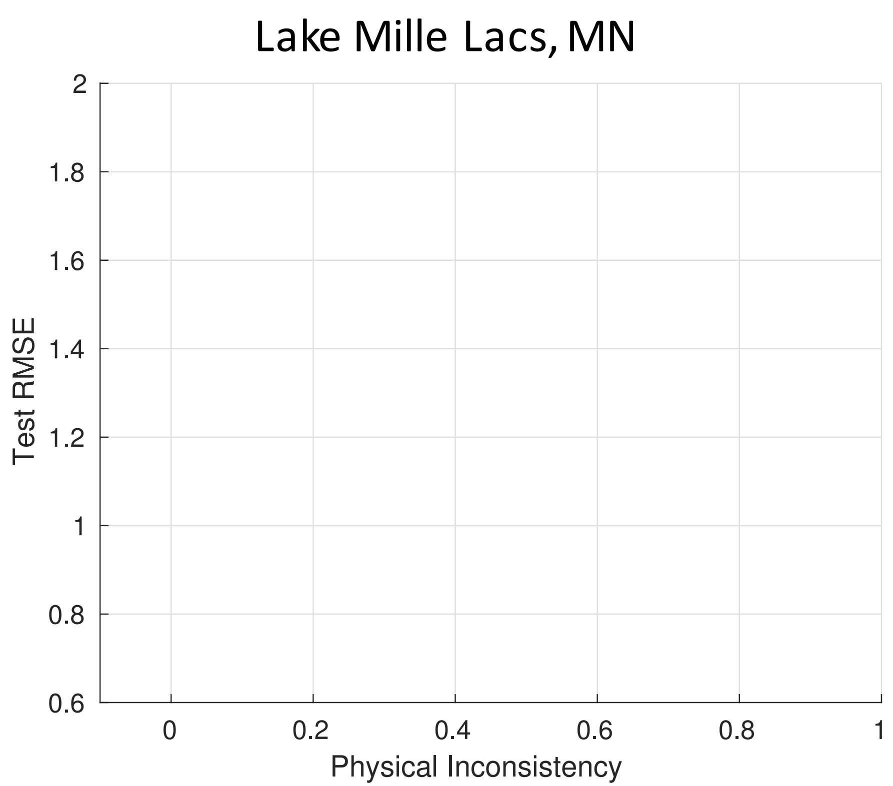
- Measure violations of physical relationships b/w Y_{pred} and other variables
- Does not require labeled data!

Physics-guided Neural Network (PGNN)[1]

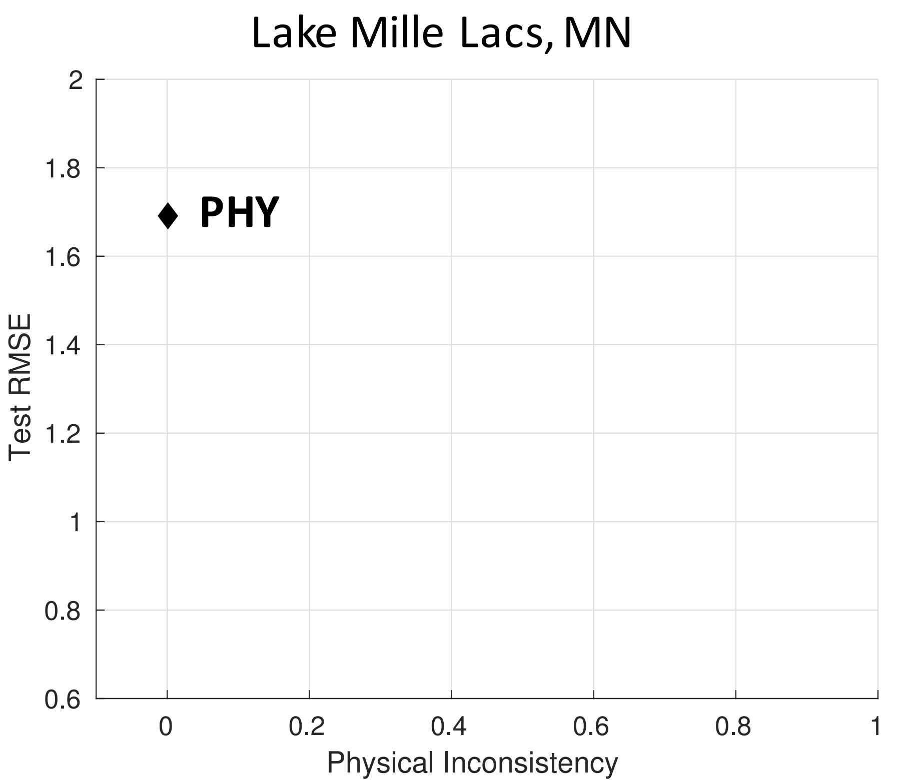


[1]Karpatne et al., "**Physics-guided neural networks (PGNN): An Application in Lake Temperature Modeling**," arXiv: 1710.11431, 2017.

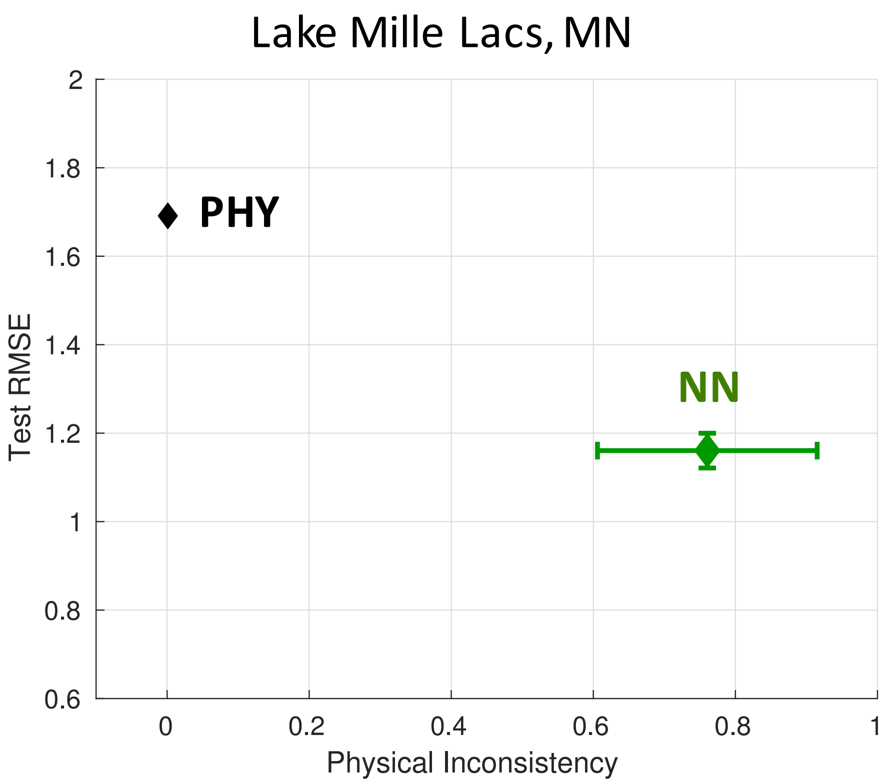
Experimental Results



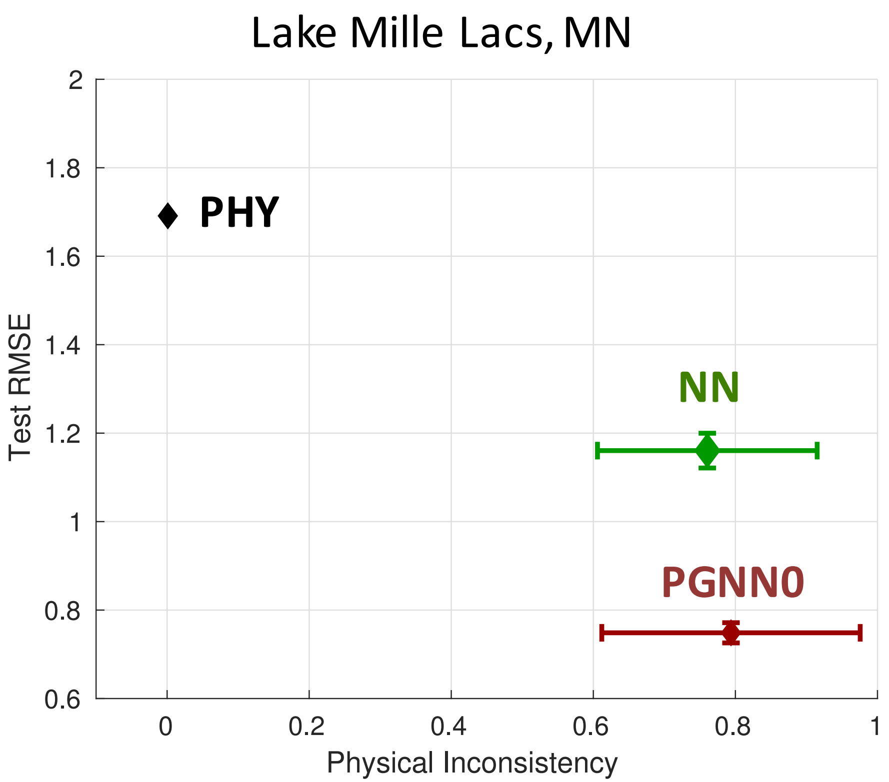
Experimental Results



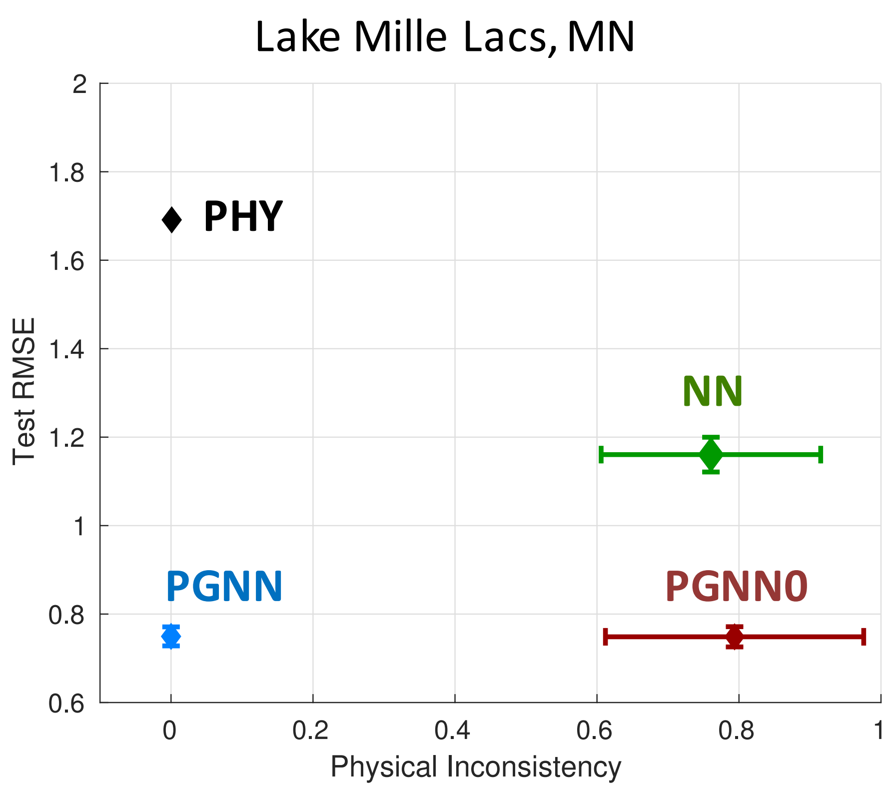
Experimental Results



Experimental Results

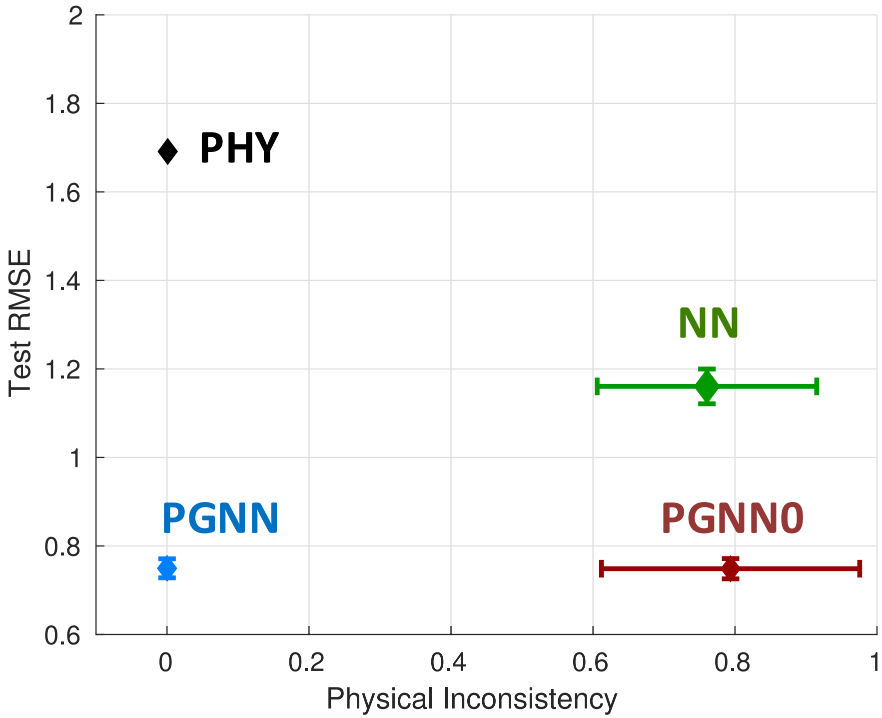


Experimental Results

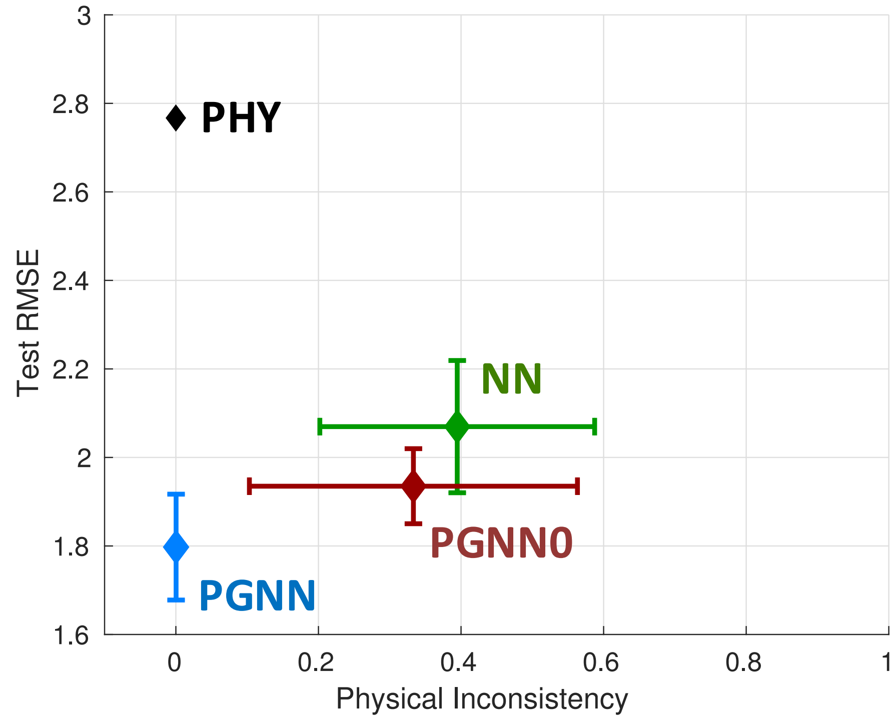


Experimental Results

Lake Mille Lacs, MN

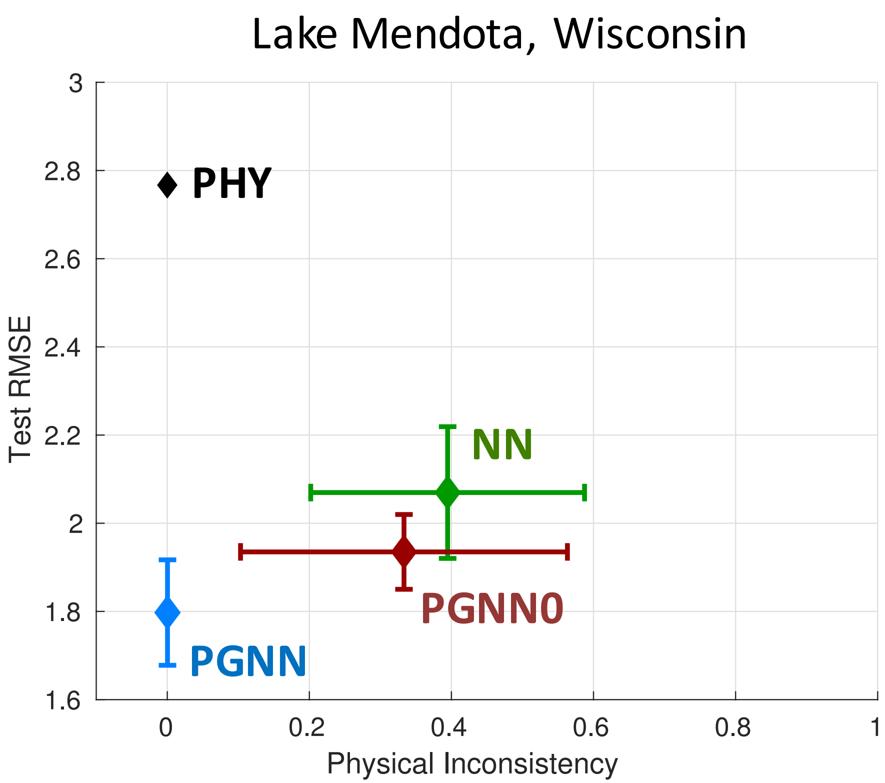


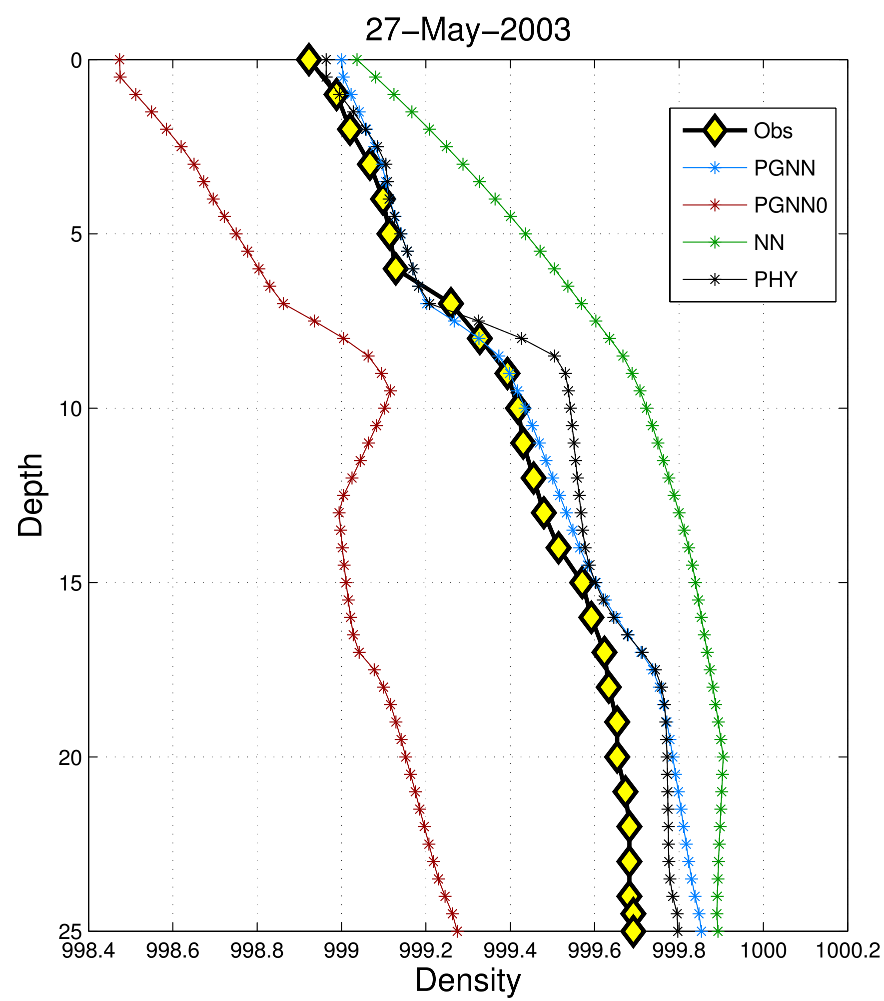
Lake Mendota, Wisconsin



PGNN ensures Generalizability + Physical Consistency

Analyzing Physical Inconsistency

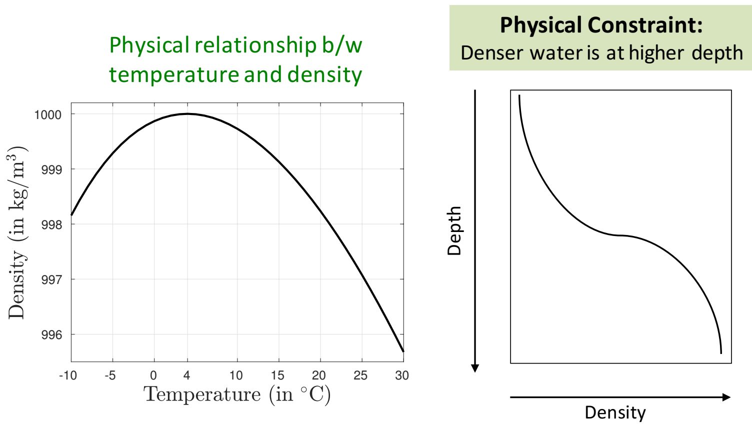




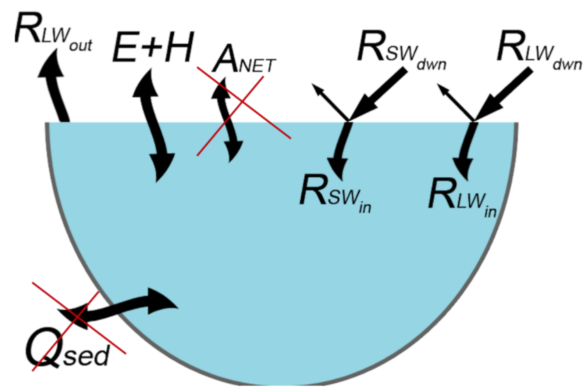
Include **physical consistency** as another evaluation criterion, going beyond standard metrics for test error

Alternate Ways of Incorporating Physics in ML

- Other Physics-based Loss Functions:



Depth-Density Constraint in Multi-layer Perceptron Network

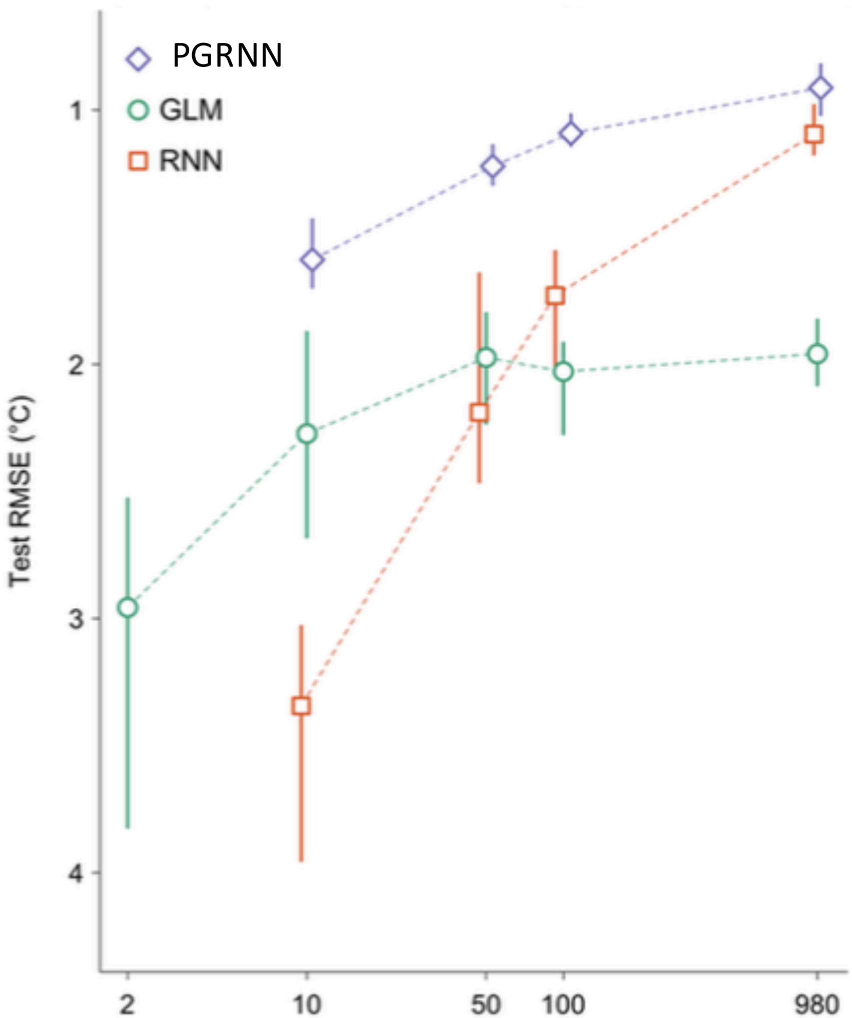


$dU_t/dt = R_{SW}(1-\alpha_{SW}) + R_{LWin}(1-\alpha_{LW}) - R_{LWout} - E - H$

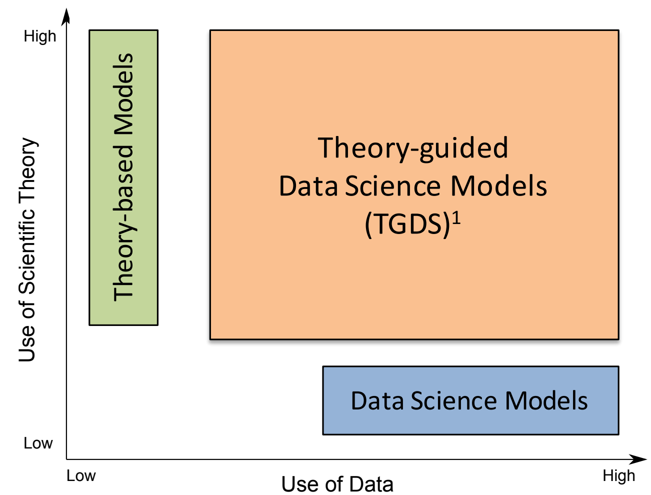
Conservation of Energy in Recurrent Neural Networks

- Pre-training ML models using Physics
  - Train ML methods using physical simulations
  - Fine-tune using observational data

Physics-guided Recurrent Neural Networks (PGRNN)



Number of training samples



AGU 2018 Poster IN41D-0872:
Read et al., *Process-Guided Data-Driven modeling of water temperature: Anchoring predictions with thermodynamic constraints in the Big Data era*

Jia et al., **Physics Guided RNNs for Modeling Dynamical Systems: A Case Study in Simulating Lake Temperature Profiles,** arxiv: 1810.13075, 2018.

Summary and Future Directions

- Research themes in TGDS
  - Physics-guided *Learning* of ML models
    - Loss Functions, Priors, Constraints, …
  - Physics-guided *Design* of ML models
    - Architecture of NN models, LSTM connections, activation functions
  - Pre-training ML models using Physical Simulations
    - Train Using Combination of Physical Simulations + Observations
  - Building Hybrid-Physics-Data Models
    - Rectify Model Outputs, Replace Model Components using ML
  - Inferring Parameters/States in Physics-based Models
    - Parameter Calibration, Data Assimilation