Evaluating Hyperparameter Selection Techniques for the Ratio-Coupled Loss Function

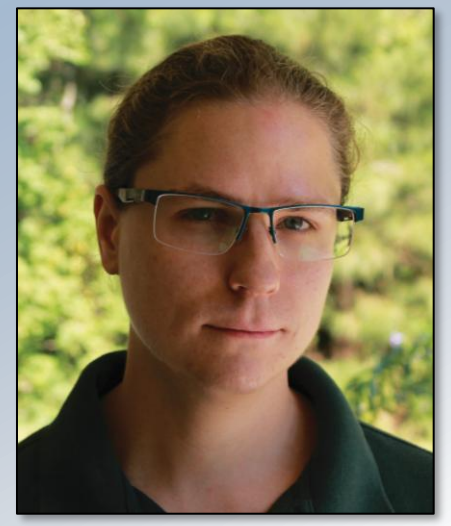
Presented By: Austin Schmidt
Email: sbaustin@uno.edu

Authors
Austin B. Schmidt
Pujan Pokhrel
Md Meftahul Ferdaus
Elias Ioup
Mahdi Abdelguerfi
Julian Simeonov

9/25/2025

Biography
Austin B. Schmidt (M.Sc.)

- In association with the *Canizaro Livingston Gulf States Center for Environmental Informatics* **(GulfSCEI)**
- *SMART Scholar* – US Department of Defense *(DoD)* Scholarship-for-Service
- Research background: ML/AI, timeseries forecasts, surrogate modeling



MDPI Machine Learning and Knowledge Extraction (2022)
Machine learning based restaurant sales forecasting

IEEE OCEANS (2022)
Angle Classification of 3D-printed Container Models Dropped into the Towing Tank

IEEE Journal of Oceanic Engineering (2024)
Forecasting Buoy Observations Using Physics-Informed Neural Networks

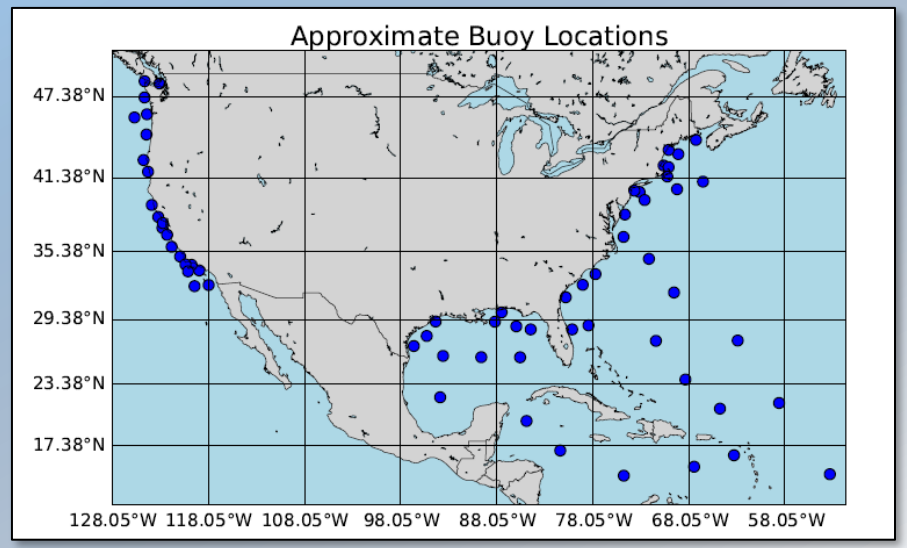
IARIA The First International Conference on Technologies for Marine and Coastal Ecosystems (COCE) (2024)
(1) A Multiple-Location Modeling Scheme for Physics-Regularized Networks: Recurrent Forecasting of Fixed-Location Buoy Observations
(2) Physics-Regularized Buoy Forecasts: A Multi-Hyperparameter Approach Using Bounded Random Search

Royal Meteorological Society (RMetS) (Under Review)
An algorithm for modeling differential processes utilising a ratio-coupled loss

Problem and Motivation

Surrogate Models and Fixed-Location Forecasting

- Fixed-location forecasting allows for surrogate predictions of non-gridded or sparse data
  - Not constrained to the numerical model resolutions
  - Useful in predicting observations for sensor locations
- Numerical models can take a long time to generate high-resolution predictions
  - Surrogate create short-term predictions on demand
- Hyperparameter optimization was shown to be time inefficient
  - *Grid* and *Random* searches showed existence of improved results
- Although promising, previously did **not** compare similar methodologies



Contribution

A Comparison of Methodologies Using a Non-Linear PDE

- ✓ Explored surrogate prediction capability of fixed-locations for the Cahn-Hilliard PDE
- ✓ Compares *convex* and *non-convex* ratio-coupled losses directly
- ✓ First time direct comparisons of hyperparameter searches for ratio-coupled loss function
- ✓ Introduces a novel way to use the optimized-λ approach
- ✓ Justifies further research into surrogate models for approximating previously unseen time series

Methodology

The Cahn-Hilliard Equation

- The Cahn-Hilliard equation models the phase separation process of binary mixtures
- Often used in fluid mechanics to model droplet formation and density of atoms
- In modern work, is coupled with Navier-Stokes equations
- This work investigates surrogate forecasting of fixed-locations in this equation space

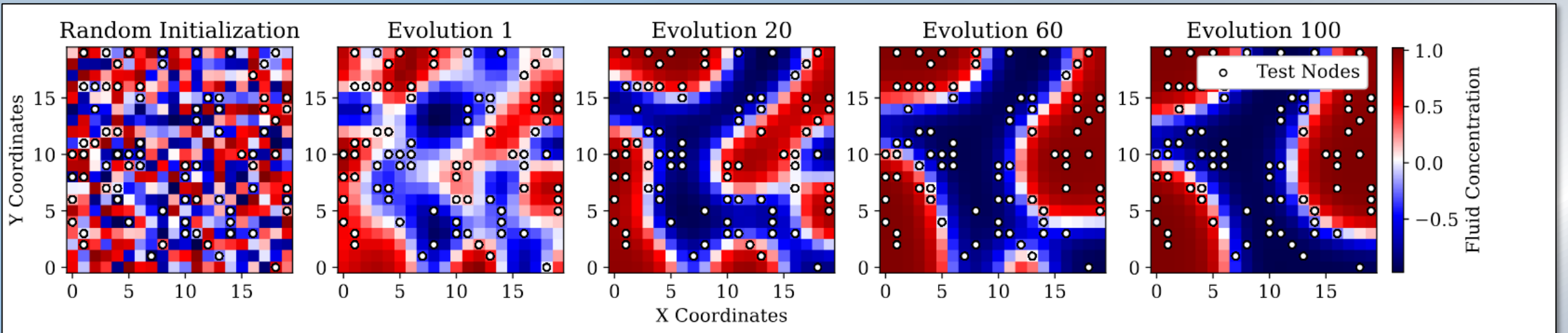


Figure 1. Evolution of the Cahn-Hilliard equation over 100 evolutions. Displays the locations of the 80 reserved observation testing nodes.

Methodology

The Cahn-Hilliard Dataset

Fixed meta-features describe space and time information

Concentration is the sole prediction target – is coupled at training time

Feature	Input?	Output?	Description (Unit)
X	Yes	No	Horizontal location ([0-19])
Y	Yes	No	Vertical location ([0-19])
Time	Yes	No	Timestep value ([0-99])
Concentration	Yes	Yes	Fluid concentration at the current location ([-1,1])

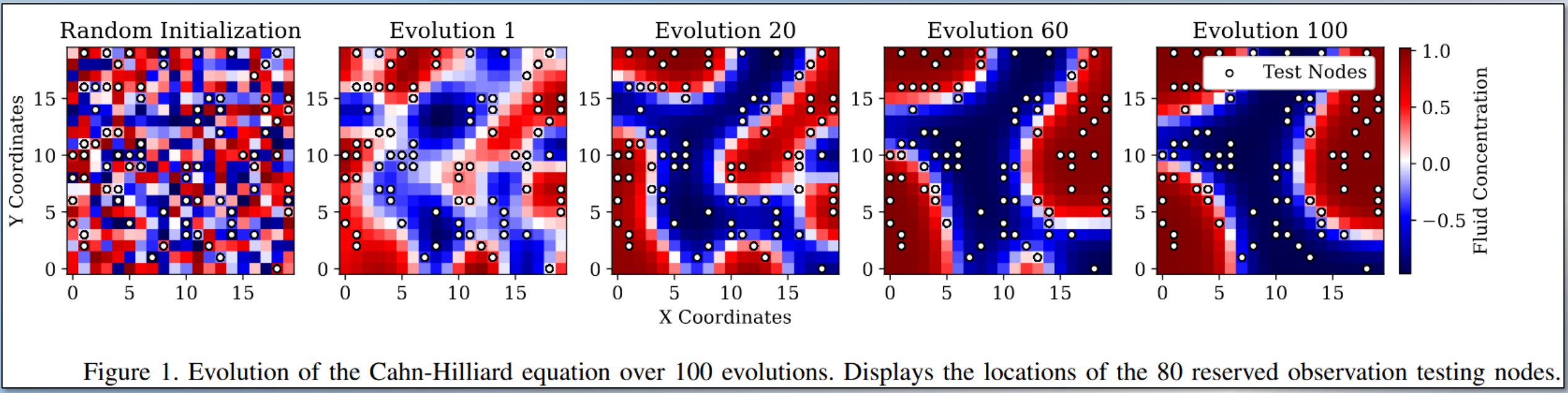


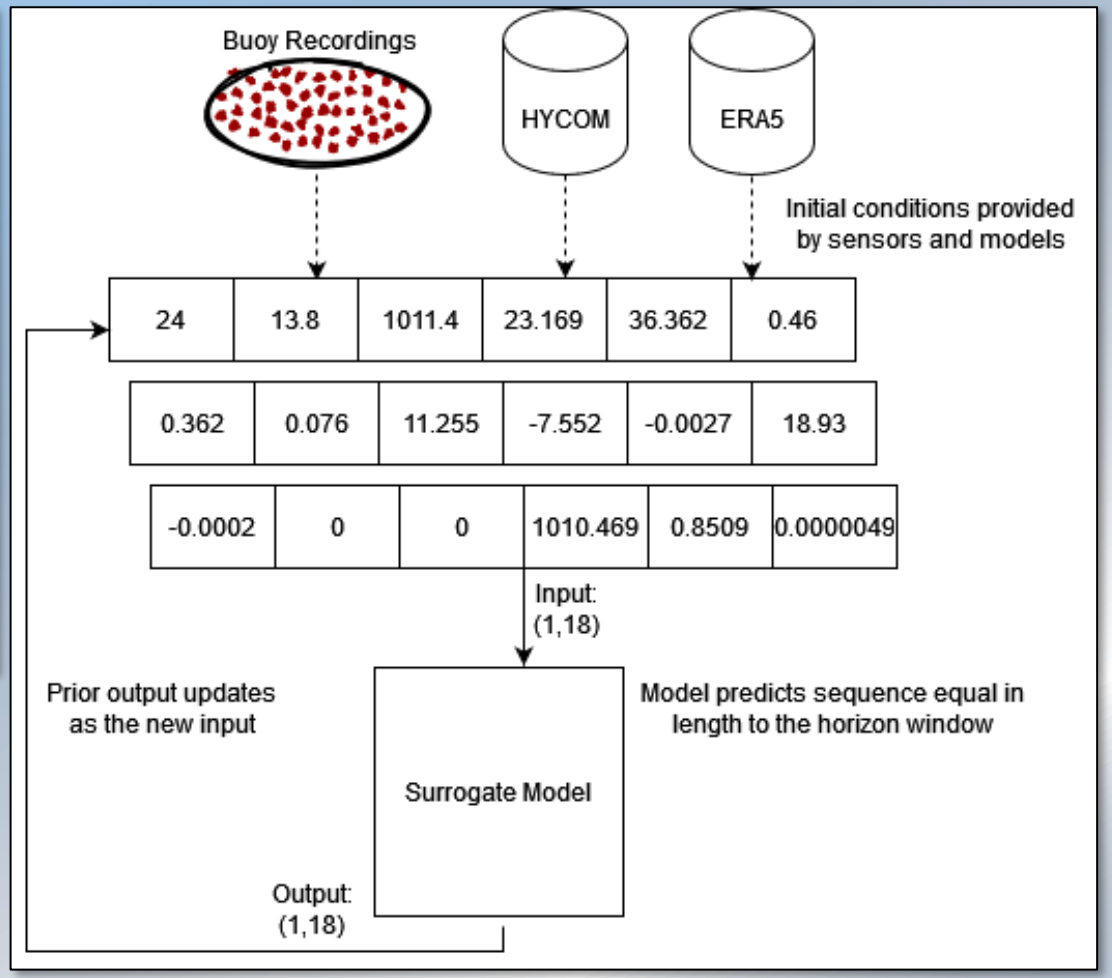
Figure 1. Evolution of the Cahn-Hilliard equation over 100 evolutions. Displays the locations of the 80 reserved observation testing nodes.

Methodology

Model Architecture and Training Specifics

TABLE I. LSTM MODEL ARCHITECTURE BY LAYER. THE TOTAL NUMBER OF TRAINABLE PARAMETERS IS 526,241. N REPRESENTS THE BATCH SIZE.

Layer Type	Shape	Parameters	Activation
Input Layer	(N, 4, 1)	0	None
Reshape	(N, 1, 4)	0	None
LSTM	(N, 1, 256)	267,264	Tanh
Dropout	(N, 1, 256)	0	None
LSTM	(N, 1, 128)	197,120	Tanh
Dropout	(N, 1, 128)	0	None
LSTM	(N, 1, 64)	49,408	Tanh
Dropout	(N, 1, 64)	0	None
LSTM	(N, 1, 32)	12,416	Tanh
Dense	(N, 1)	33	Linear



Methodology

Loss Function Definitions

Residuals captured for the coupled features

$$\Delta_1 = g(\hat{y}, y_o),$$

$$\Delta_2 = g(\hat{y}, y_m),$$

Ratio-coupled loss

$$\Omega_{\text{ratio-coupled loss}} = \lambda * \Delta_1 + (1 - \lambda) * \Delta_2.$$

Huber loss: g(ŷ,y)

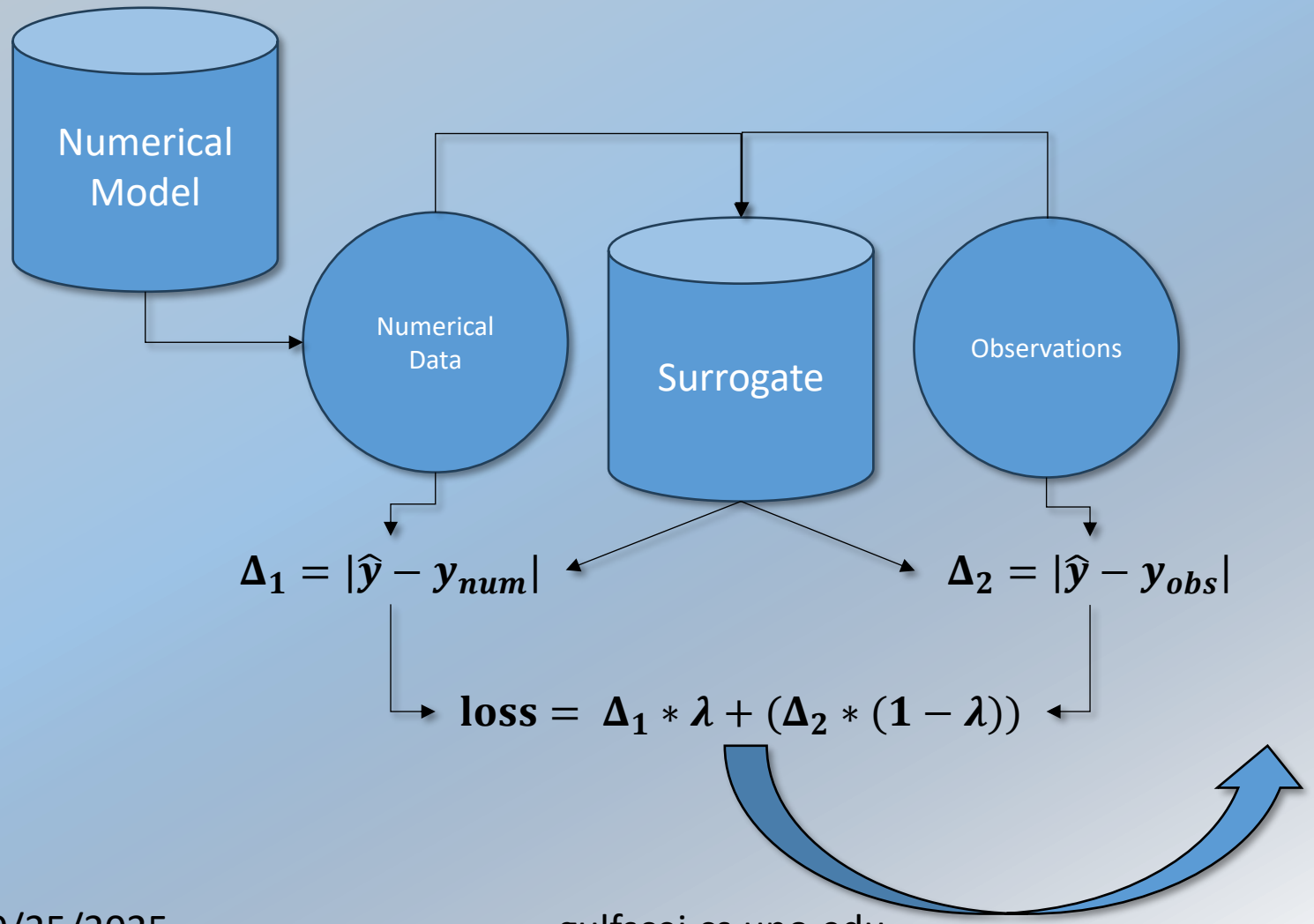
$$g_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta) & \text{if } |a| > \delta, \end{cases},$$

Convex ratio-coupled loss

$$\Omega_{\text{convex ratio-coupled loss}} = (\lambda * \Delta_1)^2 + ((1 - \lambda) * \Delta_2)^2.$$

Methodology

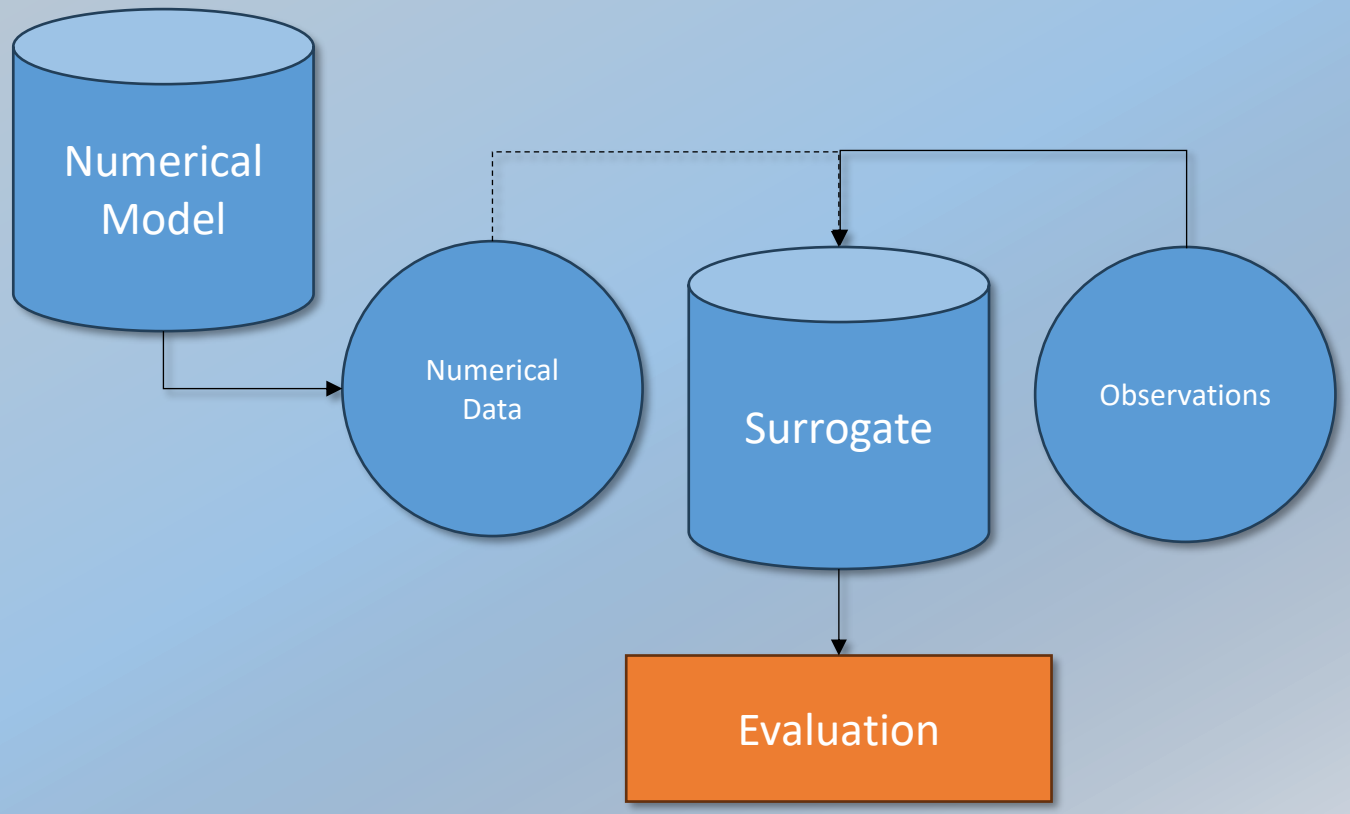
Ratio-Coupled Physics-Regularized Loss Function



λ	Δ_1	Δ_2
0.0	0%	100%
0.5	50%	50%
1.0	100%	0%

Methodology

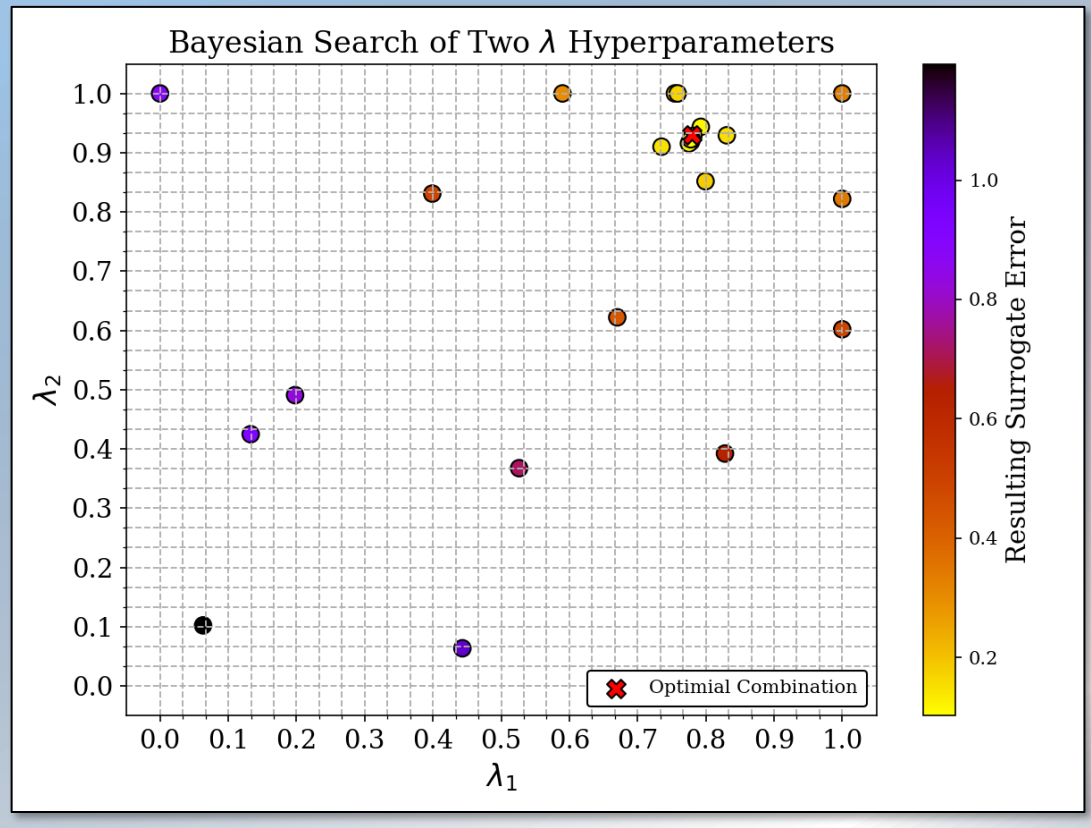
Ratio-Coupled Physics-Regularized Loss Function



Methodology

Explored Hyperparameter Searches

Hyperparameter Search	Implementation Details
Grid	Uses a range of λ hyperparameter values at predetermined intervals in λ ∈ [0, 1] with a fixed step size of 0.1 such that all λ ∈ {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} are tested.
Random	20 random λ values are selected from a uniform distribution with a precision of 0.01.
Optimized	The λ values are optimized during the training process by minimizing loss. Uses the convex loss function. Slowly refines to 'best' data coupling.
Optimized*	Uses the result of the optimized algorithm and uses this statically with the non-convex loss.
Bayes	There are 8 initial random λ values chosen and evaluated. Uses expected value to select next choices. A total of 30 iterations are completed. The final model is trained using the full number of epochs.



Results

Top Ranked Results and Search Technique Comparison

TABLE II. TOP 10 RESULTS SORTED BY THE CALCULATED RMSE OVER THE FULL FORECAST HORIZON.

Rank	λ Search	λ Value	Full Horizon RMSE	Single Step RMSE
1	Optimized*	0.95961833	0.045182	0.013879
2	Grid	0.9	0.045754	0.014290
3	Random	0.89	0.045981	0.014298
4	Random	0.95	0.046696	0.014519
5	Grid	1.0	0.047465	0.014367
6	Bayes	0.7319939	0.047566	0.013782
7	Random	0.79	0.048094	0.014380
8	Grid	0.7	0.048827	0.014427
9	Random	0.68	0.048832	0.014167
10	Grid	0.8	0.048988	0.014345

TABLE III. COMPARISON OF THE BEST RESULTS FOR EACH SEARCH TECHNIQUE.

Rank	λ Search	λ Value	Full Horizon RMSE	Single Step RMSE	Total Time Taken (≈ s)
1	Optimized*	0.95961833	0.045182	0.013879	3,109
2	Grid	0.9	0.045754	0.014290	18,701
3	Random	0.89	0.045981	0.014298	30,644
7	Bayes	0.7319939	0.047566	0.013782	5,344
34	Optimized	0.95961833	0.055039	0.014998	1,582

Results

Average Forecast Error and Example Forecast

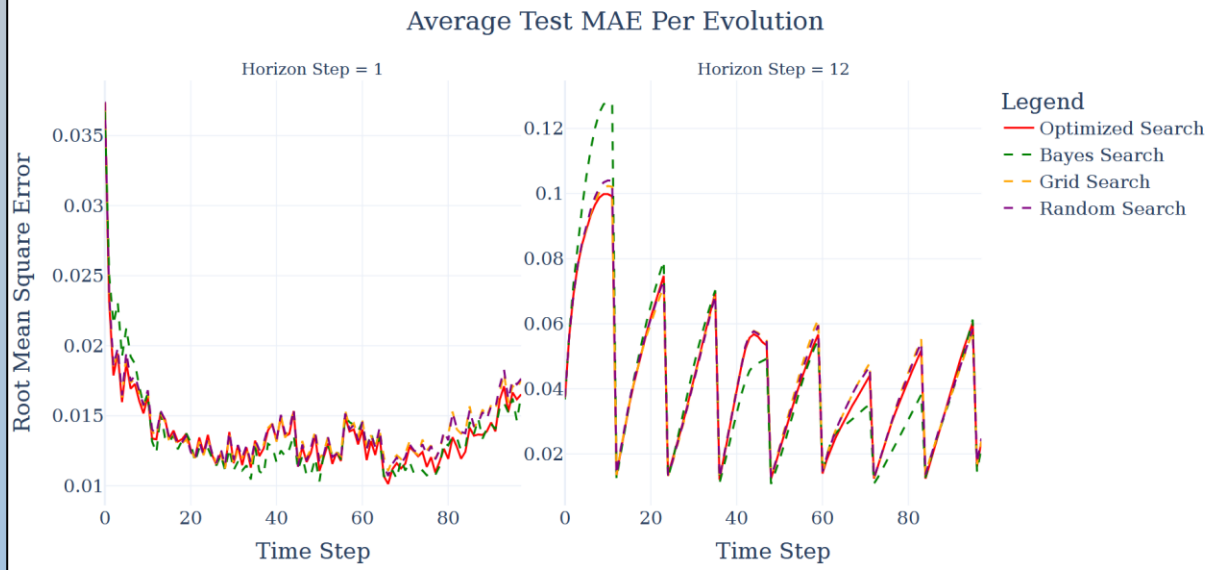


Figure 3. Evolution of average error over the evolution period. The single horizon error (left) and the full horizon error (right) highlight when the forecast problem is most difficult for each selected model.

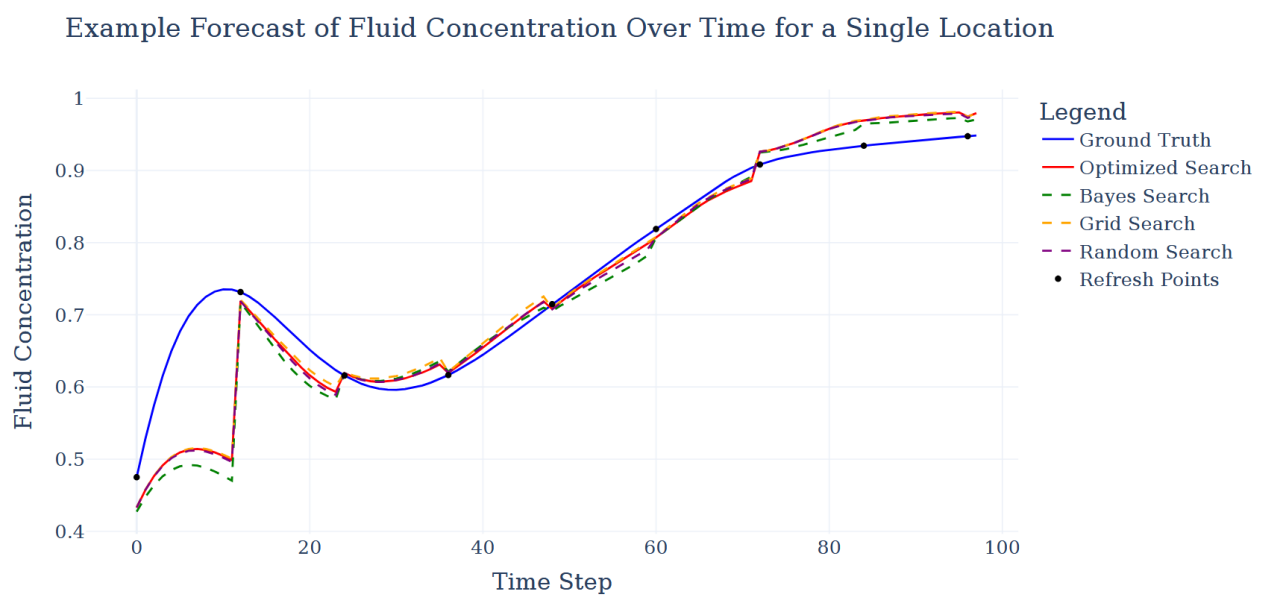


Figure 2. Example plot of a single observation node with a 12-step forecast horizon. Fresh initial values are seen every twelfth step.

Conclusion

Practical Implications of Results

- Comparisons between search techniques provide **guidance** on how to find the best λ
- Optimized search might not always give an optimal result
  - Training can be somewhat unstable
  - **But…** Resulting λ values are representative of Gaussian noise in the data
  - Suggested order to try when under time constraints:
    - Optimized - Static Optimized - Bounded Random
- Special mention to Bayes search
  - Small improvements could make this viable on small problems
- Using λ and a ratio-coupled loss is still better than when no data is combined

Conclusion

Future Investigations

1. Stabilize training of convex loss function
   - ML-related tasks (early stopping, special architectures, etc.)
   - Separate convex and non-convex loss in the same training loop

2. Validation on new datasets
   - More "*real-world*" data to compare against
   - 2D timeseries investigations (sparse predictions from available observations…)
   - Uncover *why* optimized search works well on some data but not others

3. Investigation into different noise patterns
   - Gaussian distribution gives good results, but would Uniform or Poisson?
   - Which is a good approximator of missing data samples?

Questions?