

1

## Introduction

Hybrid Neural Ordinary Differential Equations allow system dynamics to be defined jointly by both explicit mathematical formula and by neural networks.

Compared to traditional ordinary differential equation models, hybrid neural ordinary differential equations are better able to model unknown dynamics by the approximation properties of the included neural networks.

Conversely, hybrid neural networks can more easily conform to and incorporate existing knowledge of the modeled system than pure neural differential equations

Purely Mechanistic

$$\dot{\mathbf{X}} = \mathbf{n}(\mathbf{X}, t|\theta) \longleftrightarrow \dot{\mathbf{X}} = \mathbf{F}(g(\mathbf{X}, t|\mathbf{p}), \mathbf{n}(\mathbf{X}, t|\theta)) \longleftrightarrow \dot{\mathbf{X}} = g(\mathbf{X}, t|\mathbf{p})$$

Model Generality

Model Specificity

## Aims

1. Survey applications of hybrid neural ordinary differential to biological systems.
2. Investigate possible advantages of the hybrid neural differential equations for modeling biological systems through investigating test models
3. Assess the applicability of hybrid neural differential equations to unknown mechanism discovery.

3

## Background

Hybrid NODEs can be an extremely general class of models, we consider the subset which emerges as a fusion of mechanistic modeling and Neural differential equations,

Neural differential equations were most recently introduced by Chen et al. (2018). in a similar spirit to the incorporation of neural networks into chemical process models in the 1990s as in Psichogios et al. (1992).

Uses of neural networks generally rely on the universal approximator properties of neural networks to discover the underlying dynamics of a system, given a large enough network and enough data.

Hybrid NODEs have been applied with some success to a wide variety of modeling problems in biological systems. The table below reviews such applications in the literature.

### Hybrid NODE and NODE Implementations in Biological Systems Modeling

Study	Domain	Hidden Layers	Width	Activation
Bonnaire et al. 2021	Ecology	1	41	Sigmoid
Arroyo et al 2023	Ecology	1	20-80	Sigmoid, Hyperbolic Tangent
Wu et al. 2005	Ecology	1	10	Sigmoid
Rackauckas et al. 2021	Ecology	3	5	Gaussian Radial Basis
Psichogios et al. 1992	Process Modeling	-	-	Sigmoid
Van Can et al. 1998	Process Modeling	1	1-25	Hyperbolic Tangent
Oliveira et al. 2004	Process Modeling	1	8	Sigmoid
Quaghebeur et al. 2021	Process modeling	2	50,20	Hyperbolic Tangent
Quaghebeur et al. 2021	Process modeling	3	50,50,20	Hyperbolic Tangent
Quaghebeur et al. 2022	Process modeling	2	30,30	Hyperbolic Tangent
Nogueira et al. 2022	Process modeling	1	7	Hyperbolic Tangent
Kuwahara et al. 2023	Epidemiology	2	3,3	Gaussian Error Linear Unit
Dandekar et al. 2020	Epidemiology	2	10	ReLU
Acuesta et al. 2022	Epidemiology	1	10	ReLU
Song et al. 2022	Epidemiology	1	32	Hyperbolic Tangent

4

## Methods

### Model 1: Lotka Volterra

$$\frac{dR}{dt} = \alpha R - \beta RL$$

$$\frac{dL}{dt} = \gamma RL - \delta L$$

Note: Red highlighting indicates mechanistic components which are assumed unknown in the experiments. These components are set equal to 0 when missing.

### Model 2: Glycolysis

$$\frac{dS_1}{dt} = J_0 - \frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q}$$

$$\frac{dS_2}{dt} = 2 \frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q} - k_2 S_2 (N - S_5) - k_6 S_2 S_5$$

$$\frac{dS_3}{dt} = k_2 S_2 (N - S_5) - k_3 S_3 (A - S_6)$$

$$\frac{dS_4}{dt} = k_3 S_3 (A - S_6) - k_4 S_4 S_5 - \kappa (S_4 - S_7)$$

$$\frac{dS_5}{dt} = k_4 S_4 S_5 - k_5 S_5$$

$$\frac{dS_6}{dt} = -2 \frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q} + 2k_3 S_3 (A - S_6) - k_5 S_6$$

$$\frac{dS_7}{dt} = \psi \kappa (S_4 - S_7) - k_7 S_7$$

All work was done in python with Pytorch and Torchdiffeq libraries

We consider 2 distinct test models of meaningfully different complexity. The 2 species classic Lotka-Volterra model of population dynamics forms our simple case, and was previously been used in Rackauckas et al (2020).

A model of glycolysis in yeast forms our more complex case. The glycolysis model is from Wolf and Heinrich (2000) originally, but was considered in a similar setting in Quaghebeur et al. (2021).

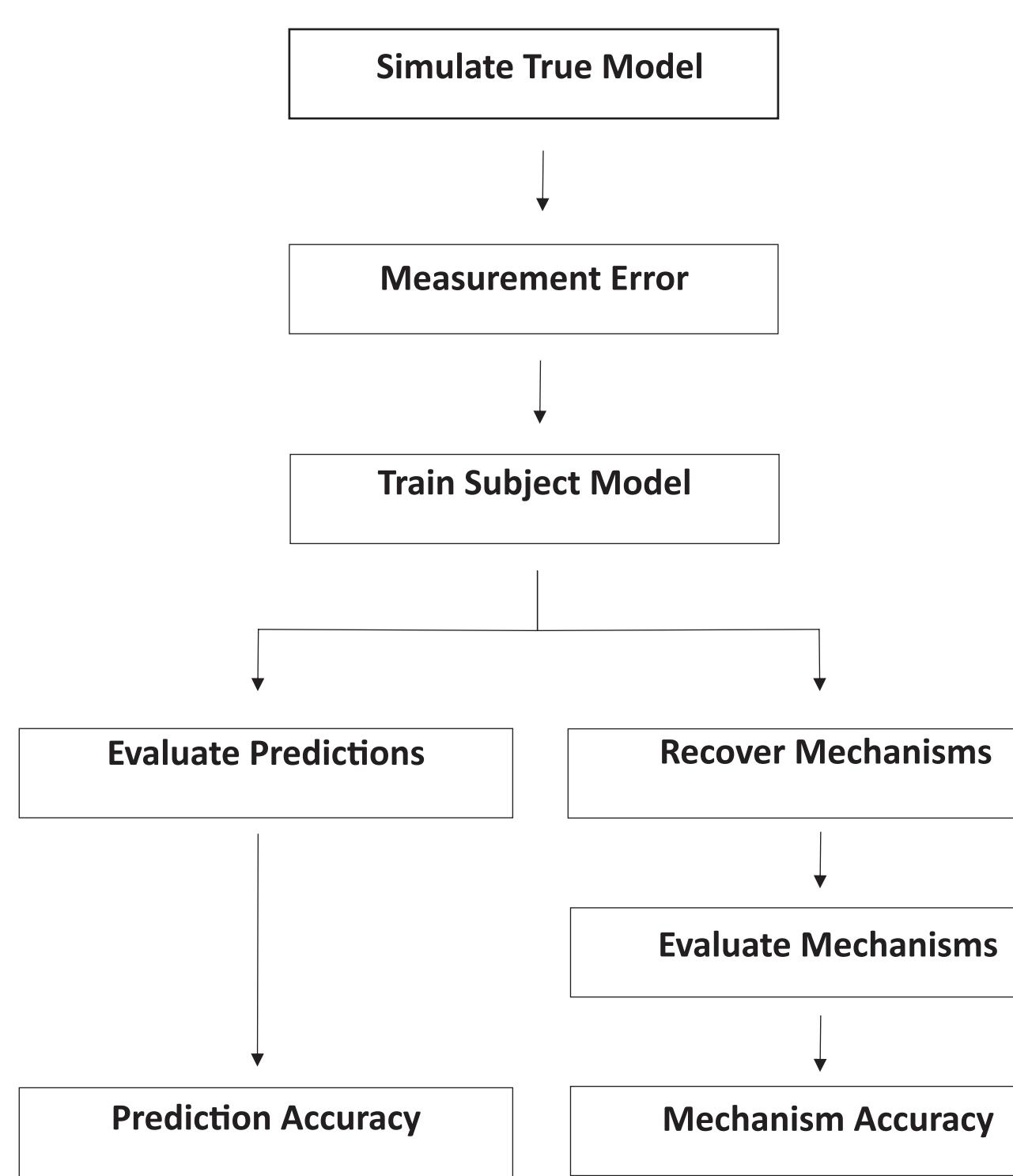
### Pure Neural ODE Model

$$\frac{dS}{dt} = N(S_{missing}|\theta)$$

### Hybrid Neural ODE Model

$$\frac{dS}{dt} = \frac{dS_{missing}}{dt} + N(S_{missing}|\theta)$$

### Computational Procedure



We consider both a pure and hybrid neural differential equations for both systems. The hybrid neural differential equation is defined additively.

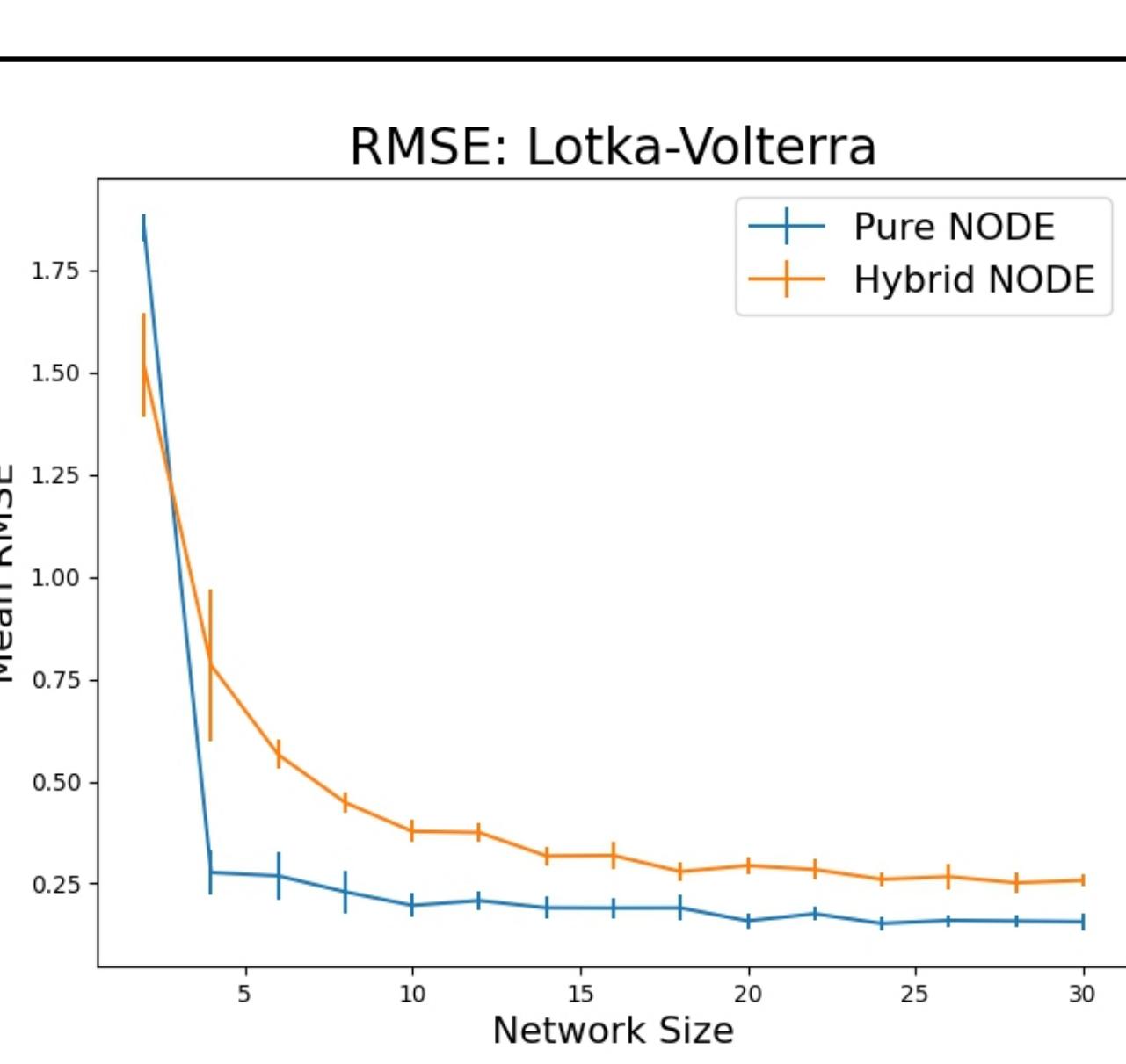
The computational procedure is demonstrated to the left. The Lotka-Volterra and Glycolysis models are trained using 50 and 150 data points respectively. Accuracy is evaluated using 25 and 75 points in the testing set, respectively.

Mechanism recovery is accomplished by symbolic regression on the neural network components.

Both accuracies are evaluated by RMSE against prediction and true mechanisms.

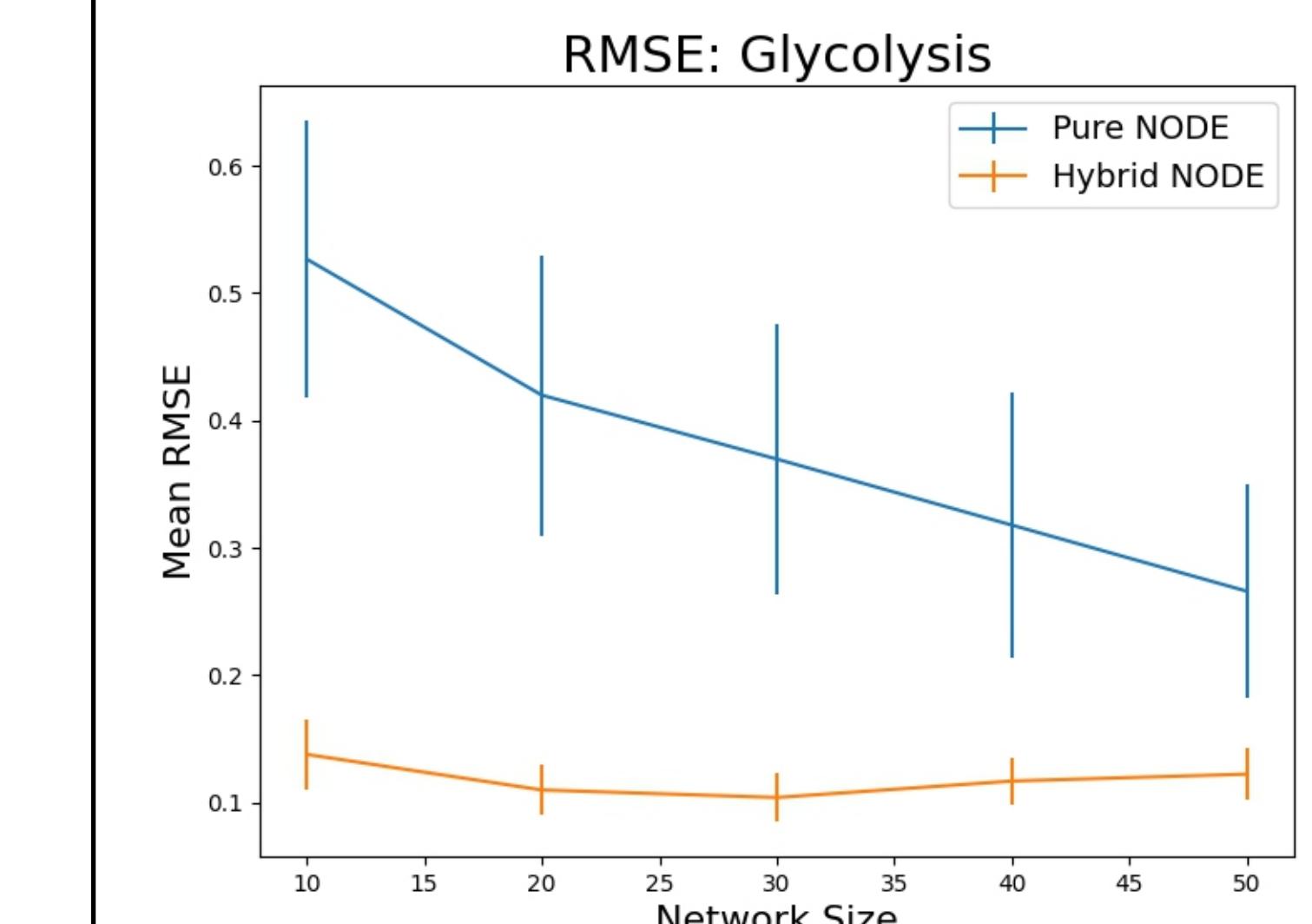
5

## Preliminary Results



All models use a single layer feedforward neural network with Tanh activation functions. Thirty simulations were taken for each data point in each model.

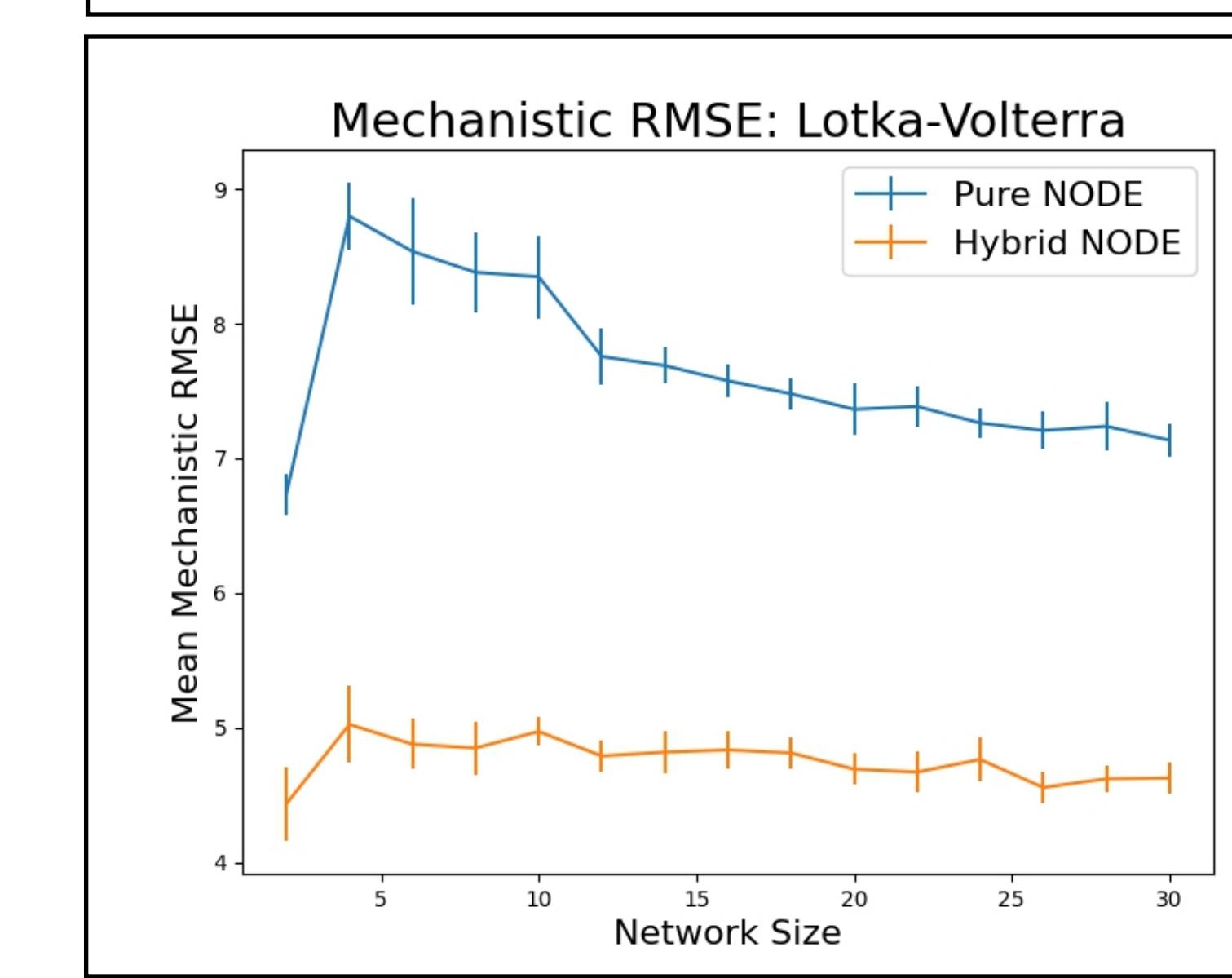
The plot to the right compares the performance of the pure and hybrid neural differential equation models on the Lotka-Volterra model. We see that except for the case of 2 neurons, the pure neural differential equation outperforms the hybrid neural differential equation model.



The image to the right compares the performance of the pure and hybrid neural differential equations in predicting the glycolysis model. We see that the hybrid neural differential equation model outperforms the pure neural differential equation model.

Differences between the two predictive cases might be understood in terms of the complexity of the considered models. Overfitting the Lotka-Volterra system with a neural differential equation may be easy, but overfitting the glycolysis model in such a manner is much harder.

The image to the right compares the recoverability of the mechanistic component using symbolic regression for both pure and hybrid neural differential equations. Predictably, the hybrid case performs better, as the network is better specialized to the unknown mechanism.



## Future Work

Future work in the current project is expected to follow three trends:

1. Expand scope of examples, especially to include from epidemiology or immunology.
2. Expand scope of computer experiments, especially regarding the impact of data size

Future projects may consider the following directions:

1. Adapting the training algorithm to be online rather than batched.
2. Applications of mechanism recovery to large sparse differential equation systems.

7

## Selected References

- Wolf, J., & Heinrich, R. (2000). Effect of cellular interaction on glycolytic oscillations in yeast: a theoretical investigation. *Biochemical Journal*, 345(2), 321-334.
- Quaghebeur, W., Nopens, I., & De Baets, B. (2021). Incorporating unmodeled dynamics into first-principles models through machine learning. *IEEE Access*, 9, 22014-22022.
- Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supek, R., ... & Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*.
- Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.
- Psichogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AICHE Journal*, 38(10), 1499-1511.

8

## Acknowledgements



Research reported in this publication was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number R01GM152736