Some Preliminary Transfer Learning Results for Dynamical Systems

Joshua E. Hammond,
Brian A. Korgel, Michael Baldea
The University of Texas at Austin

Tyler A. Soderstrom
ExxonMobil Technology and Engineering

Objective: Use limited data from a target system and the model of a similar source system trained on abundant source data to learn an accurate neural-network representation of the target system.

Hypothesis: If $\mathbf{f}^\mathbf{S} \leftrightarrow \mathbf{f}^\mathbf{T}$ and $\mathcal{D}^\mathbf{S} \leftrightarrow \mathcal{D}^\mathbf{T}$:

1. The parameter space of $\mathbf{\Phi}^\mathbf{S}$ is close to that of $\mathbf{\Phi}^\mathbf{T}$.
2. A path exists from the source model parameters to the target model parameters.

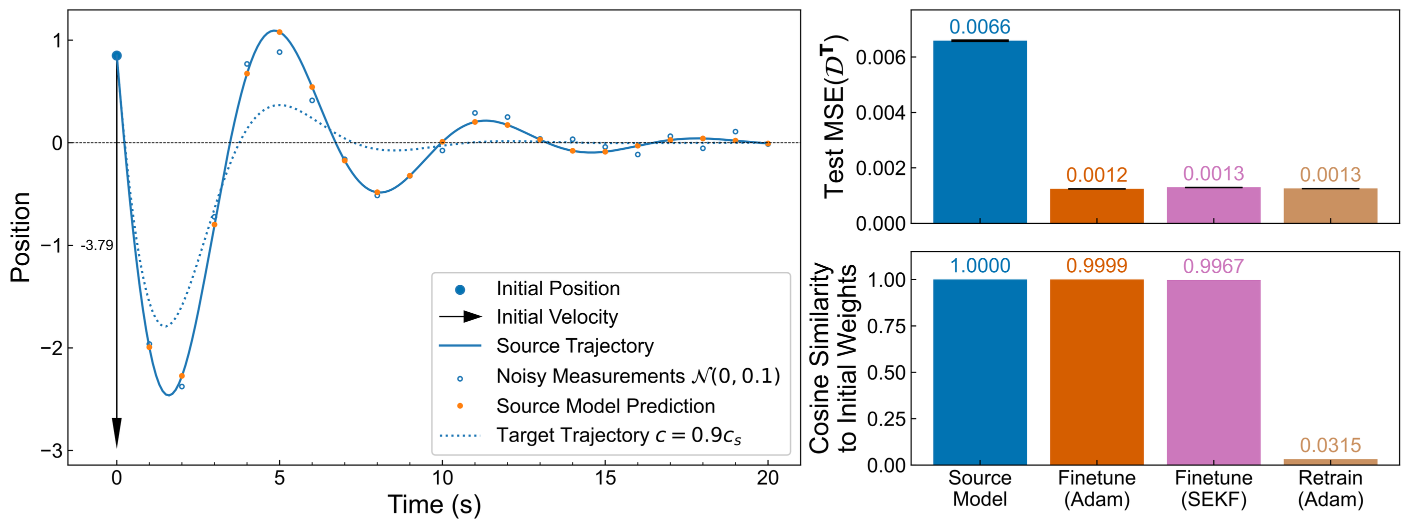
Spring-Mass-Damper System

$m\ddot{x} + c\dot{x} + kx = u$

Given: $x_0, \dot{x}_0$ **Predict:** $x_1, \ldots, x_{20}$

Transfer: $m, c, k, u \pm 10\%$

$\dim(\mathcal{D}^\mathbf{S}) = 100,000$, $\dim(\mathcal{D}^\mathbf{T}) = 1,000$



CSTR System[1]

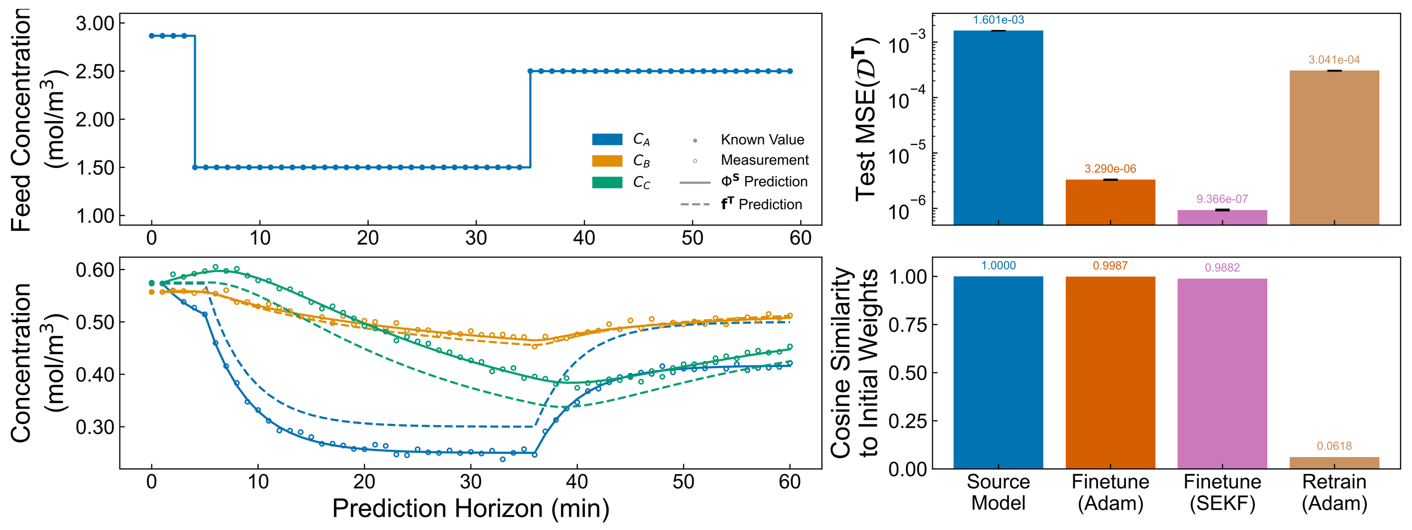
$A \rightarrow B \rightleftharpoons C$

Given: $C_{A,0}, C_{B,0}, C_{C,0}, C_{Af,0\ldots59}$

Predict: $C_{A,1\ldots60}, C_{B,1\ldots60}, C_{C,1\ldots60}$

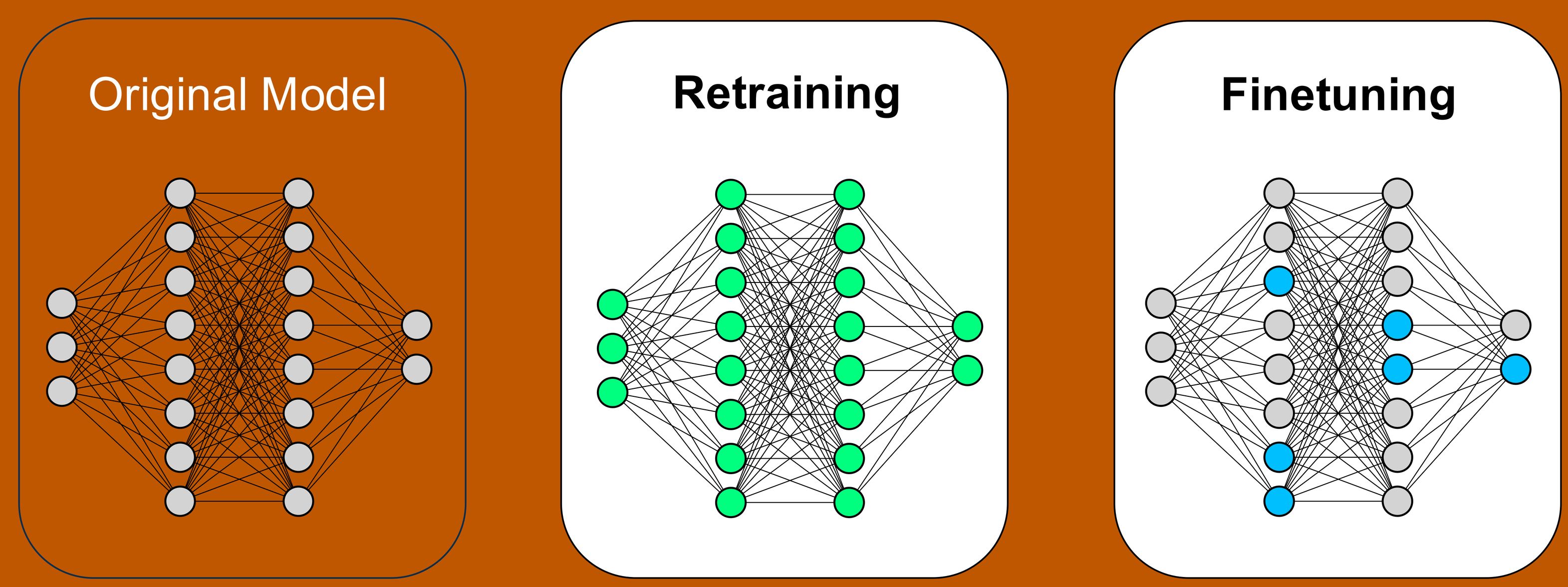
Transfer: $\Delta V_{rxn} - 20\%$

$\dim(\mathcal{D}^\mathbf{S}) = 1$ Year, $\dim(\mathcal{D}^\mathbf{T}) = 1$ Day



[1] Kumar, P., & Rawlings, J.B. (2023). Structured nonlinear process modeling using neural networks and application to economic optimization.

Small changes to an existing neural-network model can accurately represent a similar system using **less data** than initial training.



Take a picture or visit **bit.ly/TWCCC-Transfer-Learning:**

1. Full results and presentation
2. Previous work updating data-driven models of slowly-evolving dynamical systems

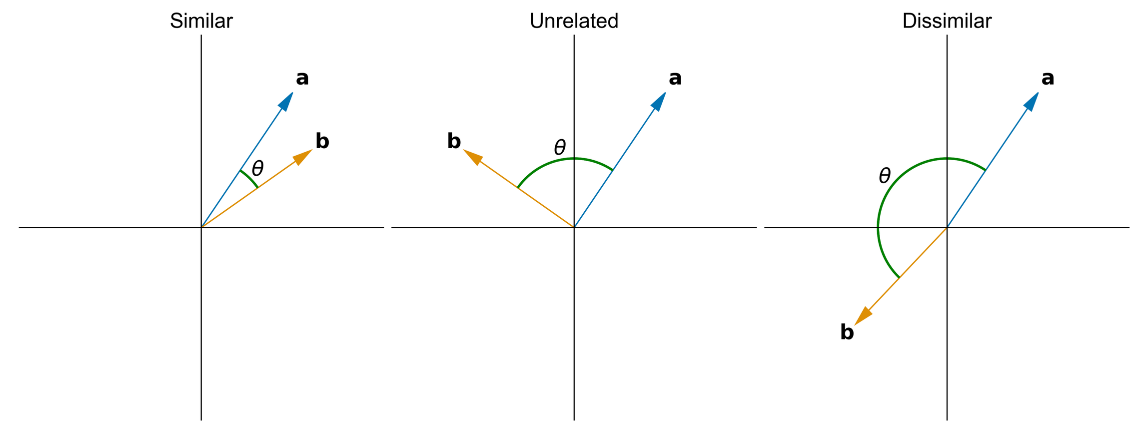
Machine Learning Models of Dynamical Systems

$\frac{d\mathbf{x}}{dt} = \mathbf{f}^\mathbf{S}(\mathbf{x}, \mathbf{u}, \mathbf{p}^\mathbf{S})$ $\quad \mathbf{x}_{\mathbf{k+1}} = \mathbf{\Phi}^\mathbf{S}(\mathbf{x}_\mathbf{k}, \mathbf{u}_\mathbf{k}, \boldsymbol{\pi}^\mathbf{S})$

$\frac{d\mathbf{x}}{dt} = \mathbf{f}^\mathbf{T}(\mathbf{x}, \mathbf{u}, \mathbf{p}^\mathbf{T})$ $\quad \mathbf{x}_{\mathbf{k+1}} = \mathbf{\Phi}^\mathbf{T}(\mathbf{x}_\mathbf{k}, \mathbf{u}_\mathbf{k}, \boldsymbol{\pi}^\mathbf{T})$

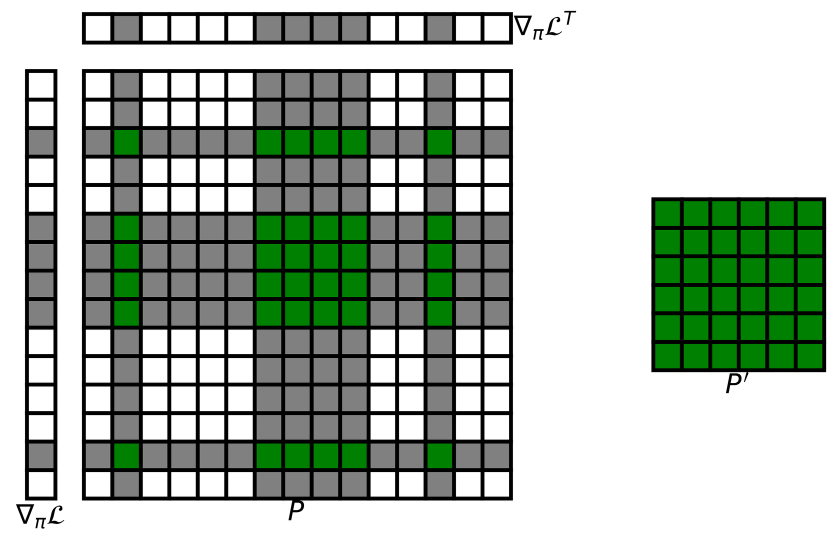
Defining Similarity

We use *cosine similarity* to quantify the similarity of individual NN outputs and the weights of two NNs



$\text{cosine\_similarity}(\mathbf{a}, \mathbf{b}) \triangleq \cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \in [-1, 1]$

The Subset Extended Kalman Filter (SEKF)[2]



The SEKF reduces the computational cost of EKF operations by only updating a subset of the NN model parameters $\boldsymbol{\pi}$ at each time step. We use the gradient of the loss function with respect to the parameters $\nabla_{\boldsymbol{\pi}}\mathcal{L}$ to select which subset of parameters to update.

The gradients of the selected parameters along with the corresponding subset of the covariance matrix $\mathbf{P}$ are used in the EKF update equations.

[2] Hammond et. al "Staying Alive: Online Neural Network Maintenance and Systemic Drift" arXiv:2503.17681

Nomenclature

$\mathbf{S}$	Source	$\mathbf{T}$	Target
$\mathbf{f}$	Governing Equations	$\mathcal{D}$	Dataset
$\mathbf{\Phi}$	Neural Network	$\mathbf{x}$	States
$\mathbf{u}$	Inputs	$\mathbf{p}$	System Parameters
$\boldsymbol{\pi}$	NN parameters	$\nabla_{\boldsymbol{\pi}}\mathcal{L}$	Gradient of Loss Function w.r.t. $\boldsymbol{\pi}$

The University of Texas at Austin
McKetta Department of Chemical Engineering
Cockrell School of Engineering