Machine Learning for Reparameterization of Multi-scale Closures

Hilary Egan
National Renewable Energy Laboratory (NREL)
Golden, CO 80401
hilary.egan@nrel.gov

Meagan Crowley
National Renewable Energy Laboratory (NREL)
Golden, CO 80401
meagan.crowley@nrel.gov

Hariswaran Sitaraman
National Renewable Energy Laboratory (NREL)
Golden, CO 80401
hariswaran.sitaraman@nrel.gov

Peter Ciesielski
National Renewable Energy Laboratory (NREL)
Golden, CO 80401
peter.ciesielski@nrel.gov

Abstract

Scientific machine learning (ML) is becoming increasingly useful in learning closure models for multi-scale physics problems; however, many ML approaches require a vast array of training data and can struggle with generalization and interpretability. Here, rather than learning an entire closure operator, we adopt an existing reduced-dimension model of the microphysics and learn an optimal re-parameterization of the solver. We demonstrate two approaches for the training the reduced dimension closure model 1) an a priori method that optimizes the closure parameterization and the neural network parameters separately and 2) an a posteriori method that simultaneously optimizes both. We find that while each method is capable of achieving a target loss, the a posteriori method is capable of achieving the target loss with a smaller network, and smaller training data sizes.

1 Introduction

Many physics problems occur over a vast range of length scales, time scales, and/or dominant physical processes, from climate science to materials physics. Simulating these complex phenomena in detail across all scales simultaneously is typically computationally intractable; however, the interaction across scales drives the physics at the heart of many science and engineering problems. This difficulty has led to the development of many multi-scale numerical methods for representing and interrogating systems across the scale hierarchy. One common approach is to construct constitutive relationships representing the effect of microscale processes at macroscale levels. Scientific ML has become an increasingly important tool in the development of such closure models [e.g. Pathak et al., 2022, E et al., 2023, Stachenfeld et al., 2022]; however, many black-box ML methods require large amounts

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

of expensive simulations to acquire training data, struggle to generalize, and lack interpretability [Brunton et al., 2020, Mishin, 2021].

Rather than learning the full time derivative operator in a conservation law based differential equation system, in this paper we use a simple reduced-dimension (0D) model as the closure to calculate the time derivative and instead learn a reparameterization that improves the predictive capability of the reduced dimension model. Using the simulation of biofuel pyrolysis as a motivating example, in this paper we demonstrate the implementation of both an a priori (I) and an a posteriori (II) learning scheme where a neural network (NN) is optimized to predict parameters supplied to a reduced dimension differentiable ODE solver. This parameterization can then be directly implemented in the macroscale model without needing to modify the large scale solver.

2 Methods

2.1 Particle Scale Models

The high fidelity microscale model is a finite element method (FEM) based model of time-dependent biomass pyrolysis of a single particle using COMSOL Multiphysics 5.6. We use a 2D axisymmetric model with anisotropic heat and mass transport, simultaneous multi-species chemical reactions. The reaction scheme, heat capacities, molecular weights, and baseline Arrhenius rate constants from the Miller-Bellan scheme [Xiong et al., 2013]; we list the relevant reaction equations and baseline rate constants in Appendix Tables 2 and 3 respectively. The full model is described in Pecha et al. [2021].

To create the training and test datasets we use Latin Hypercube sampling to select 250 data trajectories across oven temperature, particle Feret Length (FL), and particle aspect ratio; 20% (50) of the trajectories were reserved as test data leaving 200 as the full training set. We also linearly sampled an additional set of trajectories to form a validation set where we could verify trends in error with input parameter.

2.2 Reactor Scale Models

The macroscale model used in this work is an Euler-Euler multiphase fluid model, from within open-source computational fluid dynamics library, OpenFOAM; the multifluid approach is routinely used for large scale gas-solid multiphase systems Xiong et al. [2013] such as in a biomass pyrolysis reactor. The governing conservation partial differential equations for mass, momentum and energy for each phase are solved along with drag, heat transfer, and chemical production interaction terms between phases, similar to our earlier models on bioreactors Sitaraman et al. [2023].

2.3 Optimization of Closure Parameterization

Let the full high-fidelity partial differential equation (PDE) be denoted

$$F(u;\lambda) = 0 \tag{1}$$

where F represents the PDE operator, $u(x,t)$ is the solution, and λ represents physical parameters of the system or boundary conditions. Then, let A be some reduction operator such that

$$A(F(\bar{v};\lambda)) = \bar{F}(\bar{v};\lambda) = 0 \tag{2}$$

where the desired solution $\bar{u} = A(u)$ is not necessarily equivalent to $\bar{v}$ (and in most cases is not) as the model reduction operator A and PDE operator F do not commute. Furthermore, let μ be explicitly defined as a set of empirical parameters that arise from applying the reduction operator (e.g. a spatially averaged heat transfer coefficient), such that $\bar{F}(\bar{v};\lambda,\mu) = 0$.

Following the formalism of Sanderse et al. [2024], the goal of closure modeling is to find G_θ with parameters θ such that $G_\theta(\bar{v}';\lambda) = 0$ with $\bar{v}' \approx \bar{u}$. One natural choice for the form of G_θ is $G_\theta = \bar{F} + C$; this then defines a commutator error $C(u,\bar{u},\mu) \approx m_\theta(\bar{u},\mu)$. On the other hand, if the exact form of F is not known and one only specifies that the reduced model should be of the evolution form, then the entire solution $G_\theta(\bar{v},\mu) = \frac{\partial \bar{v}}{\partial t} + m_\theta(\bar{v},\mu)$ can be learned without specifying that m_θ must take the form of the commutator. Here we define

$$G_\theta := \bar{F}(\bar{v}';\lambda,\mu_\theta) \tag{3}$$

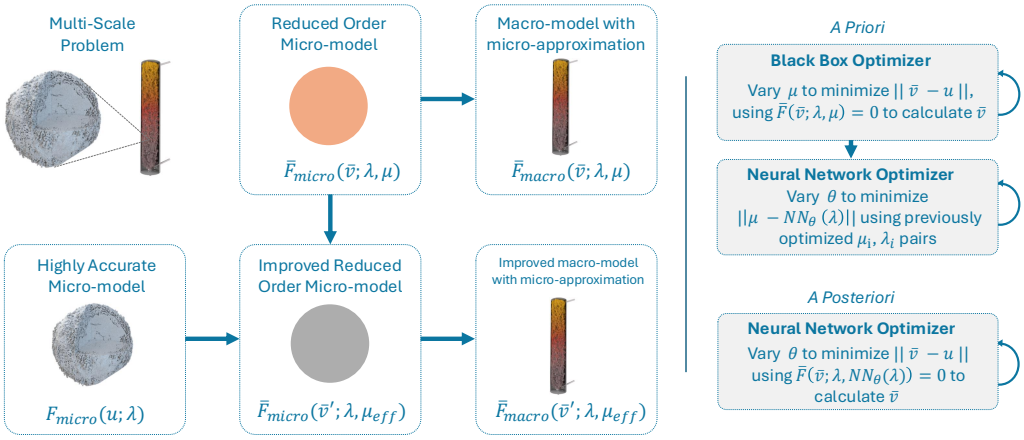

Figure 1: Workflow showing training process for a priori and a posteriori models as related to the micro and macro scale models.

where $\mu_\theta := f_\theta(\lambda)$ which we will approximate by a neural network $NN_\theta(\lambda)$. This approximation is inherently limiting in multiple ways: 1) defining G_θ as a reparameterization of $\bar{F}$ and 2) allowing μ_θ to only depend on λ, not u; however, this trade-off allows us to minimize the volume of high-fidelity microscale simulations needed to train a reliable functions, requires no modification of the macroscale solver to implement, and does not introduce any operator dependent stability concerns.

A guiding intuition in our approach is that learning an effective closure is very similar to learning an effective surrogate model, yet existing approximations (e.g. a 0D representation of a 3D process) already embed much of the physics that is of interest. Therefore, in many engineering cases where the ultimate goal is improved predictability of some Quantity of Interest (QoI) and macroscale models already introduce a non-negligble error, learning adjustments to an existing approximation is faster than learning the entire closure from scratch and still improves upon the state-of-the-art.

In this paper we contrast two approaches to learning $\mu_\theta = f_\theta(\lambda)$: an a priori (I) and a posteriori (II) method. In both methods $\bar{u_i}$ are sampled from the high-fidelity microscale model such that $F(u_i;\lambda_i)=0$, while $\bar{v}_i$ are calculated using an approximate solver $\bar{F}$ such that $\bar{F}(v_i;\lambda_i,\mu_i)=0$. Parameters $\lambda_i \in \Lambda$ and Λ is a predefined parameter region of interest. Both methods are illustrated graphically in Figure 1.

1. **A Priori**: In the a priori method, for each λ_i a black box optimizer is used to minimize $||\bar{u}_i - \bar{v}_i||$ through varying μ_i where $\bar{F}(\bar{v}_i;\lambda_i,\mu_i)=0$. The parameters of the neural network NN_θ are then optimized on $||\mu_i - \mu_i'||$ where $\mu_i' = NN_\theta(\lambda_i)$.
2. **A Posteriori** In the a posteriori method, a differentiable $\bar{F}$ solver is used in the neural network optimization. Thus, $||\bar{u}_i - \bar{v}_i||$ can be optimized for directly using $\bar{F}(\bar{v}_i;\lambda_i,NN_\theta(\lambda_i))=0$.

While ensuring that $\bar{F}$ is differentiable is a high barrier for many applications, the increasing popularity of auto-differentiation has led to a robust industry of developing fast, fully differentiable solvers. In particular, when considering multi-scale problems where continuum approximations average out fine-scale variation and spatial gradients are smaller than time gradients for the property of interest, a given grid cell the reduced PDE $\bar{F}$ can be approximated by a non-spatially dependent ordinary differential equation (ODE) $\bar{F}'$ of which many fast differentiable options exist.

3 Results

Table 1 summarizes the architectures and corresponding losses for both schemes. We find that the best a priori and a posteriori scheme achieve similar losses in the final solution, with the a posteriori

model slightly outperforming the a priori model in the test dataset. We also note that there is a non-negligible lower limit in model error due to the inherent approximation in constructing G' as a non-spatially varying approximator. By comparing multiple black box optimizers across the training & test datasets we estimate that error to be roughly 1E-3; therefore, our best models are approaching the lower loss limit.

Figure 2 shows several example roll-outs of the various models for our test dataset. In all cases both the a priori and a posteriori models improve consistency with the full 3D model compared to a baseline computed by running the same 0D model without modifying the chemistry scheme. Note that in each rollout the particle parameters (listed in Appendix Table 4) are fed to the neural networks a single time to compute the effective activation energies for a 0D model, rather than computing a derivative at each time step. The effective activation energies are then fed as fixed parameters to the same differentiable IVP solver used in training the a posteriori models to integrate the solution over the time period of interest.

Model	**Hidden Layers**	**Yield Solution MSE Train**	**Yield Solution MSE Test**	E_A **MSE Train**	E_A **MSE Test**
Neural ODE	(256,256)	1.11E-03	1.21E-03	-	-
Offline A	(256,256)	1.06E-03	2.19E-03	7.32E-02	1.44E+00

Table 1: Comparison of errors across model for the a priori and the a posteriori model. Note that as the a posteriori model is not trained based on existing E_A targets we do not report E_A loss.

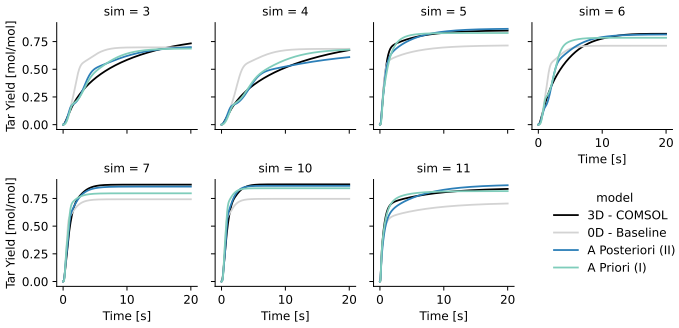

Figure 2: Example rollouts for test data set points showing the desired quantity of interest (Tar Yield) over time for the full 3D COMSOL model (*black*), the baseline approximator with no parameter variation (*grey*), the a priori model (*green*), and the a posteriori model (*blue*).

A key concern for using machine learning methods to train surrogate models (or parameterizations thereof) is the volume of training data required, particularly for relatively expensive high-fidelity models. In Figure 3a we compare the efficacy of the trained models on number of training data points using data randomly selected from the initial training data set to form the smaller train subsets. We find that the a posteriori method requires far fewer data points to reach the target test accuracy than the a priori method and thus generalizes far better for small dataset sizes.

From a domain science perspective, the goal of this study is to improve the predictive capability of the reactor scale yields based on particle scale properties; therefore in Figure 3b and c we investigate the dependence of yields on particle Feret length at both the micro and macro scale. As tar yield depends strongly on temperature through Arrhenius reaction rates, neglecting the influence of temperature gradients across the particle leads to a very minimal Feret length trend in the baseline model (larger particle sizes will have correspondingly larger temperature gradients across the particle). Both approaches to correcting the reduced dimension model do show a stronger trend in yield with Feret length than the baseline; however, the a posteriori model generally shows a closer match to the full 3D results. Furthermore, the a posteriori model appears to be much smoother, an important factor for incorporating the model into optimization schemes at the reactor scale.

We hypothesize that both the difference in surrogate smoothness and the dependence of model accuracy on the training data size is due to the properties of the loss landscape in a priori black box

optimization. The moderate degeneracy of the output space parameters combined with a relatively flat loss landscape create a situation where training data points close in particle parameter space may not have values close in effective activation energy space from the black box optimizer, while co-optimizing incentivizes a smoother relationship. This interpretation is supported by Appendix Figures 4; however, further study is necessary to fully support this conclusion.

As the a posteriori model has generally performed better in comparisons thus far, we exclusively show the impact of implementing this surrogate parameterization at the macroscale. As expected, the reactor model with surrogate parameterization shows a much stronger dependence on particle Feret length than the baseline fixed parameter model. While the trend in decreasing yield with Feret length are similar in the particle scale and reactor scale model, the full two-fluid reactor model also includes impacts of particle residence time and spatial temperature dependence of the reactor and is thus worthy of a separate comparison.

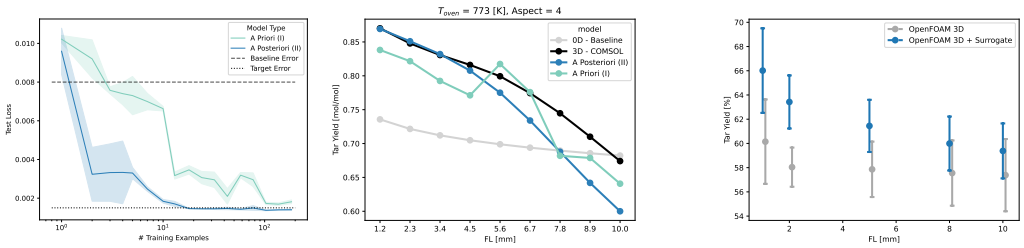

Figure 3: *Left:* Mean test accuracy versus number of training data points for the a priori (green) and a posteriori (blue) models. Shaded regions show 1 standard deviation. The dashed line shows the error of the baseline model with no parameter variation while the dotted line shows an estimate of the target data error (constrained by the capability of the non-spatially varying approximator). *Center:* Final tar yield versus particle Feret Length (FL) at the particle scale for the full 3D COMSOL model (*black*), the baseline approximator with no parameter variation (*grey*), the a priori model (*green*), and the a posteriori model (*blue*). *Right:* Average tar yield versus particle Feret Length at the reactor scale for the baseline OpenFOAM model (*grey*) and the OpenFOAM model with the a posteriori model for activation energies (*green*). Error bars show standard deviation from time averaged fluxes at the outlet after the reactor simulations reach steady state.

4 Conclusion

In this paper we have investigated two approaches for the training of a simple machine learning model to predict the optimal parameterization of a reduced dimension model: 1) an a priori method where we initially use a black box optimizer to find the optimal parameterization to match each training data trajectory and then train a neural network on the input parameter/pre-optimized parameter pairs and 2) an a posteriori method where we incorporate a differentiable initial value problem solver directly into the training loss of the neural network and optimize in a single stage. We find that while each method is capable of achieving a target loss, the a posteriori method is capable of achieving the target loss with a smaller network, smaller training data sizes, and produces a smoother output parameter landscape, though further investigation is needed to assess to what degree these conclusions depend on the loss landscape of the parameterization. Finally, we demonstrate the utility of this method for improving the predictive capability of macroscale models for multi-scale problems, without requiring any modification of existing solvers.

References

S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52(Volume 52, 2020):477–508, Jan. 2020. ISSN 0066-4189, 1545-4479. doi: 10.1146/annurev-fluid-010719-060214. URL `https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-010719-060214`. Publisher: Annual Reviews.

W. E, H. Lei, P. Xie, and L. Zhang. Machine learning-assisted multi-scale modeling. *Journal of Mathematical Physics*, 64(7):071101, July 2023. ISSN 0022-2488. doi: 10.1063/5.0149861. URL https://doi.org/10.1063/5.0149861.

Y. Mishin. Machine-learning interatomic potentials for materials science. *Acta Materialia*, 214: 116980, Aug. 2021. ISSN 1359-6454. doi: 10.1016/j.actamat.2021.116980. URL https://www.sciencedirect.com/science/article/pii/S1359645421003608.

J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators, Feb. 2022. URL http://arxiv.org/abs/2202.11214. arXiv:2202.11214 [physics].

M. B. Pecha, N. E. Thornburg, C. A. Peterson, M. F. Crowley, X. Gao, L. Lu, G. Wiggins, R. C. Brown, and P. N. Ciesielski. Impacts of Anisotropic Porosity on Heat Transfer and Off-Gassing during Biomass Pyrolysis. *Energy & Fuels*, 35(24):20131–20141, Dec. 2021. ISSN 0887-0624. doi: 10.1021/acs.energyfuels.1c02679. URL https://doi.org/10.1021/acs.energyfuels.1c02679. Publisher: American Chemical Society.

B. Sanderse, P. Stinis, R. Maulik, and S. E. Ahmed. Scientific machine learning for closure models in multiscale problems: a review, Mar. 2024. URL http://arxiv.org/abs/2403.02913. arXiv:2403.02913 [cs, math].

H. Sitaraman, J. Lischeske, Y. Lu, and J. Stickel. A reacting multiphase computational flow model for 2, 3-butanediol synthesis in industrial-scale bioreactors. *Chemical Engineering Research and Design*, 197:38–52, 2023.

K. Stachenfeld, D. B. Fielding, D. Kochkov, M. Cranmer, T. Pfaff, J. Godwin, C. Cui, S. Ho, P. Battaglia, and A. Sanchez-Gonzalez. Learned Coarse Models for Efficient Turbulence Simulation, Apr. 2022. URL http://arxiv.org/abs/2112.15275. arXiv:2112.15275 [physics].

Q. Xiong, S.-C. Kong, and A. Passalacqua. Development of a generalized numerical framework for simulating biomass fast pyrolysis in fluidized-bed reactors. *Chemical Engineering Science*, 99: 305–313, 2013.

A Supplementary Information

Component	Reaction	Y	A [s^{-1}]	E [MJ·$kmol^{-1}$]
Cellulose	k_{1c}	–	2.80E+19	242.4
	k_{2c}	–	3.28E+14	196.5
	k_{3c}	$Y_c = 0.35$	1.30E+10	150.5
Hemicellulose	k_{1h}	–	2.10E+16	186.7
	k_{2h}	–	8.75E+15	202.4
	k_{3h}	$Y_h = 0.60$	2.60E+11	145.7
Lignin	k_{1l}	–	9.60E+08	107.6
	k_{2l}	–	1.50E+09	143.8
	k_{3l}	$Y_l = 0.75$	7.70E+06	111.4
Tar	k_4	–	4.25E+06	108.0

Table 2: Kinetic parameters of CSW pyrolysis. Our optimization varies the activation energy vector (last column).

Material No.	Reaction rate equation
1	$\frac{dm_1}{dt} = -k_{1c}m_1$
2	$\frac{dm_2}{dt} = -k_{1h}m_2$
3	$\frac{dm_3}{dt} = -k_{1l}m_3$
4	$\frac{dm_4}{dt} = k_{1c}m_1 - (k_{2c} + k_{3c})m_4$
5	$\frac{dm_5}{dt} = k_{1h}m_2 - (k_{2h} + k_{3h})m_5$
6	$\frac{dm_6}{dt} = k_{1l}m_3 - (k_{2l} + k_{3l})m_6$
7	$\frac{dm_7}{dt} = k_{2c}m_4 + k_{2h}m_5 + k_{2l}m_6 - k_4m_{10}$
8	$\frac{dm_8}{dt} = (1 - Y_c)k_{3c}m_4 + (1 - Y_h)k_{3h}m_5 + (1 - Y_l)k_{3l}m_6 + k_4m_{10}$
9	$\frac{dm_9}{dt} = Y_ck_{3c}m_4 + Y_hk_{3h}m_5 + Y_lk_{3l}m_6$

Table 3: Chemical reaction rate equations of all the components.

Sim	T_{oven} [K]	FL [mm]	Aspect
0	8.02E+02	8.08E+00	3.38E+00
1	7.79E+02	9.85E+00	3.80E+00
2	7.42E+02	1.06E+00	5.18E+00
3	7.99E+02	7.96E+00	6.50E+00
4	8.00E+02	1.26E+00	2.32E+00
5	8.10E+02	2.78E+00	6.47E+00
6	7.26E+02	4.06E-01	5.57E+00

Table 4: Simulation parameters for the test simulations shown in Figure 2

Figure 4: Difference between the baseline activation energy value and the optimized/predicted values for the black-box optimized values (*top subfigure*), the predictions from the a priori optimized NN (*middle subfigure*), and the predictions from the a posteriori optimized NN (*bottom subfigure*). Each column shows a different activation energy from the 10 chemical reaction pathways (see Table 2). Values are calculated using three separate 2 dimensional sweeps with fixed oven temperature (*top row*), fixed aspect ratio (*middle row*), and fixed Feret Length (*bottom row*). Note the relative discrepancy in smoothness of the black box optimization and a priori model (trained on similar black box data) in comparison to the a posteriori model.