Towards long rollout of neural operators with local attention and flow matching-inspired correction: An example in frontal polymerization PDEs

Pengfei Cai
Massachusetts Institute of Technology
pengfeic@mit.edu

Sulin Liu
Massachusetts Institute of Technology
sulinliu@mit.edu

Qibang Liu
University of Illinois at Urbana-Champaign
qibang@illinois.edu

Philippe H. Geubelle
University of Illinois at Urbana-Champaign
geubelle@illinois.edu

Rafael Gomez-Bombarelli
Massachusetts Institute of Technology
rafagb@mit.edu

Abstract

Recent advances in frontal ring-opening metathesis polymerization (FROMP) offer a sustainable energy-efficient alternative for the rapid curing of thermoset polymers. Predicting the dynamics and spontaneous patterning of FROMP require numerically solving reaction-diffusion PDEs but it is computationally expensive. Neural operators serve as surrogate PDE solvers but face challenges in long temporal rollouts. Herein, we augment Fourier neural operators with large kernel attention to better learn local instabilities in FROMP. Inspired by conditional functional flow matching, we proposed a correction scheme to refine neural PDE predictions to extend rollout accuracies. Our work paves a step forward in predicting long-term dynamics of FROMP and other complex time-dependent problems.

1 Introduction

1.1 Frontal polymerization

Thermoset polymers are widely used in many industries, such as aerospace, transport, and energy sectors, due to their strong specific mechanical properties and thermo-chemical stability. Existing manufacturing of thermosets, via bulk curing, is energy-inefficient and unsustainable as it requires long curing duration at high temperatures in large autoclaves. Recent advancements in frontal ring-opening metathesis polymerization (FROMP) have enabled the rapid and stable curing of thermosets [18, 20]. In FROMP, the heat of polymerization released by the ring-opening metathesis reactions can propagate further FROMP. Since only an initial thermal trigger is required, FROMP can be an energy-efficient and sustainable alternative for thermoset manufacturing.

At the continuum, FROMP can be modeled as coupled thermo-chemical PDEs (see Eq. (1)) where T and α are the temperatures (in K) and degrees of cure (dimensionless) respectively. The reaction term provides the heat source in the exothermic reaction and the heat diffusion term describes the heat transport ahead of the advancing polymerization front. Unstable frontal polymerization (FP) occurs commonly for cyclooctadiene (COD), particularly when the initial conditions meet certain

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

thermal criteria, for example when the T_0 is low such that the reaction is too slow to initiate. Thermal instabilities in unstable FP lead to spontaneous patterning, and can result in materials with clearly defined regions of varying stiffness (by >2 orders of magnitude) [14, 19]. To predict these patterns, the FP PDE needs to be solved over longer timescales to extract the spatial variations in temperatures, particularly the $T_{\max}$. However, numerical simulations are computationally expensive, especially for 2D and 3D problems. For instance, solving a single FP PDE on a 2D domain can take 6 hours with 8 CPUs. A sufficiently small dt is needed for convergence, particularly for our problem where stiff reaction kinetics is present.

1.2 Neural operators and the rollout problem for time-dependent PDEs

Neural PDE models aim to learn a data-driven surrogate PDE solver to replace numerical solvers. Notably, neural operators learn the mapping between function spaces [8]. Some examples include the Fourier [9], Laplace [2], convolutional [17], and spectral neural operators [13, 3], and the use of transformers [10] and diffusion models [7]. For time-dependent problems, neural PDEs learn to predict solutions for each time step autoregressively [1] but they face the problem of error-accumulation over long temporal rollouts. In an autoregressive setting, predicted next-step solution is fed back into the neural operator to predict subsequent steps successively. Without convergence guarantees as in numerical solvers, the errors accumulate over multiple time steps. Some efforts have been made to improve long term rollout stabilities, such as rolling out longer time steps during training [1] and iteratively denoising the prediction over some refinement steps (PDE-refiner) [12].

Our reaction-diffusion problem necessitates an accurate neural PDE solver over long rollouts particularly because thermal instabilities are very sensitive to slight spatiotemporal changes in local temperatures and degrees of cure, especially at the propagation front. Herein, we adopt the Fourier neural operator (FNO) and augment it with large kernel attention layers and a flow matching-inspired correction step to extend the neural operator's temporal rollout for FROMP (Fig. 1).

2 Methods

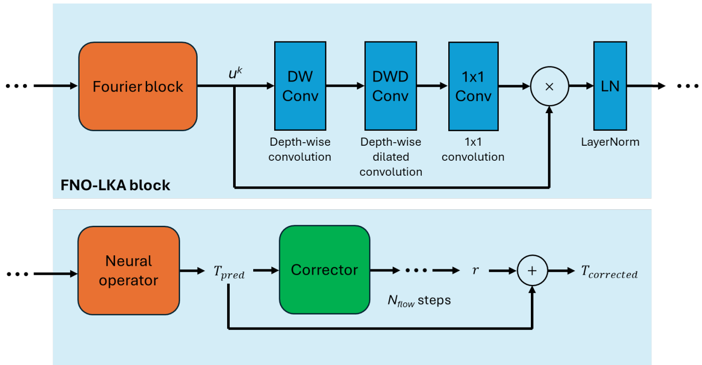

Figure 1: Model architecture and the ODE-flow correction scheme during inference

2.1 FNO with large kernel attention

Large kernel attention (LKA) was first proposed in visual attention networks [4] for image segmentation tasks and it was also used in a neural PDE setting for multiscale problems [22]. While FNO can capture long range dynamics, they fall short in predicting local instabilities over longer rollouts since higher frequency modes are truncated away in the Fourier layers. Thus, we appended an LKA layer after each Fourier block. LKA includes depth-wise convolution, depth-wise dilation convolution and a 1×1 kernel for channel convolution. An element-wise product is applied between the attention value map and the output from the Fourier block (Fig. 1). To stack alternate FNO and LKA blocks, layer normalization is included after every LKA block. In an FNOLKA model, low frequency features are generally learnt by the Fourier layers and high frequency and local features are captured by the LKA layers. More implementation details are in Appendix A.2.

2.2 Correcting predictions with ODE-flows

Flow matching [11, 21] is a useful generative modeling framework and it has been applied to materials design [15], protein generation [5], and accelerating molecular dynamics simulations [16]. In flow matching, samples from a prior distribution $r_0 \sim p_0(r)$ are transported to another distribution $r_1 \sim p_1(r)$ where the underlying model parameterizes the vector field. We propose a flow matching correction scheme to extend the long term rollout accuracies of spatiotemporal predictions. Herein, a corrector model is trained to refine predictions from a base neural PDE model. In our setting, the marginal vector field model $v_\tau(r;\theta)$ aims to learn the vector field from r_0 (random noise) to r_1 (true PDE residual) conditioned on the approximate prediction U_{pred} fully rolled out from a base model (i.e. another neural operator). Thus, the learned vector fields are different for different U_{pred}. By conditioning on a given U_{pred}, the transport converges to a single point of the true residual, rather than a distribution of residuals.

We adapted the functional flow matching (FFM) scheme [6] and use a neural operator as the underlying model. As an example, we use the FNOLKA architecture and adapted it to take in additional input channels: initial prediction U_{pred}, the current residual r_τ, the current flow time τ (temporal embedding), and spatial embedding x. Our FFM model is trained to directly predict the PDE residual given the U_{pred} from a neural operator. To construct a dataset, we rollout a trained FNOLKA$_{\rightarrow 20}$ model to obtain U_{pred} over 300 temporal steps for the 1D problem. The marginal vector field is defined as $v_\tau(r;\theta) = \frac{\hat{r}_1(r,\tau;\theta)-r}{1-\tau}$. During training, $\tau \sim \mathcal{U}(0,1)$ and the FFM model aims to predict the r_1 residual directly, and the loss function is the mean squared error between the true and predicted residuals. During inference (1), random noise is first sampled $r_0 \sim \mathcal{N}(0,\sigma^2)$ and the trained FFM corrector model (Corrector($U_{\mathrm{pred}}, r_\tau, x, \tau$)) predicts the residual $\hat{r_1}(x,t)$ for the entire rollout trajectory given the $U_{\mathrm{pred}}(x,t)$. The predicted residual is updated iteratively over $N_{\text{flow}} = 10$ steps by interpolating with $v_\tau(r;\theta)$. The final corrected prediction is simply given by $U_{\mathrm{corrected}} = U_{\mathrm{pred}} + r_1$.

Algorithm 1 Correction algorithm during inference

```
input  Initial prediction $U_{\mathrm{pred}}(x,t)$ from a neural operator
output  Corrected prediction $U_{\mathrm{corrected}}(x,t)$
$r_0 \sim \mathcal{N}(0,\sigma^2)$
for $i \leftarrow 0$ to $N_{\text{flow}} - 1$ do
    $\tau \leftarrow i/N_{\text{flow}}$
    $\tilde{r_1} \leftarrow$ Corrector($U_{\mathrm{pred}}, r_\tau, x, \tau$)
    $r_{\tau'} \leftarrow r_\tau + (\tilde{r_1} - r_\tau)/(1-\tau)N_{\text{flow}}$
end for
$U_{\mathrm{corrected}} \leftarrow U_{\mathrm{pred}} + r_1$
```

3 Results

3.1 Evaluating models and the correction scheme over long rollouts for the 1D problem

We trained both the FNO and FNOLKA models with 10-step or 20-step unrolling to predict the evolution of $T(x,t)$ to model heat transport in frontal polymerization. Given the previous 10 steps of PDE solution, the model predicts the next-step T_{pred} before it is fed back with the previous 9 steps to predict subsequent steps autoregressively. We find that training with a relative Sobolev H^1 norm improves performance, particularly for our problem where both T and α have sharp gradients at the propagation fronts. The loss is accumulated for the next 10 or 20 unrolled steps during training. To compare long term rollout accuracies, we rollout the prediction trajectory up to 300 steps forward using only the first 10 steps of T_{true} as the initial input. For the corrector model, trained to refine FNOLKA$_{\rightarrow 20}$, we take the full prediction rollout T_{pred} as input for the corrector to refine over 10 N_{flow} steps. To avoid distribution shifts, instead of predicting a correction at every time step and feeding it back to a trained neural operator successively, the corrector herein directly refines $T_{\mathrm{pred}}(x,t)$ up to 300 time steps at once. Fig. 2 shows an example of prediction rollouts from an FNO, FNOLKA, and an FNOLKA with a corrector model (more examples are in Appendix A.3).

To compare the long-term rollout accuracies of the models, we rollout the FNO and FNOLKA (with and without correction) models over 300 steps ahead and evaluate the relative L_2 losses at every step.

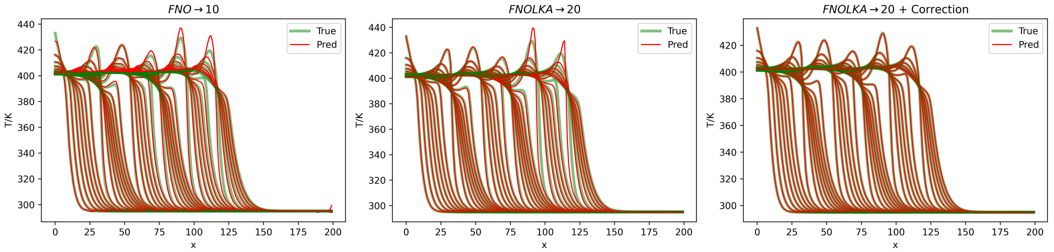

Figure 2: Prediction rollout over 200 steps on a test set sample by FNO, FNOLKA, and FNOLKA with flow matching correction.

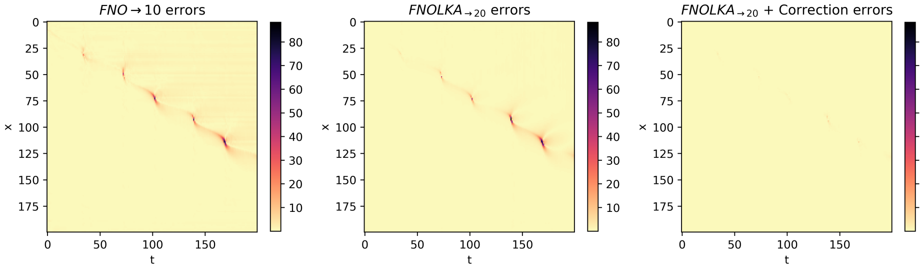

Figure 3: Prediction errors over 200 steps for the same test set sample by FNO, FNOLKA, and FNOLKA with flow matching correction.

The mean performances and their max-min ranges for all 308 test samples are plotted for each model and scheme (Fig. 4). We find that while adding LKA layers help reduce the mean rollout errors, the rate of error accumulation is faster for the FNOLKA$_{\rightarrow 10}$ model trained to predict the next 10 steps as it fails to generalize to the longer-term dynamics. Forcing the model to predict more steps, i.e. next 20 steps, during training helps to stabilize the rollout errors for both FNO$_{\rightarrow 20}$ and FNOLKA$_{\rightarrow 20}$. The combination of both LKA layers and longer unrolled training works best in achieving long rollout stabilities for the 1D problem.

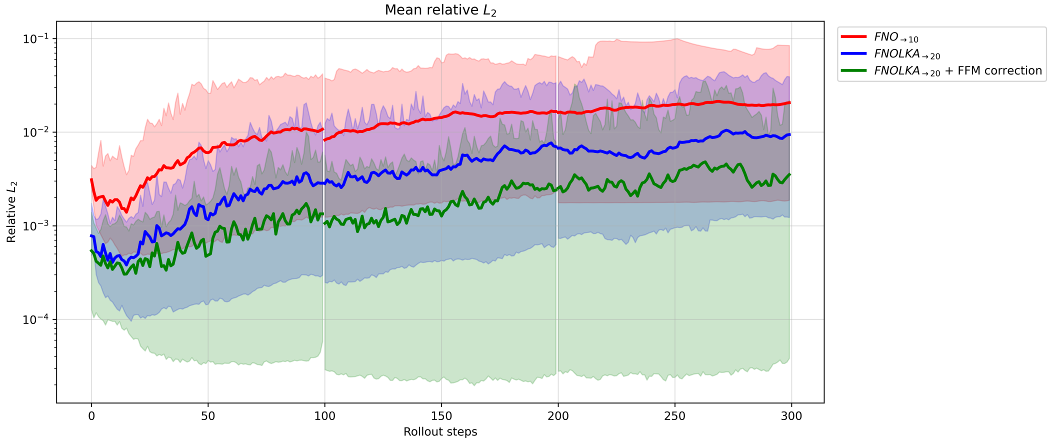

Figure 4: Mean and ranges of relative L_2 errors in predictions over long rollouts for different models

For the flow matching-based corrector model, the predictions from FNOLKA$_{\rightarrow 20}$ are effectively refined, extending long-term rollout accuracies. Errors, particularly due to instabilities in the prediction rollout, are corrected by the FFM model at each time step. We find that correction of $N_{\text{flow}} = 2$ steps is sufficient for refinement, thus our approach is efficient during inference. While the difference in mean relative L_2 is small between the FNOLKA$_{\rightarrow 20}$ models without and with correction for the first 20 rollout steps, the difference is apparent over longer temporal rollout. Specifically at the 300th step, there is an improvement in mean L_2 by 62.6%. Therefore, including a FFM correction step is effective in refining long-term rollout trajectories, specifically to predict spatiotemporal dynamics of frontal polymerization in our current setting. More work is ongoing to evaluate our flow matching correction scheme on other problems over long temporal rollouts, including other complex time-dependent PDEs and climate forecasting.

3.2 Extracting morphogenic patterns from unrolled trajectories

Extracted $T_{\max}(x, y)$ values can predict morphogenic patterns of spatially varying mechanical properties in pCOD thermosets. To predict $T_{\max}$ of the entire domain, we need to rollout predictions from the neural operator over many time steps. For this task, similarly, we trained an FNOLKA model on 2D numerical solutions that were generated by varying initial and boundary conditions (Appendix A.1).

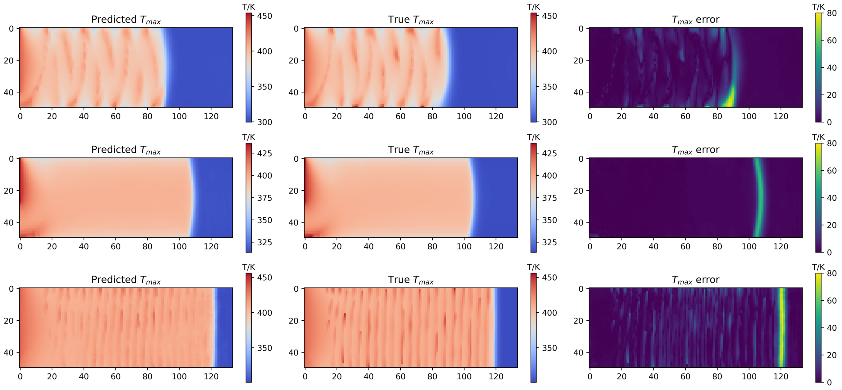

Figure 5: Predicted $T_{\max}$ extracted from 2D FNOLKA rollouts on test set examples

Generally, the predicted $T_{\max}(x, y)$ match the true values well and the trained FNOLKA$_{\rightarrow 20}$ can predict the spatiotemporal dynamics of FROMP for both stable and unstable cases. Nevertheless, instabilities that result in local complex patterns are still challenging for the model to learn. For future work, we will similarly train a FFM corrector model to refine these predictions.

4 Conclusion

By refining spatiotemporal predictions with a flow matching-based corrector model, augmenting FNO with large kernel attention layers and adopting longer temporal rollout during training, we demonstrated that longer rollout accuracies can be achieved. In this work, we proposed a correction scheme, inspired by conditional functional flow matching, to refine predictions from a neural operator. Our corrector model shows promise in refining spatiotemporal predictions and have reduced errors by 62.6% (at the 300th step). Our work presents a step forward in predicting spatiotemporal dynamics and spontaneous patterning in frontal polymerization, and it can be extended to other complex, time-dependent problems, such as climate forecasting, where the underlying dynamics and complete governing PDEs remain unknown.

Acknowledgments and Disclosure of Funding

This work was supported as part of the Regenerative Energy-Efficient Manufacturing of Thermoset Polymeric Materials (REMAT), an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences under award DE-SC0023457.

References

[1] Brandstetter, J., Worrall, D., and Welling, M. Message Passing Neural PDE Solvers, March 2022. URL http://arxiv.org/abs/2202.03376. arXiv:2202.03376 [cs, math].

[2] Cao, Q., Goswami, S., and Karniadakis, G. E. LNO: Laplace Neural Operator for Solving Differential Equations, May 2023. URL http://arxiv.org/abs/2303.10528. arXiv:2303.10528 [cs].

[3] Du, Y., Chalapathi, N., and Krishnapriyan, A. Neural spectral methods: Self-supervised learning in the spectral domain, 2024. URL https://arxiv.org/abs/2312.05225.

[4] Guo, M.-H., Lu, C.-Z., Liu, Z.-N., Cheng, M.-M., and Hu, S.-M. Visual attention network, 2022. URL https://arxiv.org/abs/2202.09741.

[5] Jing, B., Berger, B., and Jaakkola, T. AlphaFold meets flow matching for generating protein ensembles. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22277–22303. PMLR, 21–27 Jul 2024.

[6] Kerrigan, G., Migliorini, G., and Smyth, P. Functional Flow Matching, December 2023. URL http://arxiv.org/abs/2305.17209. arXiv:2305.17209 [cs, stat].

[7] Kohl, G., Chen, L.-W., and Thuerey, N. Benchmarking Autoregressive Conditional Diffusion Models for Turbulent Flow Simulation.

[8] Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural Operator: Learning Maps Between Function Spaces, April 2023. URL http://arxiv.org/abs/2108.08481. arXiv:2108.08481 [cs, math].

[9] Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations, 2021.

[10] Li, Z., Meidani, K., and Farimani, A. B. Transformer for Partial Differential Equations' Operator Learning, April 2023. URL http://arxiv.org/abs/2205.13671. arXiv:2205.13671 [cs].

[11] Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[12] Lippe, P., Veeling, B. S., Perdikaris, P., Turner, R. E., and Brandstetter, J. PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers, October 2023. URL http://arxiv.org/abs/2308.05732. arXiv:2308.05732 [cs].

[13] Liu, Z., Wu, Y., Huang, D. Z., Zhang, H., Qian, X., and Song, S. SPFNO: Spectral operator learning for PDEs with Dirichlet and Neumann boundary conditions, December 2023. URL http://arxiv.org/abs/2312.06980. arXiv:2312.06980 [cs, math].

[14] Lloyd, E. M., Feinberg, E. C., Gao, Y., Peterson, S. R., Soman, B., Hemmer, J., Dean, L. M., Wu, Q., Geubelle, P. H., Sottos, N. R., and Moore, J. S. Spontaneous Patterning during Frontal Polymerization. *ACS Central Science*, 7(4):603–612, April 2021. ISSN 2374-7943. doi: 10.1021/acscentsci.1c00110. URL https://doi.org/10.1021/acscentsci.1c00110. Publisher: American Chemical Society.

[15] Miller, B. K., Chen, R. T. Q., Sriram, A., and Wood, B. M. FlowMM: Generating Materials with Riemannian Flow Matching, June 2024. URL http://arxiv.org/abs/2406.04713. arXiv:2406.04713 [cond-mat, physics:physics, stat].

[16] Nam, J., Liu, S., Winter, G., and Gómez-Bombarelli, R. Generative acceleration of molecular dynamics simulations for solid-state electrolytes.

[17] Raonić, B., Molinaro, R., De Ryck, T., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., and de Bézenac, E. Convolutional Neural Operators for robust and accurate learning of PDEs, May 2023. URL http://arxiv.org/abs/2302.01178. arXiv:2302.01178 [cs].

[18] Robertson, I. D., Yourdkhani, M., Centellas, P. J., Aw, J. E., Ivanoff, D. G., Goli, E., Lloyd, E. M., Dean, L. M., Sottos, N. R., Geubelle, P. H., Moore, J. S., and White, S. R. Rapid energy-efficient manufacturing of polymers and composites via frontal polymerization. *Nature*, 557(7704):223–227, May 2018. doi: 10.1038/s41586-018-0054-x.

[19] Sottos, N., Paul, J., Gao, Y., Go, Y. K., Koett, L. R., Sharma, A., Chen, M., Lessard, J., Topkaya, T., Leal, C., Moore, J., and Geubelle, P. Molecularly Architected Polymers Enabled by Frontal Polymerization, January 2024. URL https://www.researchsquare.com/article/rs-3777388/v1. ISSN: 2693-5015.

[20] Suslick, B. A., Hemmer, J., Groce, B. R., Stawiasz, K. J., Geubelle, P. H., Malucelli, G., Mariani, A., Moore, J. S., Pojman, J. A., and Sottos, N. R. Frontal polymerizations: From chemical perspectives to macroscopic properties and applications. *Chemical Reviews*, 123(6):3237–3298, 2023. doi: 10.1021/acs.chemrev.2c00686. PMID: 36827528.

[21] Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024. URL `https://arxiv.org/abs/2302.00482`.

[22] Zhao, X., Sun, Y., Zhang, T., and Xu, B. Local Convolution Enhanced Global Fourier Neural Operator For Multiscale Dynamic Spaces Prediction, November 2023. URL `http://arxiv.org/abs/2311.12902`. arXiv:2311.12902 [cs, math].

A Appendix

A.1 Data generation

The frontal polymerization in cyclooctadiene (COD) can be described by the following coupled thermo-chemical reaction-diffusion equations in terms of temperature T and degree of cure α:

$$\rho C_p \frac{\partial T}{\partial t} = \kappa \nabla^2 T + \rho H_r \frac{\partial \alpha}{\partial t} \tag{1a}$$

$$\frac{\partial \alpha}{\partial t} = A \exp\left(-\frac{E}{RT}\right) \alpha^m (1-\alpha)^n \tag{1b}$$

The cure kinetics parameters, A, E, m and m in Eq. (1b) were extracted from non-linear fitting of differential scanning calorimetry (DSC) scans. The material properties and cure kinetics parameters for COD are listed in Table 1.

Table 1: Cure kinetics and material properties of COD.

$\kappa \left(\frac{\mathrm{W}}{\mathrm{mK}}\right)$	$\rho \left(\frac{\mathrm{kg}}{\mathrm{m}^3}\right)$	$C_p \left(\frac{\mathrm{J}}{\mathrm{kg}K}\right)$	$A \left(\frac{1}{\mathrm{s}}\right)$
0.133	882	1838.5	2.13×10^{19}
$E \left(\frac{\mathrm{kJ}}{\mathrm{mol}}\right)$	n	m	$H_\mathrm{r} \left(\frac{\mathrm{J}}{\mathrm{g}}\right)$
132	2.514	0.817	220.0

We utilized the finite element method and the open-source package, FEniCS, to solve Eq. (1) to generate numerical solutions for model training. For 1D problems, a 1D domain with length of 10 mm was simulated, where the mesh element size and time step are $\Delta x = 10\ \mu\mathrm{m}$ and $\Delta t = 1\ \mathrm{ms}$ respectively. For boundary conditions, a thermal trigger T_{trig} is applied on the left end of the resin (over $x = 0$) to start the reaction and the other end is adiabatic. Various initial temperature and degrees of cure in a Sobol sequence with scrambling in space of $T_0 \in [10, 40]\ ^oC$, $\alpha_0 \in [0.01, 0.3]$ are used to generate a diverse dataset.

For the 2D problem, a rectangular domain with size of $20 \times 5\ \mathrm{mm}^2$ was simulated, and the discretizations were set as $\Delta x = 10\ \mu\mathrm{m}$, $\Delta y = 20\ \mu\mathrm{m}$, and $\Delta t = 1\ \mathrm{ms}$. On a fraction of the left-end ($x = 0$ and $0 \leq y \leq d$), a thermal trigger T_{trig} is applied to start the reaction. A heat convection boundary condition (with heat transfer coefficient h_L) is imposed on the top and bottom of the domain. The other boundaries are adiabatic. To generate a diverse dataset, process conditions of initial temperatures, initial degrees of cure, heat loss coefficient, and the trigger width are generated in Sobol sequence with scramble in space of $T_0 \in [10, 40]\ ^oC$, $\alpha_0 \in [0.01, 0.3]$, $h_L \in [0, 120]\ \frac{W}{K \cdot m^2}$ and $d \in [1.5, 5]\ \mathrm{mm}$ respectively.

A.2 Model training

For all models for the 1D problems, we employ FNOs with 16 Fourier modes and a channel width of 64, and incorporating spatial embedding as an additional input. GeLU activation functions were used and instance normalizations are included between spectral convolution layers. For LKA-based models, the LKA channel width is 64. For all models, a total of 4 Fourier or Fourier-LKA blocks are used. The dataset is divided into an 80-20 train-test split, with each input-output pair comprising 10 previous steps and 20 subsequent steps, with each step's solution having a spatial dimension of 200. In total, our dataset consists of 29,481 training samples and 7,383 test samples, and we utilize a batch size of 64. We train all models using an AdamW optimizer with a learning rate of $6e^{-5}$, weight decay of $1e^{-4}$, and a cosine annealing learning rate scheduler over 500 epochs. During each batch, the model performs either a 10-step or 20-step forward pass, and the relative H^1 loss over the subsequent 10 or 20 steps is accumulated before backpropagation.

For the 2D problem, we only trained an FNOLKA model but with similar hyperparameters. A batch size of 16 was used and we have a 85-15 train-test split (1178 and 208 solutions for train and test respectively). The model takes in 10 previous steps and the spatial (x, y) embeddings to predict the next step's solution. In each step, the prediction is appended to the previous 9 steps for the model to predict the subsequent step. During training, the model predicts the next 20 steps and a relative H^1 loss was also used.

For the corrector model, we train either an FNO or FNOLKA model to predict the residual r_1 given the flow time $\tau \sim U(0,1)$, the current interpolated r_τ, spatial embedding x and conditioned on the T_{pred} in each batch. For both architectures, we use a channel width of 64 and 16 Fourier modes. We use an AdamW optimizer, a cosine annealing learning rate scheduler, a batch size of 128 and a learning rate of $1e^{-3}$ to train the models over 1000 epochs. To generate training data, we used the trained FNOLKA$_{\rightarrow 20}$ model to predict the full temporal rollout up to 500 steps.

A.3 Supplementary figures

Here, we show more unrolled predictions by different models and schemes for the 1D time-dependent problem.

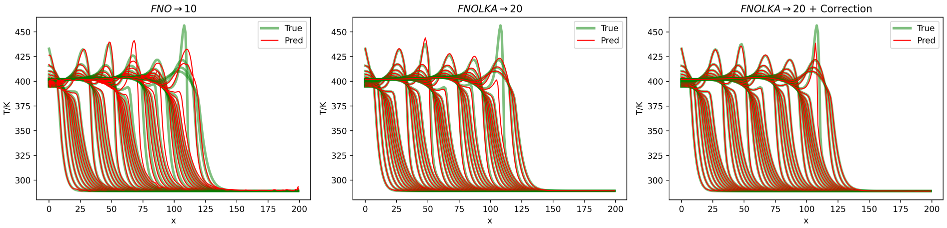

Figure 6: Prediction rollouts over 200 steps for test-set example 2 by different models

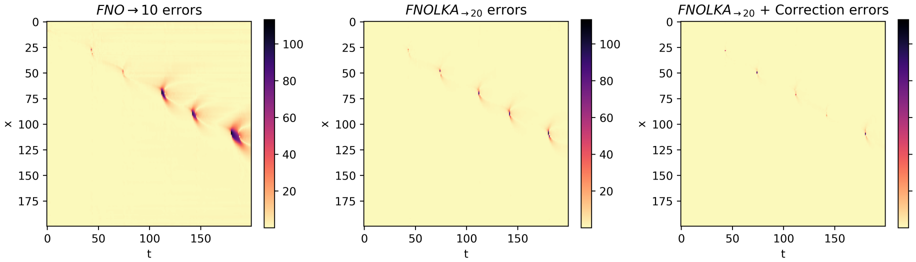

Figure 7: Prediction errors over 200 steps for test-set example 2 by different models

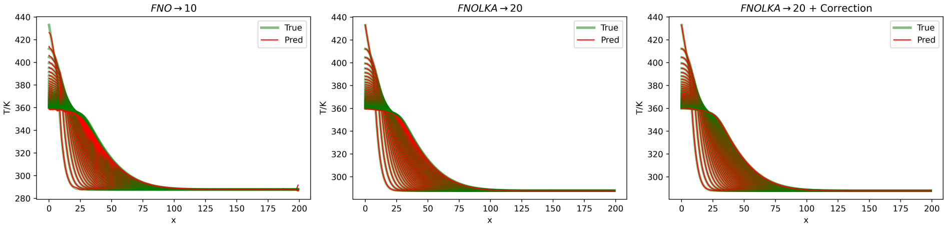

Figure 8: Prediction rollouts over 200 steps for test-set example 3 by different models

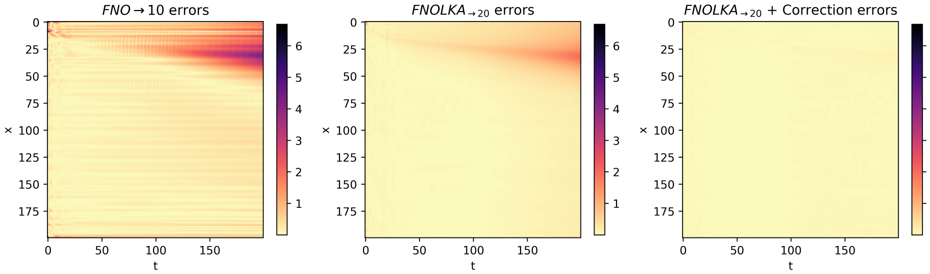

Figure 9: Prediction errors over 200 steps for test-set example 3 by different models

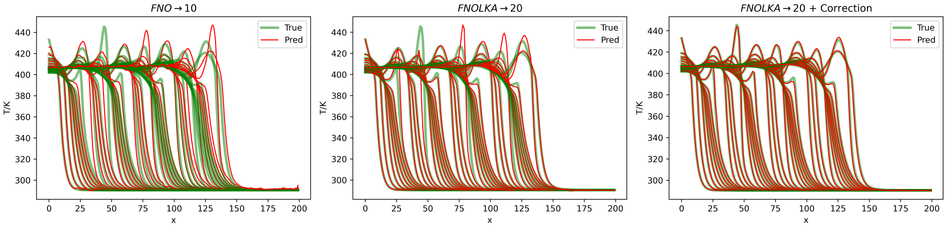

Figure 10: Prediction rollouts over 200 steps for test-set example 4 by different models

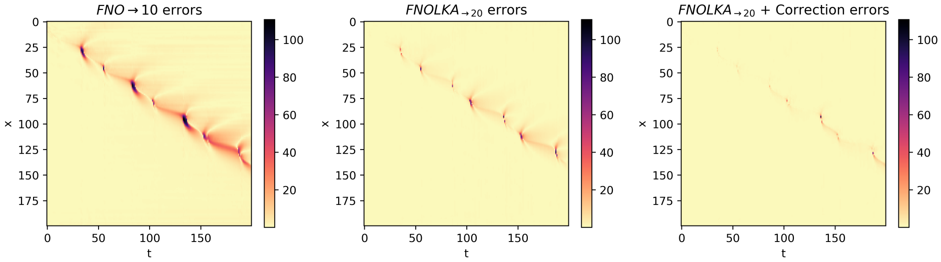

Figure 11: Prediction errors over 200 steps for test-set example 4 by different models

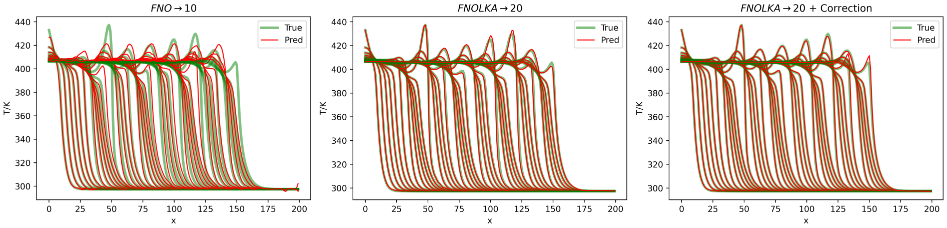

Figure 12: Prediction rollouts over 200 steps for test-set example 5 by different models

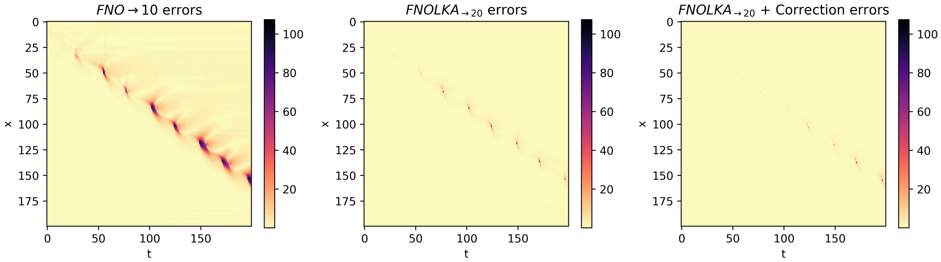

Figure 13: Prediction errors over 200 steps for test-set example 5 by different models