NEURAL CONSTITUTIVE LEARNING FOR GENERALIZED REACTION-DIFFUSION SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalized reaction-diffusion systems encompass diverse transport mechanisms and coupled reaction kinetics. A central question for neural PDE solvers is what should be learned so that a common interface can accommodate phase-field and degenerate transport, local reactions, and multispecies coupling. We propose the Neural Constitutive Laws–Mass-Compression-Transport (NCL-MCT) Solver, which learns PDE-specific constitutive responses while retaining temporal evolution in a shared MCT integrator. Transport is represented through mobility and thermodynamic driving force, and reaction through relative reaction rates. These constitutive responses depend on the current density rather than explicitly on the initial condition or elapsed time, motivating their reuse across different initial conditions and time horizons. The same interface supports velocity-data supervision and known-law supervision, neither of which requires time integration during training. When constitutive laws are known, supervision can be evaluated on independently sampled density fields, enabling trajectory-free constitutive learning without generating solution trajectories. Across seven systems, separately trained constitutive modules share the same interface and MCT integrator and achieve relative rollout L^2 errors of 10^{-4} to 10^{-2}. Tests with unseen initial-condition families and an extended time horizon assess reuse beyond training conditions, while separate experiments demonstrate trajectory-free constitutive learning. These results support constitutive responses as an effective learning target for a shared neural PDE framework.

1 INTRODUCTION

Generalized reaction-diffusion partial differential equations (PDEs) describe the evolution of spatial densities through transport, diffusion, and local production or depletion. They underpin models of biological pattern formation, tumor growth, and intracellular condensation (Turing, 1952; Lu et al., 2022; Groves et al., 2026). Their dynamics span linear, nonlinear, degenerate, and higher-order diffusion, together with local and coupled reactions (Cahn & Hilliard, 1958; Vázquez, 2006). These processes may also couple with pressure, fluid flow, or mechanical deformation (Lu et al., 2019; 2022). Despite this diversity, density balance provides a common evolution structure, while constitutive laws determine the system-specific transport and reaction responses.

Most neural PDE methods learn the solution or its evolution: physics-informed neural networks (PINNs) represent solution fields directly (Raissi et al., 2019), neural operators learn solution maps (Li et al., 2021; Lu et al., 2021), and other approaches learn temporal derivatives for numerical integration (Hou et al., 2026). Constitutive laws instead close governing equations by specifying the physical responses that distinguish one system from another. In solid mechanics, neural constitutive laws have shown that such responses can be learned while retaining the governing evolution (Ma et al., 2023). This suggests a complementary learning target for generalized reaction-diffusion systems: learning PDE-specific constitutive responses while retaining the shared evolution structure. The remaining challenge is to define a common constitutive interface that accommodates phase-field and degenerate transport, local reactions, and multispecies coupling, while supporting both velocity-data supervision and known-law supervision.

We address this problem with the Neural Constitutive Laws–Mass-Compression-Transport (NCL-MCT) Solver, which separates PDE-specific constitutive responses from shared temporal kinemat-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ics. Guided by the energetic variational approach (EnVarA) (Hyon et al., 2010), the constitutive interface represents transport through mobility and thermodynamic driving force, and reaction through relative reaction rates. The constitutive modules learn these responses, while the shared MCT integrator evolves the density through the decomposition $\rho = MI$, separating material mass from compression associated with local volume change. Each PDE trains its own constitutive modules while sharing the same interface and MCT integrator, under either velocity-data supervision or known-law supervision. When known constitutive laws are evaluated on independently sampled density fields, the same framework further enables trajectory-free constitutive learning without generating solution trajectories. For the autonomous systems considered here, the constitutive responses depend on the current density rather than explicitly on the initial condition or elapsed time, motivating their reuse across different initial conditions and time horizons.

We evaluate NCL-MCT Solver on seven systems spanning phase-field and degenerate transport, local reactions, and coupled species. Across these systems, the shared interface achieves relative rollout L^2 errors of 10^{-4} to 10^{-2} and is competitive with or better than the evaluated baselines on most systems. Without retraining, the learned constitutive modules achieve roughly $6\times$ lower error on unseen initial-condition families than the strongest finite baseline. Over an extended time horizon of $1.67\times$ the training horizon, their errors increase by at most $1.5\times$, compared with up to $5\times$ for the evaluated baselines, while remaining an order of magnitude lower in absolute error. Our contributions are threefold:

- We formulate constitutive learning for generalized reaction-diffusion systems, representing transport through mobility and thermodynamic driving force and reaction through relative reaction rates, with velocity-data or known-law supervision, including trajectory-free constitutive learning from independently sampled density fields.
- We develop NCL-MCT Solver, which couples PDE-specific constitutive modules to a shared MCT integrator through a common constitutive interface.
- We demonstrate cross-system applicability across seven systems and evaluate reuse of the learned constitutive modules, without retraining, on unseen initial-condition families and an extended time horizon.

2 Related Work

2.1 Solution Learning with Physical Constraints

Physics-informed neural networks (PINNs) represent PDE solutions directly and impose governing equations through residual losses (Raissi et al., 2019; Karniadakis et al., 2021), while neural operators such as FNO (Li et al., 2021), DeepONet (Lu et al., 2021), and CNO (Raonić et al., 2023) learn mappings between input functions and solution fields. Physical structure can further enter through PDE residuals, conservation, hard constraints, architectural constraints, or energy-based regularization (Li et al., 2024; Hansen et al., 2023; Hao et al., 2026; Liu et al., 2024a; Tanaka et al., 2025). These approaches constrain the solution representation or prediction, whereas constitutive learning places the trainable model in the physical response laws used by an explicit evolution procedure.

2.2 Time-Derivative Learning with Numerical Integration

TI-DeepONet (Nayak & Goswami, 2026) and PITI-DeepONet (Mandl et al., 2026) learn temporal derivative operators and advance solutions through numerical integration, while CFO (Hou et al., 2026) learns the complete PDE right-hand side through flow matching. These approaches retain explicit time integration but learn the state evolution at the derivative level. In contrast, constitutive learning separates transport and reaction into PDE-specific physical responses that are supplied to a shared numerical integrator.

2.3 Component Learning within Numerical Solvers

Learned numerical solvers may replace selected discretizations or physical components while retaining established PDE update procedures (Bar-Sinai et al., 2019; Kochkov et al., 2021). FINN (Karlbauer et al., 2022), PeRCNN (Rao et al., 2023), PAPM (Liu et al., 2024b), and FluxGNN (Horie &

Mitsume, 2024) learn fluxes, sources, nonlinear terms, or conservative exchanges within structured numerical formulations. These approaches preserve physical organization through component definitions and numerical assembly. In contrast, NCL-MCT Solver focuses specifically on PDE-specific constitutive responses and couples them to a shared MCT integrator through a common constitutive interface.

2.4 Constitutive and Variational Learning

Constitutive laws close governing equations by specifying system-specific physical responses, with classical examples in elasticity and plasticity (Treloar, 1943; Arruda & Boyce, 1993; Drucker & Prager, 1952). Variational formulations further organize these responses through energy, dissipation, and kinematics. EnVarA (Hyon et al., 2010) provides such a structure by relating energetic and dissipative forces. EVNN (Hu et al., 2024) and energy-dissipation approaches (Lu et al., 2026a;b) embed related principles into learning and numerical evolution, while VONNs (Huang et al., 2022) directly learns free-energy and dissipation potentials from macroscopic observations. Stat-PINNs (Huang et al., 2025) further learns free energy and dissipative operators from short-time particle data, addressing non-uniqueness in thermodynamic identification. Recent differentiable phase-field models infer free energies and kinetic laws through PDE evolution (Cohen et al., 2026). In mechanics, NCLaw (Ma et al., 2023), UniPhy (Mittal et al., 2025), and PC-NCLaws (Xie et al., 2025) learn material constitutive responses within numerical simulators. DOOL (Chang et al., 2026) learns dissipative fluxes or change rates by minimizing an Onsager Rayleighian without solution labels and advances them through explicit conservation or change laws. Together, these methods establish physical closures as a learning target, but differ in the learned quantities and evolution mechanisms. For generalized reaction-diffusion systems, NCL-MCT Solver uses a common constitutive interface: mobility and thermodynamic driving force for transport, and relative reaction rates for reaction. These PDE-specific responses are coupled to a shared MCT integrator and can be supervised directly without differentiating through time integration.

3 Preliminaries

Following the energetic variational approach (EnVarA) (Hyon et al., 2010; Wang et al., 2020), we organize generalized reaction-diffusion systems into **transport and reaction constitutive laws** coupled to **shared temporal kinematics**. For a smooth, positive density ρ on $\Omega \subset \mathbb{R}^d$ with problem-specific boundary conditions, the density balance is

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = \rho r. \tag{1}$$

Constitutive laws from energetic variations. The constitutive laws close Equation 1 by specifying the transport velocity $\mathbf{u}$ and relative reaction rate r. For a free energy $E[\rho]$ and quadratic dissipation $\mathcal{D} = \int_\Omega \eta |\mathbf{u}|^2 \, d\mathbf{x}$, the maximum dissipation principle gives $\eta \mathbf{u} = -\rho \nabla (\delta E / \delta \rho)$, hence

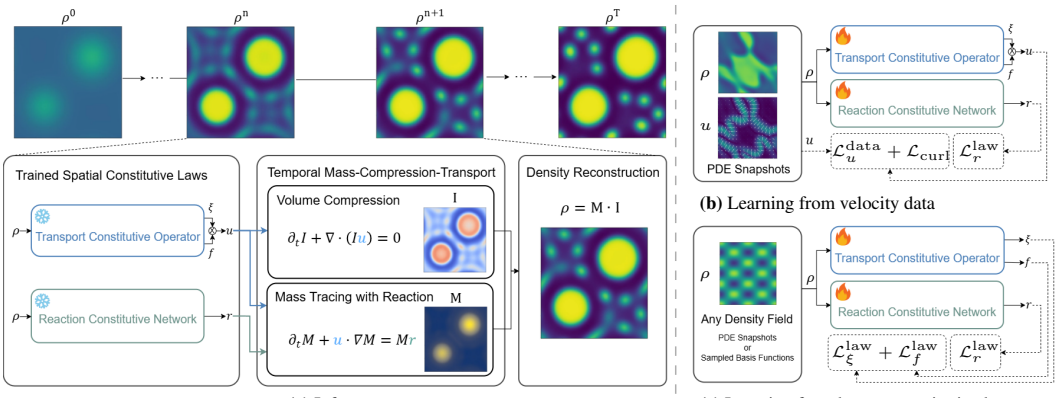
$$\mathbf{u} = \xi \mathbf{f}, \qquad \xi = \frac{\rho}{\eta} > 0, \qquad \mathbf{f} = -\nabla \frac{\delta E}{\delta \rho}, \qquad r = \frac{S(\rho)}{\rho}, \tag{2}$$

where ξ is the transport mobility, $\mathbf{f}$ the thermodynamic driving force, and $S(\rho)$ the reaction source. System-specific choices are listed in Appendix D.

Although specifying E and η determines both constitutive factors, $(\lambda\xi, \mathbf{f}/\lambda)$ yields the same $\mathbf{u}$ for any $\lambda > 0$. Thus, velocity data constrain only their product, while separate identification of mobility and force requires additional constitutive information; related identifiability issues also arise in thermodynamic learning (Huang et al., 2025). The factorization nevertheless provides structure: $\xi > 0$, while $\mathbf{f} = -\nabla(\delta E / \delta \rho)$ is curl-free for smooth fields. Transport and reaction also enter the balance differently: $\mathbf{u}$ acts through a divergence, whereas r acts pointwise, motivating separate constitutive representations and numerical treatment.

Mass-compression-transport kinematics. The MCT formulation separates these effects through the decomposition $\rho = MI$:

$$\partial_t I + \nabla \cdot (I \mathbf{u}) = 0, \qquad \partial_t M + \mathbf{u} \cdot \nabla M = Mr, \tag{3}$$



(a) Inference **(b)** Learning from velocity data **(c)** Learning from known constitutive laws

Figure 1: Overview of NCL-MCT Solver. (a) At inference, frozen constitutive modules are coupled to the shared MCT integrator. At each step, the transport velocity $\mathbf{u} = \xi\mathbf{f}$ and relative reaction rate r are reevaluated from the current density to evolve I and M and reconstruct $\rho = MI$. During training, transport is supervised either (b) from reference velocities on PDE snapshots, with curl regularization, or (c) directly from known mobility and force laws on density fields, including independently sampled density fields. For reactive systems, r is supervised by the known reaction law in both settings.

with $I(\mathbf{x}, 0) = 1$ and $M(\mathbf{x}, 0) = \rho(\mathbf{x}, 0)$. For a regular material flow with local volume ratio J, $I = J^{-1}$ records compression or expansion, while $M = \rho J$ represents mass per unit reference volume. Along a material trajectory, M changes through reaction according to $M = \rho^0 \exp\big(\int_0^t r\, ds\big)$, and the product $\rho = MI$ recovers the original density balance. Appendix A derives these identities. The representation assumes $\rho > 0$ and a regular material flow; near vanishing density, the relative reaction rate and factorization require additional care.

4 Methodology

To share one solver interface across diffusive, phase-field, reactive, and multispecies systems, we introduce NCL-MCT Solver, which couples PDE-specific constitutive modules to a shared mass-compression-transport (MCT) integrator. Following the EnVarA formulation in Section 3, transport is represented through positive mobility and thermodynamic driving force, and reaction through relative reaction rates. For the autonomous systems considered here, these constitutive responses depend on the current density rather than explicitly on the initial condition or elapsed time. Figure 1 summarizes the framework. A **Transport Constitutive Operator** learns the mobility-force factorization, while a **Reaction Constitutive Network** learns relative reaction rates from local species densities (Section 4.1). At inference, the frozen constitutive modules are reevaluated at each MCT step to advance the compression and mass factors and reconstruct the density (Section 4.2). For each PDE, the constitutive modules are trained separately under either velocity-data supervision or known-law supervision, without time integration during training (Section 4.3).

4.1 Neural Constitutive Models

Transport constitutive laws depend on spatial variation of the density field, whereas the reaction laws considered here are local functions of species densities. We therefore represent transport with a field-to-field operator and reaction with a pointwise network.

The **Transport Constitutive Operator** uses a Convolutional Neural Operator (CNO) (Raonić et al., 2023). The transport responses in Equation 2 involve spatial differential operators of the current density field, up to third order in the phase-field systems. Local convolutions capture this differential structure, while the multiscale architecture spans the spatial scales over which these responses act. This local representation is also well suited to compactly supported densities in degenerate transport systems. Following the EnVarA mobility-force form, we parameterize

$$(\widehat{\xi}, \widehat{\mathbf{f}}) = \mathcal{C}_{\theta_u}[\rho], \qquad \widehat{\mathbf{u}} = \widehat{\xi}\,\widehat{\mathbf{f}}, \tag{4}$$

where ρ is the density field and $\mathcal{C}_{\theta_u}$ is the CNO with trainable parameters θ_u. A Softplus activation enforces $\widehat{\xi} > 0$ by construction. Known-law supervision directly constrains the driving force, while under velocity-data supervision its gradient structure is encouraged by the curl penalty (Section 4.3).

The **Reaction Constitutive Network** is a multilayer perceptron (MLP) applied pointwise:

$$\widehat{r} = \mathcal{R}_{\theta_r}[\rho], \tag{5}$$

where $\mathcal{R}_{\theta_r}$ maps the current local species densities to relative reaction rates. This preserves the locality of the reaction law; predicting the relative reaction rate rather than the source term also lets reaction enter the mass-factor evolution multiplicatively through Equation 3. Nonreactive systems set $r = 0$ and omit this module. Network architectures are detailed in Appendix B.

4.2 Shared MCT Time Integration

The MCT integrator contains no trainable update rule. Given the current density ρ^n, the constitutive modules provide the transport velocity and relative reaction rates required by the shared MCT integrator. Each time step applies a reaction half-step, a transport step, and a second reaction half-step. The transport velocity is evaluated from ρ^n and held fixed within the step, while the relative reaction rates are reevaluated at each reaction stage. The compression factor I is advanced with a conservative finite-volume method (Eymard et al., 2000), the mass factor M is transported with a semi-Lagrangian method (Staniforth & Côté, 1991), and their product reconstructs $\rho^{n+1} = M^{n+1}I^{n+1}$. This splitting mirrors the separation of transport and reaction in the continuous formulation, with I tracking local compression and M tracking material mass under transport and reaction. Because the integrator only queries constitutive responses, numerically evaluated and learned constitutive responses enter through the same interface (Appendix I.5). The complete update does not guarantee exact discrete mass conservation of the reconstructed density or unconditional discrete energy dissipation; numerical details and empirical diagnostics are provided in Appendices C and I.2.

4.3 Learning Constitutive Laws

We train the constitutive modules under two sources of supervision: **velocity-data supervision** and **known-law supervision**. The supervision source is distinct from the source of input density fields, which may be trajectory snapshots or independently sampled density fields. For reactive systems, both modes use known-law supervision for the relative reaction rate. Neither mode requires time integration during training.

Velocity-data supervision. When paired density-velocity observations are available, the transport response can be learned without specifying the underlying mobility and force laws:

$$\mathcal{L}_u^{\text{data}} = \ell(\widehat{\mathbf{u}}, \mathbf{u}^{\text{data}}), \qquad \widehat{\mathbf{u}} = \widehat{\xi}\widehat{\mathbf{f}}, \tag{6}$$

where $\mathbf{u}^{\text{data}}$ is the reference transport velocity and $\ell(\widehat{q}, q) = B^{-1}\sum_b \|\widehat{q}^{(b)} - q^{(b)}\|_2^2 / (\|q^{(b)}\|_2^2 + \varepsilon)$ is the per-sample relative squared error over a batch of B density fields, with $\varepsilon = 10^{-12}$. This normalization prevents low-magnitude constitutive quantities from being underweighted.

Velocity data constrain only the product $\widehat{\xi}\widehat{\mathbf{f}}$; they neither resolve the factorization freedom in Section 3 nor enforce the expected gradient structure of the driving force. For smooth fields, the EnVarA force $\mathbf{f} = -\nabla(\delta E/\delta\rho)$ is curl-free. In two dimensions, we encourage this property with

$$\mathcal{L}_{\text{curl}} = \frac{\left\langle (\partial_x \widehat{f}_y - \partial_y \widehat{f}_x)^2 \right\rangle}{\left\langle (\partial_x \widehat{f}_y)^2 + (\partial_y \widehat{f}_x)^2 \right\rangle + \varepsilon}. \tag{7}$$

Here, $\widehat{f}_x$ and $\widehat{f}_y$ are the predicted driving-force components, and $\langle\cdot\rangle$ denotes the arithmetic mean over samples and spatial grid points in a training batch. The penalty is applied to the driving force because spatially varying mobility may produce a transport velocity with nonzero curl; it does not guarantee an underlying energy functional.

The relative reaction rate is supervised by the known source law:

$$\mathcal{L}_r^{\text{law}} = \ell(\widehat{r}, r^{\text{law}}), \qquad r^{\text{law}} = \frac{S(\rho)}{\rho}, \tag{8}$$

where S is the prescribed local reaction source. Because S can be evaluated directly on any input density field, all reported reaction modules use known-law supervision. Extracting relative reaction rates from trajectory data would instead require separating reaction from transport. The complete velocity-data objective is

$$\mathcal{L}_{\mathrm{vel}} = \mathcal{L}_u^{\mathrm{data}} + \lambda_{\mathrm{curl}}\mathcal{L}_{\mathrm{curl}} + \mathcal{L}_r^{\mathrm{law}}, \tag{9}$$

with $\lambda_{\mathrm{curl}} = 0.01$. The reaction term is omitted for nonreactive systems, and the curl term is omitted in one dimension.

Known-law supervision. When the mobility and thermodynamic driving-force laws are known, their evaluations on the current density provide direct constitutive targets:

$$\mathcal{L}_\xi^{\mathrm{law}} = \ell(\widehat{\xi}, \xi^{\mathrm{law}}), \qquad \mathcal{L}_f^{\mathrm{law}} = \ell(\widehat{\mathbf{f}}, \mathbf{f}^{\mathrm{law}}). \tag{10}$$

These losses constrain the two constitutive factors individually, removing the factorization ambiguity inherent in supervising only their product $\widehat{\mathbf{u}} = \widehat{\xi}\widehat{\mathbf{f}}$. The force target also supplies the expected gradient structure. Using the same reaction loss $\mathcal{L}_r^{\mathrm{law}}$, the complete known-law objective is

$$\mathcal{L}_{\mathrm{law}} = \mathcal{L}_\xi^{\mathrm{law}} + \mathcal{L}_f^{\mathrm{law}} + \mathcal{L}_r^{\mathrm{law}}. \tag{11}$$

With partial constitutive knowledge, known responses can remain analytical while only the unknown constitutive modules are learned. When known-law supervision is applied to independently sampled density fields, the framework enables **trajectory-free constitutive learning** without generating PDE solution trajectories.

5 Experiments

We evaluate NCL-MCT Solver along three complementary questions. First, *cross-system applicability* asks whether a common constitutive interface covers systems with different transport mechanisms, reactions, and species counts. Second, *generalization beyond training conditions* evaluates reuse of trained constitutive modules on unseen initial-condition families and an extended time horizon. Third, *trajectory-free constitutive learning* tests whether known-law supervision can train constitutive modules directly on independently sampled density fields without generating solution trajectories. PDE-specific constitutive modules are trained separately for each system while sharing the same constitutive interface and MCT integrator. Unless otherwise specified, training uses a 128×128 grid and ten snapshots from each of 100 trajectories, yielding $1{,}000$ training samples. For the trajectory-based benchmarks, we compare against methods with distinct learning targets: F-FNO (Tran et al., 2023) learns a state map, CFO (Hou et al., 2026) learns a temporal derivative, and DOOL (Chang et al., 2026) learns dissipative fluxes through an Onsager variational objective. All methods use the same sampled training states where applicable, and the baselines are trained for $100{,}000$ updates. We report mean $\pm$ standard deviation over ten test trajectories of the space-time relative L^2 rollout error E_{roll} and the maximum per-time relative L^2 error $E_{\max}$. For the extended time horizon, we additionally report $\Delta E = \bar{E}(T) - \bar{E}(T_{\mathrm{train}})$, where $\bar{E}(t)$ is the mean per-time relative L^2 error across test trajectories. Vel. and Law denote velocity-data and known-law supervision, respectively. Appendices D-I provide experimental details, additional rollouts, generalization results, and ablations.

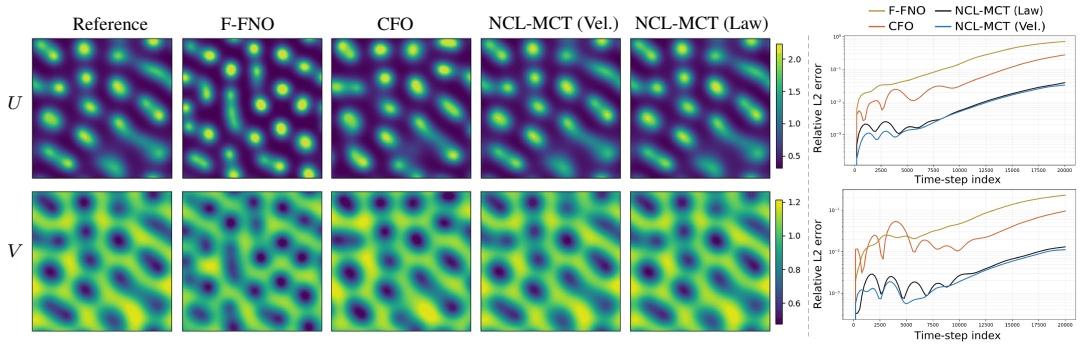
5.1 Applicability across Reaction-Diffusion Systems

Generalized diffusion. We first consider linear diffusion, Cahn-Hilliard (CH), and porous medium (PM), representing spatial smoothing, phase separation, and degenerate diffusion with moving low-density fronts. The same constitutive interface and MCT integrator are used across all three systems. On CH, both NCL-MCT Solver configurations achieve roughly $8\times$ lower rollout errors than the strongest evaluated baseline (Table 1(a)). On linear diffusion, F-FNO is more accurate; HC-PINN (Hao et al., 2026) shows a similar advantage (Appendix I.3). On PM, CFO is approximately five to six times more accurate than both NCL-MCT Solver configurations, which remain five to seven times more accurate than F-FNO. PM also exposes the positive-density limitation in Section 3: compact support creates vanishing-density regions where the factorization $\rho = MI$ requires additional care near the free boundary.

Table 1: Applicability across generalized reaction-diffusion systems. E_{roll} and $E_{\max}$ denote the space-time and maximum per-time relative L^2 errors, reported as mean $\pm$ standard deviation over ten test trajectories; N_T is the rollout length. Vel./Law denote velocity-data and known-law supervision, respectively. Red/orange mark the best/second-best mean per row. Gray* denotes F-FNO results computed from the finite trajectories only; $-$ denotes not evaluated.

System	N_T	Metric	F-FNO (Tran et al., 2023)	CFO (Hou et al., 2026)	DOOL (Chang et al., 2026)	NCL-MCT (Vel.) (Ours)	NCL-MCT (Law) (Ours)
(a) Generalized diffusion							
Linear Diffusion	600	E_{roll}	$(6.52 \pm 1.65) \times 10^{-4}$	$(9.90 \pm 4.91) \times 10^{-3}$	$(5.65 \pm 1.48) \times 10^{-2}$	$(9.87 \pm 2.61) \times 10^{-4}$	$(1.57 \pm 0.46) \times 10^{-3}$
		$E_{\max}$	$(9.76 \pm 2.78) \times 10^{-4}$	$(1.19 \pm 0.60) \times 10^{-2}$	$(7.36 \pm 1.48) \times 10^{-2}$	$(1.18 \pm 0.29) \times 10^{-3}$	$(1.82 \pm 0.59) \times 10^{-3}$
CH	10,000	E_{roll}	$(1.45 \pm 0.09) \times 10^{-1}$	$(6.16 \pm 1.47) \times 10^{-2}$	$(3.32 \pm 0.53) \times 10^{-1}$	$(7.50 \pm 1.92) \times 10^{-3}$	$(7.57 \pm 1.28) \times 10^{-3}$
		$E_{\max}$	$(1.87 \pm 0.14) \times 10^{-1}$	$(8.78 \pm 2.35) \times 10^{-2}$	$(4.71 \pm 0.74) \times 10^{-1}$	$(1.04 \pm 0.30) \times 10^{-2}$	$(1.05 \pm 0.25) \times 10^{-2}$
PM	2,000	E_{roll}	$(7.87 \pm 4.00) \times 10^{-2}$	$(2.68 \pm 0.61) \times 10^{-3}$	$(7.30 \pm 2.61) \times 10^{-1}$	$(1.67 \pm 0.17) \times 10^{-2}$	$(1.66 \pm 0.25) \times 10^{-2}$
		$E_{\max}$	$(1.46 \pm 0.83) \times 10^{-1}$	$(4.47 \pm 1.19) \times 10^{-3}$	$(3.19 \pm 1.03) \times 10^{0}$	$(2.09 \pm 0.20) \times 10^{-2}$	$(2.20 \pm 0.34) \times 10^{-2}$
(b) Generalized reaction-diffusion							
Fisher-KPP	500	E_{roll}	$(5.35 \pm 1.32) \times 10^{-4}$	$(5.90 \pm 2.79) \times 10^{-3}$	–	$(5.01 \pm 1.39) \times 10^{-4}$	$(6.58 \pm 0.54) \times 10^{-4}$
		$E_{\max}$	$(7.50 \pm 1.67) \times 10^{-4}$	$(7.83 \pm 3.73) \times 10^{-3}$	–	$(5.96 \pm 1.96) \times 10^{-4}$	$(8.90 \pm 0.79) \times 10^{-4}$
Reactive CH	10,000	E_{roll}	$(1.29 \pm 0.23) \times 10^{-1}$	$(5.64 \pm 1.11) \times 10^{-2}$	–	$(8.86 \pm 2.21) \times 10^{-3}$	$(9.62 \pm 2.13) \times 10^{-3}$
		$E_{\max}$	$(1.63 \pm 0.34) \times 10^{-1}$	$(7.95 \pm 2.06) \times 10^{-2}$	–	$(1.40 \pm 0.42) \times 10^{-2}$	$(1.37 \pm 0.38) \times 10^{-2}$
Reactive PM	1,000	E_{roll}	$(3.96 \pm 3.77) \times 10^{-2}$	$(7.78 \pm 3.98) \times 10^{-3}$	–	$(3.84 \pm 0.79) \times 10^{-2}$	$(1.56 \pm 0.57) \times 10^{-2}$
		$E_{\max}$	$(4.81 \pm 4.01) \times 10^{-2}$	$(1.52 \pm 1.09) \times 10^{-2}$	–	$(5.32 \pm 1.16) \times 10^{-2}$	$(1.89 \pm 0.55) \times 10^{-2}$
(c) Multispecies reaction-diffusion							
Schnakenberg (U)	20,000	E_{roll}	$(2.73 \pm 0.36) \times 10^{-1}$	$(1.28 \pm 0.16) \times 10^{-1}$	–	$(2.52 \pm 1.77) \times 10^{-2}$	$(2.38 \pm 1.26) \times 10^{-2}$
		$E_{\max}$	$(7.06 \pm 0.63) \times 10^{-1}$	$(2.87 \pm 0.49) \times 10^{-1}$	–	$(8.28 \pm 8.55) \times 10^{-2}$	$(7.36 \pm 6.15) \times 10^{-2}$
Schnakenberg (V)		E_{roll}	$(1.18 \pm 0.18) \times 10^{-1}$	$(4.39 \pm 0.59) \times 10^{-2}$	–	$(6.74 \pm 3.37) \times 10^{-3}$	$(6.57 \pm 2.27) \times 10^{-3}$
		$E_{\max}$	$(2.33 \pm 0.26) \times 10^{-1}$	$(1.04 \pm 0.18) \times 10^{-1}$	–	$(2.19 \pm 1.93) \times 10^{-2}$	$(1.97 \pm 1.30) \times 10^{-2}$

*Finite trajectory counts for gray cells: CH 9/10, Reactive CH 8/10, Schnakenberg 9/10.

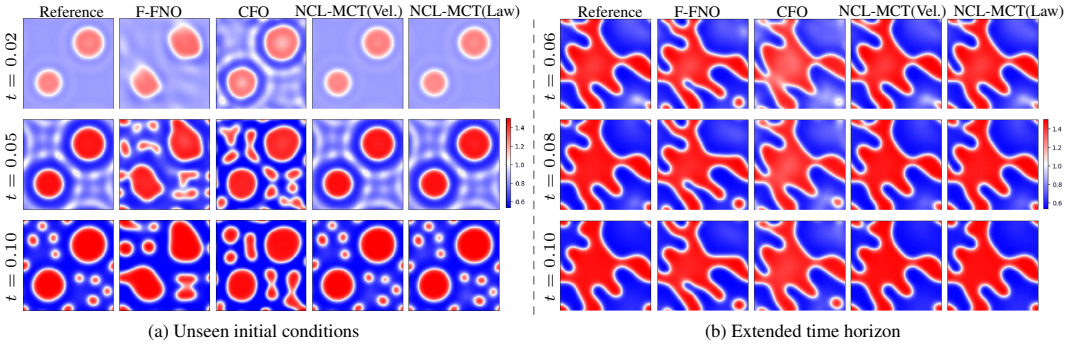


(a) Results at $t = 1$ (step 20,000) (b) Per-time error

Figure 2: Schnakenberg comparison for both species. (a) Concentrations U (top) and V (bottom) at $t = 1$ for a test trajectory on which all methods remain finite. (b) Per-time relative L^2 errors over the 20,000-step rollout.

Generalized reaction-diffusion. Reaction is incorporated by adding a pointwise relative-reaction-rate module to the same transport interface and MCT integrator. The DOOL formulation evaluated here is transport-only and is therefore omitted from these reactive benchmarks. On Fisher-KPP, reactive CH, and reactive PM (Table 1(b)), both NCL-MCT Solver configurations achieve roughly $6\times$ lower errors on reactive CH than the strongest evaluated baseline. Vel. attains the lowest mean error on Fisher-KPP, with F-FNO within one standard deviation, whereas CFO is more accurate on reactive PM. The PM ranking follows the nonreactive case, suggesting that degenerate transport remains the dominant difficulty. Known-law supervision reduces the reactive-PM error by roughly 2.5-3$\times$ relative to Vel.; the two supervision modes are otherwise comparable.

Multispecies reaction-diffusion. For two species, each species uses its own transport module, while the reaction module takes both local concentrations as input and is trained jointly with the transport modules. On Schnakenberg, both NCL-MCT Solver configurations remain stable over the full 20,000-step rollout, with mean errors three to seven times below CFO and about an order of magnitude below F-FNO across both species and metrics (Table 1(c)). F-FNO becomes nonfinite



(a) Unseen initial conditions (b) Extended time horizon

Figure 3: Generalization beyond training conditions on reactive Cahn–Hilliard. (a) From an unseen two-nucleus initial condition, F-FNO distorts the large domains and loses secondary droplets, while CFO merges them into elongated structures; both NCL-MCT Solver configurations better preserve the reference morphology. (b) Models trained on the first 60% of the interval are rolled out over the full horizon. Beyond the training window, CFO loses interior contrast and smooths narrow channels, while both NCL-MCT Solver configurations better preserve the interfaces.

Table 2: Generalization beyond training conditions, aggregated across systems: (a) unseen initial conditions (five systems, one test trajectory per system) and (b) extended time horizon (four systems, ten test trajectories per system). Entries are averages of the per-system E_{roll}, $E_{\max}$, or ΔE values. Per-system results are reported in Appendix G. DOOL is omitted because it covers too few systems for aggregation. Red/orange indicate the best/second-best comparable result per row.

	Metric	F-FNO (Tran et al., 2023)	CFO (Hou et al., 2026)	NCL-MCT (Vel.) (Ours)	NCL-MCT (Law) (Ours)
Unseen initial conditions (5 systems)	E_{roll}	$1.81 \times 10^{-1*}$	1.69×10^{-1}	2.99×10^{-2}	2.86×10^{-2}
	$E_{\max}$	$2.97 \times 10^{-1*}$	2.38×10^{-1}	4.39×10^{-2}	4.44×10^{-2}
Extended time horizon (4 systems)	E_{roll}	$6.75 \times 10^{-2\dagger}$	6.89×10^{-2}	5.85×10^{-3}	5.49×10^{-3}
	ΔE	$1.25 \times 10^{-2\dagger}$	4.82×10^{-3}	1.97×10^{-3}	2.76×10^{-3}

*F-FNO is nonfinite on CH and Fisher-KPP; the reported average uses only the 3/5 finite systems and is not directly comparable.
†For reactive CH, F-FNO statistics use the 9/10 trajectories that remain finite.

on one trajectory near the end of the rollout, while Law has the lower mean error for both species. Figure 2 shows the error histories: F-FNO errors grow steadily, CFO accumulates larger final-state discrepancies, and both NCL-MCT Solver configurations remain close to the reference.

5.2 Generalization beyond Training Conditions

Unseen initial conditions. Because the constitutive modules are evaluated from the current density rather than explicitly from the initial condition, changing the initial-condition family changes the density fields encountered during rollout. Keeping the trained models and PDE parameters of Section 5.1, we replace only the initial-condition family with unseen structured fields whose amplitudes remain close to the training range, making the shift primarily spatial (Appendix G.1). Across the five systems, both NCL-MCT Solver configurations achieve nearly $6\times$ lower average E_{roll} than CFO, the strongest baseline that remains finite across all five systems, with per-system margins from $2.4\times$ to $13\times$ (Table 2(a)). Figure 3(a) illustrates the difference on reactive CH. Linear diffusion shows the largest change in ranking: F-FNO is the most accurate method in distribution, but its error increases by roughly $260\times$ under the shift, compared with 24-41$\times$ for NCL-MCT Solver, and it becomes nonfinite on CH and Fisher-KPP. The relative degradation is not uniformly smaller for NCL-MCT Solver; on Fisher-KPP, its error also increases substantially but remains lower because its in-distribution error starts from a much smaller level.

Extended time horizon. We train on the first 60% of each reference trajectory and evaluate over the full interval, so the final 40% lies beyond the training horizon (Appendix G.2). Averaged across the four systems, both NCL-MCT Solver configurations achieve full-window rollout errors about an order of magnitude below the evaluated baselines. Their errors change by only 0.90-$1.46\times$ relative to the corresponding models trained on the full time interval (Table 2(b)). By comparison,

Table 3: Trajectory-free learning from independently sampled density fields. Both methods use the same unlabeled density fields and are evaluated on ten held-out initial conditions. E_{roll} and $E_{\max}$ are relative L^2 errors (mean $\pm$ standard deviation), and N_T is the rollout length. Law denotes known-law supervision on independently sampled states. Red marks the lower mean per row.

System	N_T	Metric	DOOL (Chang et al., 2026)	NCL-MCT (Law) (Ours)
Linear Diffusion (1D, DOOL setting)	4,000	E_{roll}	$(1.53 \pm 1.07) \times 10^{-3}$	$(3.13 \pm 1.69) \times 10^{-4}$
		$E_{\max}$	$(2.39 \pm 1.67) \times 10^{-3}$	$(4.05 \pm 2.32) \times 10^{-4}$
CH (DOOL setting)	2,000	E_{roll}	$(7.46 \pm 2.90) \times 10^{-3}$	$(6.40 \pm 6.66) \times 10^{-3}$
		$E_{\max}$	$(1.17 \pm 0.37) \times 10^{-2}$	$(8.71 \pm 8.33) \times 10^{-3}$

CFO errors increase by about $1.6\times$ on the two Cahn-Hilliard systems and $5.2\times$ on linear diffusion and Fisher-KPP. F-FNO remains strongest on linear diffusion but about an order of magnitude less accurate than NCL-MCT Solver on the two Cahn-Hilliard systems. Both NCL-MCT Solver configurations also show the smallest ΔE on linear diffusion, CH, and Fisher-KPP. Only on reactive CH does CFO show smaller growth, despite an error at T_{train} that is twelve times larger. Figure 3(b) shows the corresponding reactive-CH fields.

5.3 Trajectory-Free Constitutive Learning

Although the experiments above use trajectory snapshots as training inputs, known-law supervision does not require them: the targets ξ^{law} and $\mathbf{f}^{\mathrm{law}}$ in Equation 10 depend only on the current density field and can be evaluated on independently sampled density fields. We test this trajectory-free setting on one-dimensional linear diffusion and two-dimensional Cahn-Hilliard using the sampled training distributions of DOOL (Chang et al., 2026). For each system, DOOL and NCL-MCT Solver (Law) use the same sampled density fields; solution trajectories are reserved for held-out rollout evaluation (Appendix H). NCL-MCT Solver (Law) achieves lower mean errors on both systems (Table 3). On linear diffusion, E_{roll} and $E_{\max}$ are lower than those of DOOL by factors of 4.9 and 5.9, respectively. On Cahn-Hilliard, the corresponding factors are 1.2 and 1.4, although the two methods remain within one standard deviation. These results show that, on these benchmarks, constitutive modules trained directly on independently sampled density fields can support $2{,}000$-$4{,}000$-step rollouts without solution trajectories during training. Because the methods use different time integrators, the quantitative comparison reflects the complete solvers rather than the learning objectives alone.

6 Conclusion

We presented NCL-MCT Solver, which places neural learning in PDE-specific constitutive responses while retaining a shared MCT integrator for density evolution. Transport is represented through mobility and thermodynamic driving force, and reaction through relative reaction rates. Across seven generalized diffusion and reaction-diffusion systems, the shared constitutive interface achieves relative rollout L^2 errors of 10^{-4} to 10^{-2} and is competitive with or better than the evaluated baselines on most systems. Tests on unseen initial-condition families and an extended time horizon further assess reuse of the learned constitutive modules without retraining. With known-law supervision on independently sampled density fields, a separate experiment demonstrates trajectory-free constitutive learning without generating solution trajectories. Together, these results support learning PDE-specific constitutive responses, rather than the full solution evolution, as an effective target for a shared neural PDE framework.

The current formulation assumes positive densities, with porous-medium systems exposing its limitation near vanishing density. Future work will extend the factor evolution to such regimes, investigate constitutive identification under partially known laws, broaden the treatment of boundary conditions (Liu et al., 2024c), incorporate parameter-conditioned constitutive models for inverse problems (Cho & Son, 2025), and couple the framework to multiphysics models involving pressure, flow, and tissue mechanics (Lu et al., 2019; 2022).

Disclosure of AI Use

The authors developed the research ideas and proposed method, performed the mathematical verification, and designed and conducted all experiments. AI tools assisted with identifying relevant literature, exploring existing neural network architectures and numerical methods, implementing and debugging code, drafting portions of the manuscript, and improving its wording and clarity. All code was reviewed and validated by the authors, who take full responsibility for the content and conclusions of this work.

References

E. M. Arruda and M. C. Boyce. A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials. *Journal of the Mechanics and Physics of Solids*, 41(2):389–412, 1993. doi: 10.1016/0022-5096(93)90013-6.

Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019. doi: 10.1073/pnas.1814058116.

John W. Cahn and John E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of Chemical Physics*, 28(2):258–267, 1958. doi: 10.1063/1.1744102.

Zhipeng Chang, Zhenye Wen, and Xiaofei Zhao. Unsupervised operator learning approach for dissipative equations via Onsager principle. *SIAM Journal on Scientific Computing*, 48(4):C1060–C1085, 2026. doi: 10.1137/25M1786763.

Sung Woong Cho and Hwijae Son. Physics-informed deep inverse operator networks for solving PDE inverse problems. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0FxnSZJPmh.

Alexander E. Cohen, Samuel Degnan-Morgenstern, Simon Daubner, Jörn Dunkel, and Martin Z. Bazant. Differentiable learning and control of free-energy-driven pattern dynamics. *Physical Review Research*, 8:023344, 2026. doi: 10.1103/b8kc-vpwq.

D. C. Drucker and W. Prager. Soil mechanics and plastic analysis or limit design. *Quarterly of Applied Mathematics*, 10(2):157–165, 1952. doi: 10.1090/qam/48291.

Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. In *Handbook of Numerical Analysis*, volume 7, pp. 713–1018. Elsevier, 2000. doi: 10.1016/S1570-8659(00)07005-8.

Sarah M. Groves, Min-Jhe Lu, Astrid Catalina Alvarez-Yela, Monserrat Gerardo-Ramírez, P. Todd Stukenberg, John S. Lowengrub, and Kevin A. Janes. Cell-specific Cahn–Hilliard models predict condensed fates of the chromosomal passenger complex. *PLOS Computational Biology*, 22(8):e1014568, 2026. doi: 10.1371/journal.pcbi.1014568.

Derek Hansen, Danielle C. Maddix, Shima Alizadeh, Gaurav Gupta, and Michael W. Mahoney. Learning physical models that can respect conservation laws. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12469–12510. PMLR, 2023.

Baoli Hao, Chun Liu, Ulisses Braga-Neto, Lifan Wang, and Ming Zhong. Stability in training PINNs for stiff PDEs: Why initial conditions matter. *Foundations of Data Science*, 2026. doi: 10.3934/fods.2026016. URL https://doi.org/10.3934/fods.2026016. Early Access.

Masanobu Horie and Naoto Mitsume. Graph neural PDE solvers with conservation and similarity-equivariance. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18785–18814. PMLR, 2024.

Xianglong Hou, Xinquan Huang, and Paris Perdikaris. CFO: Learning continuous-time PDE dynamics via flow-matched neural operators. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IQhaeSzyup.

Ziqing Hu, Chun Liu, Yiwei Wang, and Zhiliang Xu. Energetic variational neural network discretizations of gradient flows. *SIAM Journal on Scientific Computing*, 46(4):A2528–A2556, 2024. doi: 10.1137/22M1529427.

Shenglin Huang, Zequn He, Bryan Chem, and Celia Reina. Variational Onsager neural networks (VONNs): A thermodynamics-based variational learning strategy for non-equilibrium PDEs. *Journal of the Mechanics and Physics of Solids*, 163:104856, 2022. doi: 10.1016/j.jmps.2022.104856.

Shenglin Huang, Zequn He, Nicolas Dirr, Johannes Zimmer, and Celia Reina. Statistical-physics-informed neural networks (Stat-PINNs): A machine learning strategy for coarse-graining dissipative dynamics. *Journal of the Mechanics and Physics of Solids*, 194:105908, 2025. doi: 10.1016/j.jmps.2024.105908.

Yunkyong Hyon, Do Young Kwak, and Chun Liu. Energetic variational approach in complex fluids: Maximum dissipation principle. *Discrete and Continuous Dynamical Systems*, 26(4):1291–1304, 2010. doi: 10.3934/dcds.2010.26.1291.

Matthias Karlbauer, Timothy Praditia, Sebastian Otte, Sergey Oladyshkin, Wolfgang Nowak, and Martin V. Butz. Composing partial differential equations with physics-aware neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10773–10801. PMLR, 2022.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. doi: 10.1073/pnas.2101784118.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=c8P9NQVtmnO`.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506.

Ning Liu, Yiming Fan, Xianyi Zeng, Milan Klöwer, Lu Zhang, and Yue Yu. Harnessing the power of neural operators with automatically encoded conservation laws. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30965–30997. PMLR, 2024a.

Pengwei Liu, Zhongkai Hao, Xingyu Ren, Hangjie Yuan, Jiayang Ren, and Dong Ni. PAPM: A physics-aware proxy model for process systems. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31080–31105. PMLR, 2024b.

Ziyuan Liu, Haifeng Wang, Hong Zhang, Kaijun Bao, Xu Qian, and Songhe Song. Render unto numerics: Orthogonal polynomial neural operator for PDEs with nonperiodic boundary conditions. *SIAM Journal on Scientific Computing*, 46(4):C323–C348, 2024c. doi: 10.1137/23M1556320.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Min-Jhe Lu, Chun Liu, and Shuwang Li. Nonlinear simulation of an elastic tumor-host interface. *Computational and Mathematical Biophysics*, 7(1):25–47, 2019. doi: 10.1515/cmb-2019-0003.

Min-Jhe Lu, Wenrui Hao, Chun Liu, John Lowengrub, and Shuwang Li. Nonlinear simulation of vascular tumor growth with chemotaxis and the control of necrosis. *Journal of Computational Physics*, 459:111153, 2022. doi: 10.1016/j.jcp.2022.111153.

Yubin Lu, Xiaofan Li, Chun Liu, Qi Tang, and Yiwei Wang. Learning generalized diffusions using an energetic variational approach. *Communications in Computational Physics*, 40(2):389–412, 2026a. doi: 10.4208/cicp.OA-2025-0141.

Yubin Lu, Xiaofan Li, Chun Liu, Qi Tang, and Yiwei Wang. Structure-aware variational learning of a class of generalized diffusions. *Physica D: Nonlinear Phenomena*, 498:135404, 2026b. doi: 10.1016/j.physd.2026.135404.

Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B. Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. Learning neural constitutive laws from motion observations for generalizable PDE dynamics. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23279–23300. PMLR, 2023.

Luis Mandl, Dibyajyoti Nayak, Tim Ricken, and Somdatta Goswami. Physics-informed time-integrated DeepONet: Temporal tangent space operator learning for high-accuracy inference. *Computer Methods in Applied Mechanics and Engineering*, 455:118917, 2026. doi: 10.1016/j.cma.2026.118917.

Himangi Mittal, Peiye Zhuang, Hsin-Ying Lee, and Shubham Tulsiani. UniPhy: Learning a unified constitutive model for inverse physics simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16208–16218, 2025. doi: 10.1109/CVPR52734.2025.01511.

Dibyajyoti Nayak and Somdatta Goswami. TI-DeepONet: Learnable time integration for stable long-term extrapolation. *Computer Methods in Applied Mechanics and Engineering*, 456:118960, 2026. doi: 10.1016/j.cma.2026.118960.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Chengping Rao, Pu Ren, Qi Wang, Oral Buyukozturk, Hao Sun, and Yang Liu. Encoding physics to learn reaction–diffusion processes. *Nature Machine Intelligence*, 5(7):765–779, 2023. doi: 10.1038/s42256-023-00685-7.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel De Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 77187–77200. Curran Associates, Inc., 2023. doi: 10.52202/075280-3376.

Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models: A review. *Monthly Weather Review*, 119(9):2206–2223, 1991. doi: 10.1175/1520-0493(1991)119⟨2206:SLISFA⟩2.0.CO;2.

Yusuke Tanaka, Takaharu Yaguchi, Tomoharu Iwata, and Naonori Ueda. Energy-consistent neural operators for Hamiltonian and dissipative partial differential equations. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1882–1890. PMLR, 2025.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=tmIiMPl4IPa`.

L. R. G. Treloar. The elasticity of a network of long-chain molecules. I. *Transactions of the Faraday Society*, 39:36–41, 1943. doi: 10.1039/TF9433900036.

Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952. doi: 10.1098/rstb.1952.0012.

Juan Luis Vázquez. *The Porous Medium Equation: Mathematical Theory*. Oxford Mathematical Monographs. Oxford University Press, 2006. doi: 10.1093/acprof:oso/9780198569039.001.0001.

Yiwei Wang, Chun Liu, Pei Liu, and Bob Eisenberg. Field theory of reaction-diffusion: Law of mass action with an energetic variational approach. *Physical Review E*, 102(6):062147, 2020. doi: 10.1103/PhysRevE.102.062147.

Xueguang Xie, Shu Yan, Shiwen Jia, Siyu Yang, Aimin Hao, Yang Gao, and Peng Yu. PC-NCLaws: Physics-embedded conditional neural constitutive laws for elastoplastic materials. In *Pacific Graphics 2025 Conference Papers*, 2025. doi: 10.2312/pg.20251266.