IADR: Interface-Augmented Neural Operator for Phase-Field Mean-Curvature Flow

Qinyi Zhang[*]
School of Mathematics
Sichuan University, Chengdu, China
2024322010008@stu.scu.edu.cn

Duanyu Feng[*]
College of Computer Science
Sichuan University, Chengdu, China
fengduanyuscu@stu.scu.edu.cn

Yangshuai Wang[†]
Department of Mathematics
National University of Singapore
yswang@nus.edu.sg

Hao Wang[†]
School of Mathematics
Sichuan University, Chengdu, China
wangh@scu.edu.cn

Abstract

The Allen–Cahn equation is one of the standard phase-field models for mean-curvature flow (MCF), parameterised by the diffuse-interface width ε. Practical simulations span a range of ε, so a single model covering this range is needed. PDE-specific neural operators are accurate but must be retrained at every ε; general neural operators cover ε in one model but lose the phase-field structure of the target. We introduce Interface-Augmented Diffusion-Reaction (IADR), a neural operator that addresses both shortcomings: a target network with a diffusion-and-reaction (DR) structure preserves the phase-field structure of the Allen–Cahn equation, while a hypernetwork that maps ε to the weights of this target network amortises the dependence on ε across the range. A single IADR matches the per-ε specialists on in-distribution data within the prescribed range, retains their cross-shape generalisation under both ε- and geometry-out-of-distribution shifts, and retains their compact DR target structure with millisecond-scale inference. We further establish a conditional K-step rollout error bound under consistency and stability assumptions. Code is available at `https://github.com/ArcueidCroft/IADR`.

1 Introduction

Mean-curvature flow (MCF) is a geometric evolution law in which every point of an interface moves along its inward normal at a speed proportional to the local mean curvature, and it underlies a wide range of applications including multi-phase fluid flow and materials microstructure coarsening [2, 4]. Among the numerical approaches for MCF, the phase-field approach replaces the sharp interface by a smooth transition layer of width ε, described by a scalar order parameter u. The L^2-gradient flow of the corresponding Cahn–Hilliard energy then yields the Allen–Cahn equation $\partial_t u = \Delta u - \frac{1}{\varepsilon^2} W'(u)$, the standard phase-field model of MCF and the focus of this paper. Its $\varepsilon \to 0$ limit is rigorously the sharp-interface MCF [13, 19], and decades of numerical analysis [5, 12] have made the Allen–Cahn equation a canonical workhorse of phase-field simulation. The interface width ε is a physically meaningful tunable parameter [12]: smaller ε sharpens the interface and tightens the sharp-interface limit but demands finer spatial resolution and shorter time steps, while larger ε smooths the dynamics and reduces compute at the cost of geometric fidelity. Practical simulations span a range of ε rather

[*]Equal contribution.
[†]Co-corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

than a single fixed value, so accurate and efficient simulation across this range is a prerequisite for interface modelling.

Neural-network operators have been studied as a faster differentiable surrogate for the Allen–Cahn equation; two categories are most relevant. *General operator-learning frameworks* (e.g., FNO, DeepONet) approximate the solution map of a parametric PDE family by a function-to-function network and condition on ε through an input channel. *PDE-specific operator learning* replaces an operator-splitting scheme such as the diffusion and reaction (DR) splitting by a trainable network with the same forward topology (a linear diffusion solve followed by a pointwise nonlinear reaction update), preserving the phase-field structure and attaining sub-grid accuracy with only a few hundred parameters per ε, but training a separate network for each ε. Both achieve useful accuracy on the Allen–Cahn equation at a fixed ε. General operators cover ε in one model but, with no phase-field structure in the target, their behaviour on the Allen–Cahn evolution remains uncertain; PDE-specific operators retain the structure but do not share parameters across ε.

Inspired by DISCO's weight-injection design [33], we close the ε-generalisation gap with a single neural operator specialised to the Allen–Cahn equation, in which one part of the network encodes the phase-field structure and another part encodes the dependence on ε. Concretely, the **target network** retains the diffusion-and-reaction factorisation of classical operator-splitting schemes (a spatial diffusion convolution composed with a pointwise nonlinear reaction), so that the phase-field structure shared by every $\varepsilon \in E$ is hard-coded into the architecture. A separate **hypernetwork** reads the scalar ε and produces the weights of this target network, so the dependence of the operator on ε is concentrated in a single small MLP and shared across the range. We call the resulting operator Interface-Augmented Diffusion-Reaction (IADR). On the theoretical side, we establish a conditional K-step rollout error bound for this hypernetwork-conditioned operator, decomposing the long-horizon error into a one-step consistency term and a slice-wise stability term, and identifying the regimes under which the rollout error stays uniformly bounded.

We empirically validate the ε-generalisation of IADR on a suite of in-distribution and out-of-distribution evaluations of the Allen–Cahn equation. The in-distribution evaluation tests IADR on the same circular geometry and the same five training ε values used at training time. The out-of-distribution evaluations test the model on unseen interface widths (both interpolating within the training interval E and extrapolating beyond it), on unseen non-circular geometries, and on the combination of the two. A single IADR matches the per-ε specialists on the in-distribution evaluation within the prescribed range, retains their cross-shape generalisation across all out-of-distribution shifts, and retains their compact diffusion-and-reaction (DR) target structure with millisecond-scale inference while remaining substantially faster than every general operator target.

In summary, our main contributions are (i) IADR, a neural operator for the Allen–Cahn equation: a target network with a diffusion-and-reaction structure, whose weights are produced by a hypernetwork from the interface-width parameter ε; (ii) a conditional K-step rollout error bound for this hypernetwork-conditioned operator, identifying a strict-contraction regime in which the long-horizon error stays uniformly bounded and an error-growth regime with an explicit bound; (iii) empirical validation showing that a single IADR matches the per-ε specialists on in-distribution data within the prescribed range, retains their cross-shape generalisation on out-of-distribution geometries and unseen interface widths, and retains their compact DR target structure with millisecond-scale inference while remaining substantially faster than every general operator target.

2 Related Work

Numerical schemes for mean-curvature flow. The Allen–Cahn equation has been studied as the phase-field model of mean-curvature flow for decades, and a mature line of semi-implicit time-stepping schemes is available. Convex splitting [14], the semi-implicit Fourier-spectral method [5], and the diffusion and reaction (DR) splitting [2] (the latter falling within the Lie–Trotter / Strang splitting family for gradient-flow PDEs [31]) attain sub-grid accuracy with well understood stability and convergence properties; the systematic numerical analysis of mean-curvature flow and related geometric PDEs is surveyed in [11], with the underlying sharp-interface theory grounded in the level-set formulation [13] and the Allen–Cahn convergence results [19]. The discretisations of these schemes, however, depend on ε explicitly and must be re-instantiated for every interface width of interest, and the schemes are not differentiable end-to-end. These two limitations motivate the

neural-network operators discussed next, which restore differentiability; the per-ε limitation, however, persists in the neural-operator literature and motivates the present paper.

Operator learning. A second body of work learns the solution map of a PDE by a parametric neural network, within the broader physics-informed machine-learning paradigm [21, 9]. Two threads are most relevant for the Allen–Cahn equation. *General operator-learning frameworks*, consolidated by Kovachki et al. [24], learn solution maps that are not specific to any single PDE: families such as the Fourier Neural Operator [27], DeepONet [29], and their graph and multipole variants [25, 26] cast the solution map as a function-to-function network with the PDE parameter concatenated to the input field, and physics-informed extensions such as PINO [28] layer a residual loss on top of this input-level conditioning, echoing the PINN framework [34]. Applied to the Allen–Cahn equation, this thread covers the range of ε within a single network, but the target carries no phase-field structure, so accuracy on physics-aligned metrics typically lags the operator-splitting schemes of the previous paragraph by a wide margin. *PDE-specific operator learning* targets the Allen–Cahn equation directly: a common design is a black-box time-stepper that maps u_t to u_{t+1} on a fixed grid, often with a convolutional or U-Net backbone [36]; physics penalties on a level-set or phase-field representation [30] and learned corrections to a numerical scheme [22] extend this thread; the trainable DR splitting network of Bretin et al. [2] preserves the physical structure of the DR splitting but is bound to a single ε. Neither thread therefore delivers both ε-coverage and the phase-field structure of the target at the same time.

Meta-learning and hypernetworks for PDE solvers. A third line conditions a neural operator on a task descriptor through meta-learning or hypernetworks, with the aim of sharing weights across a family of PDE tasks. Meta-learning frameworks such as MAML [15] and HyperNetworks [16] have been adapted to PDE solvers in Meta-MgNet [7] and Hyper-LR-PINN [8], and subsequent work extends the line to foundation-scale PDE pre-training and task-adaptive conditioning [18, 17, 23, 1, 3]. The most direct realisation for our setting is DISCO [33]: a space-time Transformer encodes a T-frame trajectory, a hypernetwork maps the resulting code to a full weight vector, and the weights are injected into a frozen U-Net target. None of these methods is specialised to the Allen–Cahn equation: the target network is designed to transfer across heterogeneous PDE families and therefore carries no phase-field structure, so their accuracy relative to the operator-learning schemes of the previous paragraph on the Allen–Cahn equation has not been established. Among the methods above, only DISCO yields a one-step neural propagator that can be rolled out under the same autoregressive protocol as IADR; Meta-MgNet conditions a multigrid smoother for linear elliptic systems and Hyper-LR-PINN fits a coordinate MLP $u(x,t)$ per initial condition, neither yielding an operator over $\mathcal{X}$. We therefore take DISCO as our head-to-head meta-learning baseline. The present paper fills this gap by specialising the target network to the Allen–Cahn equation's DR splitting structure, retaining phase-field fidelity while sharing parameters across ε.

3 Method

3.1 Problem Formulation

We consider the Allen–Cahn equation as the phase-field model of mean-curvature flow on a domain $\Omega_x \subset \mathbb{R}^2$, where the order parameter $u : \Omega_x \times [0,T] \to \mathbb{R}$ evolves according to

$$\partial_t u = \Delta u - \tfrac{1}{\varepsilon^2} W'(u), \qquad W(s) = \tfrac{1}{2} s^2 (1-s)^2. \tag{1}$$

For an evolving sharp interface $\partial\Omega(t)$ with signed distance function $d(\cdot, \Omega(t))$, the solution u of (1) is known to concentrate in a layer of width ε near $\partial\Omega(t)$; its theoretical idealisation is the exact phase-field representation

$$\mathbf{q}_\varepsilon(\cdot,t) \;:=\; q\Big(\tfrac{d(\cdot,\Omega(t))}{\varepsilon}\Big), \qquad q(r) = \tfrac{1}{2}(1 - \tanh(r/2)). \tag{2}$$

This representation agrees with the solution u of (1) up to a Hausdorff error of order $O(\varepsilon^2 |\log \varepsilon|^2)$ on the diffuse interface (Appendix A); we therefore take $\mathbf{q}_\varepsilon$ as the learning target. Its exact δt-propagator is denoted by $\mathcal{T}_{\delta t,\varepsilon}(\mathbf{q}_\varepsilon(\cdot,t)) = \mathbf{q}_\varepsilon(\cdot, t+\delta t)$. As the structural template of our network we take the first-order DR splitting

$$\mathcal{S}_{\delta t,\varepsilon}(\mathbf{q}) \;=\; \rho_{1,\varepsilon}(K_1 * \mathbf{q}), \quad \rho_{1,\varepsilon}(s) = s - \tfrac{\delta t}{\varepsilon^2} W'(s), \quad \widehat{K}_1(\xi) = \frac{1}{1 + \delta t\, 4\pi^2 |\xi|^2}. \tag{3}$$

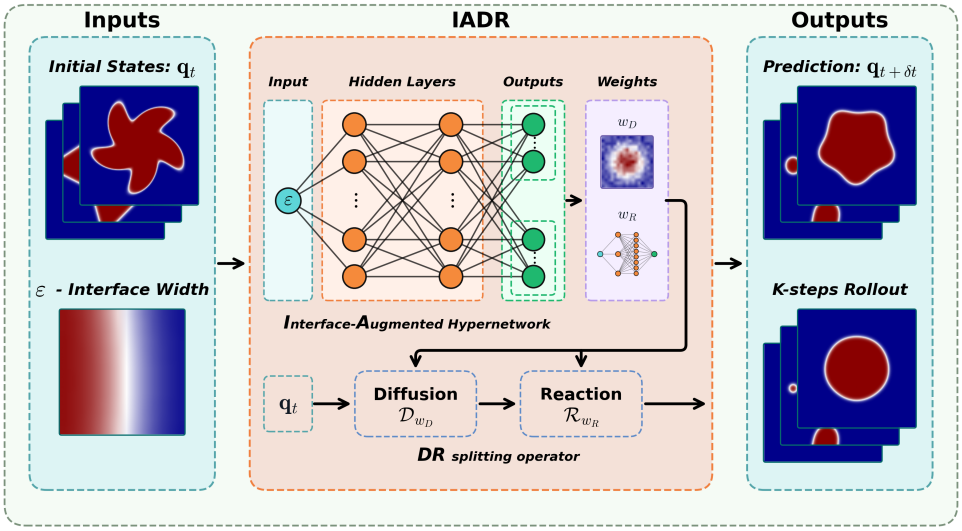


Figure 1: Overview of IADR: the hypernetwork $h_\vartheta(\varepsilon)$ emits the weights of the DR splitting network $\Phi = \mathcal{R}_{w_R} \circ \mathcal{D}_{w_D}$ (target network), which advances the phase field $\mathbf{q}_\varepsilon(\cdot, t)$ by one time step and can be applied autoregressively for K-step rollout.

The target network Φ of Section 3.2 mirrors $\mathcal{S}_{\delta t,\varepsilon}$ in topology: its weights are produced by the hypernetwork rather than derived analytically. The Γ-convergence background and the $O(\delta t^2)$ equivalence of the diffusion-then-reaction and reaction-then-diffusion orders are deferred to Appendices A and B.4.

Fix a compact computational domain $\Omega_x \subset \mathbb{R}^2$, time step $\delta t > 0$, and prescribed interface-width range $E = [\varepsilon_{\min}, \varepsilon_{\max}]$. Let $\mathcal{G}$ be a class of admissible initial sets and let $\mathcal{A}_\varepsilon = \{\mathbf{q}_\varepsilon(\cdot, t) : \Omega(0) \in \mathcal{G},\ t \in [0, T]\} \subset \mathcal{X}$, $\mathcal{A} = \{(\varepsilon, \mathbf{q}) : \varepsilon \in E,\ \mathbf{q} \in \mathcal{A}_\varepsilon\}$, with $\mathcal{X}$ the $L^2(\Omega_x)$ phase-field space. We seek a single parameter vector $\theta = \vartheta \in \mathbb{R}^m$ such that the hypernetwork-form operator

$$\mathcal{P}^{\mathrm{NN}}_{\theta;\delta t}(\varepsilon, \mathbf{q}) \;=\; \Phi\big(\mathbf{q};\ h_\vartheta(\varepsilon)\big) \tag{4}$$

approximates $\mathcal{T}_{\delta t,\varepsilon}$ on $\mathcal{A}$, with Φ the target network of Section 3.2, $h_\vartheta : \mathbb{R} \to \mathbb{R}^p$ a compact MLP reading the scalar ε, and the slice-wise abbreviation $\mathcal{N}_{\theta,\delta t,\varepsilon}(\mathbf{q}) := \Phi(\mathbf{q}; h_\vartheta(\varepsilon))$ used in Section 3.3. With μ_E on E and μ_ε on $\mathcal{A}_\varepsilon$, training minimises the population one-step loss

$$\mathcal{L}(\theta) \;=\; \int_E \int_{\mathcal{A}_\varepsilon} \big\|\Phi(\mathbf{q}; h_\vartheta(\varepsilon)) - \mathcal{T}_{\delta t,\varepsilon}(\mathbf{q})\big\|^2_{L^2(\Omega_x)}\, \mathrm{d}\mu_\varepsilon(\mathbf{q})\, \mathrm{d}\mu_E(\varepsilon). \tag{5}$$

The empirical objective, the Riemann-sum discretisation on the $H \times W$ grid, the one-step rollout training, and the operator-level setup that motivates the rollout analysis of Section 3.3 are collected in Appendix C.

3.2 Proposed Approach

Approach overview. Inspired by the trainable DR splitting network of Bretin et al. [2] and by DISCO's hypernetwork-conditioning design [33], we construct IADR by coupling an MLP hypernetwork h_ϑ with a DR target network Φ that serves as the operator template. The hypernetwork generates the target network's weights from the interface width ε, and the target network uses these weights to evolve a given configuration. This design addresses the two requirements identified in Section 2: retaining the phase-field structure encoded in the target network's architecture while covering the range of ε with a single model.

This separation is also empirically motivated: independently training Φ at the five training widths produces structured variation in its diffusion and reaction weights across ε (Appendix C, Figure 3), motivating our choice to generate both sets of weights from ε through the hypernetwork.

Target network. The target network $\Phi(\cdot; w) : \mathcal{X} \to \mathcal{X}$ is the composition of a spatial diffusion block $\mathcal{D}_{w_D}$ and a pointwise reaction block $\mathcal{R}_{w_R}$, mirroring the closed-form update $\mathbf{q} \mapsto \rho_{1,\varepsilon}(K_1 * \mathbf{q})$ of (3):

$$\Phi(\mathbf{q}; w) \;=\; \mathcal{R}_{w_R}(\mathcal{D}_{w_D}(\mathbf{q}))\,, \qquad w = (w_R, w_D) \in \mathbb{R}^{p_R+p_D} =: \mathbb{R}^p. \tag{6}$$

Here, w_D and w_R are the weight vectors of the spatial diffusion and pointwise reaction blocks, respectively, both generated by the hypernetwork h_ϑ from the interface width ε, and their dimensions are p_D and p_R, respectively. For every ε, the operator is $\Phi(\cdot; h_\vartheta(\varepsilon))$, and the whole family is parameterised by the hypernetwork parameter vector ϑ, which contains all trainable parameters, together with the fixed structure of the target network Φ. No explicit ε channel enters Φ itself; the ε-dependence is concentrated in h_ϑ. We next describe the two blocks of the DR operator and the hypernetwork in more detail.

The diffusion block $\mathcal{D}_{w_D}$ is a single $K_D \times K_D$ two-dimensional convolution acting on $\mathbf{q}$ in the spatial domain, with kernel weights $w_D \in \mathbb{R}^{K_D^2}$ emitted by h_ϑ. For a given ε, the same kernel acts at every grid point, coupling spatial locations through a linear weighted sum of neighbouring phase-field values. Its output is an intermediate field on the same grid, which is then passed to the pointwise reaction block. Structurally it is the discrete analogue of the convolution $K_1 * \mathbf{q}$ in the closed-form first-order DR step (3), where K_1 is the spatial kernel of the Helmholtz resolvent $(I - \delta t\Delta)^{-1}$. The block has no bias and no spatial-channel mixing; its capacity is exactly the K_D^2 kernel entries, all generated by the hypernetwork. The concrete kernel size K_D is collected in Appendix C.

The reaction block $\mathcal{R}_{w_R}$ is a compact scalar-to-scalar multilayer perceptron $\mathrm{MLP} : \mathbb{R} \to \mathbb{R}$ acting pointwise on $\mathbf{q}$, and for every $x \in \Omega_x$, $(\mathcal{R}_{w_R}\mathbf{q})(x) = \mathrm{MLP}(\mathbf{q}(x))$. Within the target network Φ, this block receives the intermediate field produced by the diffusion block. At each grid point, the hidden layers transform the scalar phase-field input into nonlinear features, which the output layer maps back to a scalar phase-field value. For a given ε, the same MLP is shared across all grid points, so this step updates each value separately without mixing neighbouring values. Its weight vector $w_R \in \mathbb{R}^{p_R}$ is emitted by h_ϑ. We choose an MLP as the learned analogue of the cubic reaction $\rho_{1,\varepsilon}(s) = s - (\delta t/\varepsilon^2)W'(s)$ because hard-coding the cubic form would commit the network to the analytic profile of W', whereas a small MLP allows the reaction profile to adapt to ε and to absorb mild discretisation artefacts at the working amplitude band, while the scalar-to-scalar locality and translation invariance of the classical reaction step are preserved by construction.

Hypernetwork. The hypernetwork $h_\vartheta : \mathbb{R} \to \mathbb{R}^{p_R+p_D}$ is a multilayer perceptron with L_h hidden layers, hidden width h, smooth activation, scalar input ε and output dimension equal to the full weight count of Φ:

$$h_\vartheta(\varepsilon) \;=\; \alpha \cdot \mathrm{MLP}_\vartheta(\varepsilon), \tag{7}$$

where $\alpha > 0$ is a fixed scalar multiplier applied to the raw MLP output to keep the emitted weights at a workable scale; its concrete value and the network shapes are collected in Appendix C. The trainable parameters of IADR are exactly ϑ; the target network Φ has no independently trainable parameters, and at every forward pass its weights are set to $h_\vartheta(\varepsilon)$ via the weight-injection mechanism of DISCO, after which gradients flow through both networks. Capacity is therefore concentrated in ϑ, while the target network Φ remains the same compact two-block stack at every ε. The same hypernetwork interface extends to higher-order DR splittings; details are deferred to Appendix A.3.

3.3 Theoretical Analysis

Neural operators such as IADR advance the phase field one time step at a time; accurate long-horizon prediction therefore requires that single-step errors remain controlled over autoregressive rollouts of many steps. We establish a conditional K-step rollout error bound in terms of uniform one-step consistency τ and slice-wise Lipschitz stability L. The bound identifies a strict-contraction regime ($L < 1$), in which the rollout error stays uniformly bounded in K, and an error-growth regime ($L \geq 1$), under the stated assumptions.

Theorem 1 (K-step rollout stability, informal)**.** *For each $\varepsilon \in E$, let $\mathcal{T}_{\delta t,\varepsilon}$, $\mathcal{N}_{\theta,\delta t,\varepsilon} : \mathcal{A}_\varepsilon \to \mathcal{A}_\varepsilon$ be the exact and learned slice-wise propagators. Assume there exist constants $L_T \geq 0$, $L_N \geq 0$, and $\tau \geq 0$, independent of ε, such that for every $\varepsilon \in E$ and every $\mathbf{q}, \mathbf{q}' \in \mathcal{A}_\varepsilon$,*

$$\big\|\mathcal{T}_{\delta t,\varepsilon}(\mathbf{q}) - \mathcal{T}_{\delta t,\varepsilon}(\mathbf{q}')\big\|_{\mathcal{X}} \leq L_T \,\|\mathbf{q} - \mathbf{q}'\|_{\mathcal{X}},$$

$$\big\|\mathcal{N}_{\theta,\delta t,\varepsilon}(\mathbf{q}) - \mathcal{N}_{\theta,\delta t,\varepsilon}(\mathbf{q}')\big\|_{\mathcal{X}} \leq L_N \,\|\mathbf{q} - \mathbf{q}'\|_{\mathcal{X}},$$

$$\left\|\mathcal{N}_{\theta,\delta t,\varepsilon}(\mathbf{q}) - \mathcal{T}_{\delta t,\varepsilon}(\mathbf{q})\right\|_{\mathcal{X}} \le \tau.$$

Set $L := \max\{L_T, L_N\}$. Then for every integer $K \ge 1$,

$$\sup_{\varepsilon\in E}\sup_{\mathbf{q}\in\mathcal{A}_\varepsilon}\left\|\mathcal{T}^K_{\delta t,\varepsilon}(\mathbf{q}) - \mathcal{N}^K_{\theta,\delta t,\varepsilon}(\mathbf{q})\right\|_{\mathcal{X}} \;\le\; \tau\sum_{j=0}^{K-1} L^j.$$

Corollary 1 (Two regimes)**.** *Under the assumptions of Theorem 1, the bound $\tau\sum_{j=0}^{K-1} L^j$ specialises in two regimes:*

(i) if $L \ge 1$, the bound reduces to $\tau K\, L^{K-1}$, which grows linearly in K when $L = 1$ and at geometric rate L^{K-1} when $L > 1$;

(ii) if $L < 1$, the geometric series converges and

$$\sup_{\varepsilon\in E}\sup_{\mathbf{q}\in\mathcal{A}_\varepsilon}\left\|\mathcal{T}^K_{\delta t,\varepsilon}(\mathbf{q}) - \mathcal{N}^K_{\theta,\delta t,\varepsilon}(\mathbf{q})\right\|_{\mathcal{X}} \;\le\; \frac{\tau}{1-L},$$

uniformly in K, with residual decaying to zero at geometric rate L^K.

Proof sketch. The argument is a discrete error recursion on each slice $\mathcal{A}_\varepsilon$.

Step 1 (one-step error). Under Assumption 6, the one-step prediction error $\|\mathcal{T}_{\delta t,\varepsilon}(q) - \mathcal{N}_{\theta,\delta t,\varepsilon}(q)\|_{\mathcal{X}} \le \tau$ holds uniformly for $q \in \mathcal{A}_\varepsilon$.

Step 2 (slice-wise stability). Under Assumptions 4–5, both $\mathcal{T}_{\delta t,\varepsilon}$ and $\mathcal{N}_{\theta,\delta t,\varepsilon}$ are L-Lipschitz on $\mathcal{A}_\varepsilon$ with $L = \max\{L_T, L_N\}$, so the accumulated error is amplified by at most L per step.

Step 3 (telescoping). Iterating Steps 1–2 yields $e_K \le \tau\sum_{k=0}^{K-1} L^k$, which gives the strict-contraction regime ($L < 1$, error $\le \tau/(1-L)$ uniform in K) and the error-growth regime ($L \ge 1$, error $\le \tau\, K\, L^{K-1}$) of Corollary 1. The resulting estimate explains why long-horizon rollout is governed jointly by one-step consistency τ and by the slice-wise stability L of the learned operator family. The formal version with problem-level assumptions is given in Appendix B.2; smaller slice-wise Lipschitz constants L_N tighten the bound.

Theory and experiments. Under the stated assumptions, the bound relates one-step consistency and slice-wise stability to error accumulation, guiding the design of autoregressive rollout evaluation using worst-frame L^2-T^∞ errors. Tests at unseen interface widths and on non-circular geometries assess long-horizon accuracy under parameter and geometry shifts. The lemma in Appendix B.4 establishes first-order consistency of both DR splitting orders, supporting DR splitting as the target-network template.

4 Experiments

4.1 Experimental Setup

Setup and tasks. We evaluate IADR on the Allen–Cahn equation on the flat torus $\Omega_x = [0,1]^2$, assessing generalisation along two axes: interface width and geometry. This gives one in-distribution evaluation and three out-of-distribution evaluations: *Task 1* circles at E_{train} (in-distribution); *Task 2* circles at $\{0.0069, 0.0081, 0.0093\}$ (ε-OOD); *Task 3* non-circular geometries at $\varepsilon = 0.0078 \in E_{\text{train}}$ (geometry-OOD); and *Task 4* non-circular geometries at $\{0.0069, 0.0093\}$ (joint OOD). The domain is discretised on a 256×256 periodic grid with time step $\delta t = 6.10 \times 10^{-5}$, chosen to remain in the stability region of the explicit DR splitting across the interface-width range $E = [0.0065, 0.0090]$. The ε-axis is sampled at the five training widths $E_{\text{train}} = \{0.0065, 0.0072, 0.0078, 0.0084, 0.0090\} \subset E$, two in-distribution interpolation widths $\{0.0069, 0.0081\} \subset E$, and one OOD-coarse width $\varepsilon = 0.0093 \notin E$ ($\varepsilon > \varepsilon_{\max}$, outside the training interval but still an ε-OOD evaluation). Training uses only circular initial sets at the five E_{train} widths; no non-circular geometry and no width outside E_{train} is seen at training time. Circular trajectories follow the closed-form law $R(t) = \sqrt{R_0^2 - 2t}$, used as the exact reference for both training and evaluation. The non-circular geometries are *spiral disks*, disks deformed by a small radial ripple along the polar angle, and *five-pointed stars*, a central disk with five evenly spaced triangular arms; both are used only at evaluation time, with

Table 1: Main results. Best result per row in bold, second best underlined.

Shape	ε	L^2-T^∞ ($\times 10^{-3}$)					V-T^∞ ($\times 10^{-3}$)					AIW ($\times 10^{-3}$)				
		IADR	per-ε DR	U-Net	FNO	DISCO	IADR	per-ε DR	U-Net	FNO	DISCO	IADR	per-ε DR	U-Net	FNO	DISCO
Task 1: circles at training widths (in-distribution)																
Circle	0.0065	33.98	**33.75**	921.67	518.77	diverged	3.81	**3.79**	680.39	261.42	diverged	**51.84**	51.87	69.46	76.61	diverged
	0.0072	27.06	**18.89**	951.33	510.70	6.7×10^{13}	2.97	**2.09**	709.87	269.69	7.2×10^{12}	0.34	**0.32**	25.62	24.51	398.57
	0.0078	22.35	**6.93**	959.88	508.08	4.7×10^{13}	2.37	**0.92**	728.62	269.20	5.7×10^{12}	0.48	**0.37**	17.45	20.81	411.65
	0.0084	14.78	**4.53**	974.31	508.02	2.5×10^{12}	1.39	**0.94**	745.59	264.67	2.3×10^{11}	0.54	**0.35**	23.25	16.86	419.19
	0.0090	11.80	**7.36**	984.91	509.65	1.6×10^{18}	1.18	**0.84**	755.55	268.61	3.5×10^{17}	1.00	**0.50**	27.56	14.84	409.46
Task 2: circles at non-training widths (ε-OOD)																
Circle	0.0069	28.08	**25.46**	937.71	508.98	1.1×10^{10}	3.08	**2.82**	698.39	265.67	8.7×10^{8}	**0.37**	0.41	24.17	29.90	396.72
	0.0081	19.99	**9.24**	968.03	500.88	7.1×10^{9}	2.07	**1.16**	737.03	265.36	7.1×10^{8}	0.64	**0.50**	20.79	20.59	416.88
	0.0093	65.95	**8.27**	778.98	510.99	3.1×10^{11}	9.97	**0.95**	549.11	269.45	4.8×10^{10}	16.76	**0.85**	40.81	12.11	410.05
Task 3: non-circular geometries at training widths (geometry-OOD)																
Five-pointed star	0.0078	21.85	**8.80**	757.35	553.48	406.12	3.77	**1.62**	603.77	229.99	224.92	1.98	**1.87**	11.52	19.34	184.83
Spiral disk	0.0078	4.03	**3.65**	119.04	382.58	205.19	1.08	**0.96**	33.76	86.77	16.24	0.44	0.35	**0.23**	18.04	295.82
Task 4: non-circular geometries at non-training widths (joint OOD)																
Five-pointed star	0.0069	16.73	**13.35**	720.87	556.31	407.88	2.81	**2.25**	547.74	231.35	222.74	**1.70**	1.71	7.95	20.80	180.18
	0.0093	50.26	**28.34**	790.93	550.19	393.26	10.75	**5.23**	659.78	229.31	222.89	**2.33**	2.38	9.06	13.43	186.33
Spiral disk	0.0069	7.71	**7.59**	112.61	373.37	213.23	1.97	**1.95**	27.20	82.75	16.06	0.34	0.36	**0.19**	24.03	286.82
	0.0093	14.16	**3.72**	170.10	382.93	205.85	4.24	**0.64**	58.13	88.09	16.00	0.56	0.56	**0.49**	8.02	293.18

reference trajectories computed by the higher-order DR splitting scheme. Each model takes a single conditioning frame, except DISCO, which uses 16 consecutive ground-truth frames, and rolls out autoregressively to a fixed horizon T_{eval}. Initial-set parameters, sample counts, and trajectory budgets for all three datasets are listed in Appendix D.1, Table 3.

Baselines. We select baselines from the three lines reviewed in Section 2. *For PDE-specific operator learning*, we use *Per-ε DR*: an independent DR splitting network [2] retrained at each of the five training widths, serving as our expert (SOTA) reference. *For general operator learning*, we use *FNO* [27], a Fourier neural operator trained on the same data with ε concatenated to the input channel. *For meta-learning over PDE operators*, we use *DISCO* [33]: the original DISCO pipeline unchanged, in which a space-time Transformer encodes a T-frame trajectory and a hypernetwork maps the resulting code into a frozen U-Net target. We additionally include an in-house *U-Net variant* that swaps IADR's diffusion-and-reaction target for the four-level U-Net of [33] while keeping the rest of the conditioning pipeline; this isolates the role of IADR's physics-structured target by holding the conditioning pipeline fixed and is not a representative of any external category. DISCO trains for 100 epochs and all other models for 20,000 epochs. Architectures and full hyperparameters are listed in Appendix C. Two further ablations on the role of ε and on the conditioning mechanism are reported in Appendix F.

Metrics. To assess whether a learned operator is physically faithful over an autoregressive rollout, we report three classes of metric on each trajectory. **(i) L^2 accuracy.** The L^2-T^∞ error, taken from Bretin et al. [2], is the worst-frame physical L^2 error per trajectory; it quantifies how far the predicted phase field strays from the reference solution at the worst moment of the rollout. **(ii) Geometric mass conservation.** The Volume-T^∞ error, also taken from Bretin et al. [2], is the worst-frame deviation of the predicted enclosed mass from that of the reference solution; on the circle dataset it coincides with the analytical mean-curvature decay $A'(t) = -2\pi$, so it directly tests whether the model respects the geometric mass-loss law of the underlying mean-curvature flow. **(iii) Diffuse-interface consistency.** The AIW metric, which we introduce, is the deviation of the predicted average interface width from its ε-controlled target at a fixed late-time frame; it certifies that the diffuse-interface width remains consistent with the prescribed ε rather than spreading or sharpening unphysically. We add AIW because the existing L^2 and volume metrics do not directly penalise drift in the diffuse layer. Discrete formulas and aggregation rules are given in Appendix D.2.

4.2 Main Results

Table 1 reports the three physics-aligned metrics on the four-task suite, and Table 2 the single-step inference cost; full $\text{mean} \pm \text{std}$ in Table 5.

IADR matches the per-ε DR specialists on accuracy and physical consistency within the prescribed range while every generic operator target lags behind. On Task 1 (circles at training widths), IADR's L^2-T^∞, V-T^∞, and AIW closely match the matching per-ε DR specialist, with the specialist attaining the smaller mean in five of five widths on L^2-T^∞ and V-T^∞ and in four of five widths on AIW. On Task 2 (circles at non-training widths in E), IADR also recovers the specialist level on the two interpolation widths $\{0.0069, 0.0081\}$ across every metric. By contrast, every generic operator target sits well above the per-ε specialist on the same cells: at circle-0.0078, IADR reads L^2-$T^\infty = 22.35 \times 10^{-3}$ and the specialist 6.93×10^{-3}, while FNO reads 508.08×10^{-3} and the U-Net target 959.88×10^{-3}; V-T^∞ and AIW follow the same pattern, with DISCO diverging to the 10^{13} scale. The DR splitting target therefore gives the network the level of physical consistency only a per-ε specialist achieves, and the ε-conditioning lets a single IADR hold this level across the range, realising contribution (iii) of Section 1. On the OOD-coarse width $\varepsilon = 0.0093$, by contrast, IADR's L^2-T^∞ jumps to 65.95×10^{-3}, well above the per-ε specialist's 8.27×10^{-3}. To further examine the need for full DR weight generation and evaluate IADR's ability to cover a wider ε range, we also compare IADR with Direct-ε-DR and specialist weight interpolation on circles, and train and evaluate IADR over a wider range (see Appendix E & F).

IADR preserves the per-ε specialists' cross-shape generalisation when transferred to non-circular geometries. Tasks 3 and 4 transfer the trained model to non-circular geometries (five-pointed star and spiral disk), neither seen during training. On Task 3 (training widths), IADR stays close to the matching-ε specialist on every cell across all three metrics. On Task 4 (non-training widths), IADR extends this behaviour into the joint OOD regime: at the OOD-coarse width $\varepsilon = 0.0093$, the per-ε specialist outperforms IADR on five-pointed star (28.34×10^{-3} vs 50.26×10^{-3} on L^2-T^∞); on spiral disk at the same width, IADR performs worse than the specialist (14.16×10^{-3} vs 3.72×10^{-3}) but remains well below every generic operator target on this cell (U-Net 170.10×10^{-3}, FNO 382.93×10^{-3}, DISCO 205.85×10^{-3}). Across almost every Task 3 and Task 4 cell the generic operator targets stay consistently above IADR and the per-ε specialists. A single ε-conditional model therefore inherits the per-ε specialist's level of cross-shape generalisation rather than trading it for ε-coverage, completing contribution (iii).

IADR attains this accuracy at a notably lower per-step inference cost than every generic operator target. Table 2 reports the single-step forward time on a 256×256 grid, aggregated by shape. IADR runs at roughly 1.3 ms per step across all three shape families, consistently faster than the U-Net target reproduced under the same conditioning pipeline (4.6 ms), the FNO general operator (3.3–3.4 ms), and DISCO (105–119 ms; about 80–$90\times$ slower than IADR, driven by the space-time Transformer encoder over the T-frame trajectory). IADR retains the compact DR target structure of the per-ε specialists and covers the range with one model, avoiding retraining for new ε values within that range. At inference, the additional weight generation and injection lead to a single-step time of 1.27–1.36 ms, compared with 0.61–0.62 ms for the specialists. Both remain at the millisecond scale and are substantially faster than the generic operator targets. Despite this additional per-step cost, training one IADR takes 16 h, compared with 24 h in total for the eight specialists in our comparison, demonstrating its lower total training cost across multiple ε values.

Table 2: Single-step inference time (ms/sample on 256×256, mean $\pm$ std across ε within each shape). Best result per row in bold, second best underlined.

shape	IADR	per-ε DR	U-Net	FNO	DISCO
circle	$\underline{1.30 \pm 0.02}$	$\mathbf{0.61 \pm 0.02}$	4.63 ± 0.09	3.34 ± 0.01	119.13 ± 3.55
five-pointed star	$\underline{1.27 \pm 0.01}$	$\mathbf{0.62 \pm 0.02}$	4.56 ± 0.02	3.38 ± 0.01	107.29 ± 1.77
spiral disk	$\underline{1.36 \pm 0.11}$	$\mathbf{0.62 \pm 0.03}$	4.60 ± 0.08	3.38 ± 0.00	104.62 ± 2.68

4.3 Case Study

To complement the aggregate trends in Table 1 we visualise representative rollouts that anchor the findings of Section 4.2 to single-frame qualitative behaviour.

IADR and the per-ε specialist match ground truth across the full rollout, while every generic operator target collapses. Figure 2a shows the in-distribution circle at $\varepsilon = 0.0078$. IADR

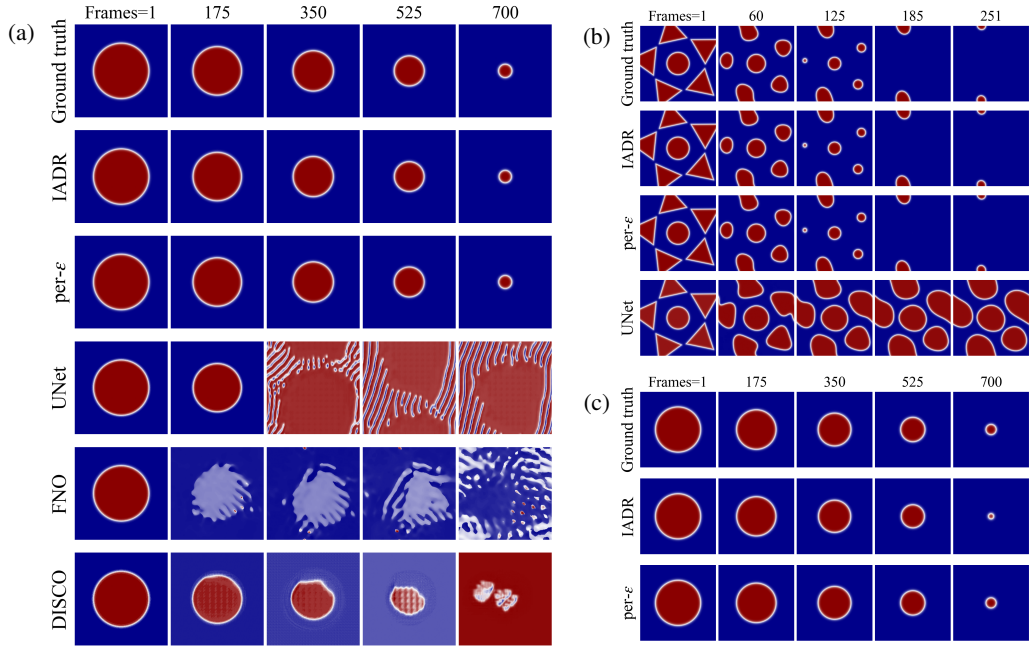


Figure 2: Case-study rollouts; frame indices and model labels are marked on each panel. a In-distribution circle at $\varepsilon = 0.0078$. b Joint-OOD five-pointed star at $\varepsilon = 0.0069$. c OOD-coarse extrapolation $\varepsilon = 0.0093 \notin E$ on a circle.

(L^2-$T^\infty = 22.35 \times 10^{-3}$) and the per-$\varepsilon$ DR specialist (6.93×10^{-3}) coincide with ground truth at every frame, while U-Net dissolves into spurious wave patterns by frame 350 (959.88×10^{-3}), FNO collapses to a low-amplitude blob by frame 175 (508.08×10^{-3}), and DISCO fragments into a disconnected cluster by frame 700 (diverged, L^2-$T^\infty \sim 10^{13}$).

IADR retains the per-ε specialist's geometric generalisation on a joint-OOD shape. Figure 2b shows the five-pointed star at $\varepsilon = 0.0069$ (joint OOD). IADR (L^2-$T^\infty = 16.73 \times 10^{-3}$), per-$\varepsilon$ DR (13.35×10^{-3}), and ground truth follow the same evolution from sharp arms through breakup to a small final residue, while the U-Net target (720.87×10^{-3}) inflates the geometry instead of contracting it. Despite never seeing a non-circular shape during training, IADR stays within $1.25\times$ of the per-ε specialist on this joint-OOD cell and improves over every generic-operator baseline by more than an order of magnitude on L^2-T^∞ (U-Net $43\times$, FNO $33\times$, DISCO $24\times$ worse), evidence that the DR-structured target preserves cross-shape generalisation under hypernetwork conditioning.

IADR's evolution speed drifts outside the training range, but the phase-field structure is preserved. Figure 2c shows the OOD-coarse extrapolation $\varepsilon = 0.0093 \notin E$. The per-$\varepsilon$ specialist (L^2-$T^\infty = 8.27 \times 10^{-3}$) tracks ground truth throughout, while IADR (65.95×10^{-3}) shows a circle at frame 700 visibly smaller than the reference, indicating a faster effective evolution speed. The interface nonetheless remains a clean phase-field profile, rather than the unstructured collapse seen for U-Net (778.98×10^{-3}), FNO (510.99×10^{-3}), or DISCO ($\sim 10^{11}$) on the same cell, because the DR backbone is hard-coded into the target.

5 Conclusion

To meet the practical need to reduce repeated training across interface widths while retaining the phase-field structure and prediction capability of DR operators, we presented IADR, a single ε-conditional neural operator for the Allen–Cahn equation that pairs a DR splitting target network with a hypernetwork reading the scalar interface width. A single IADR matches the per-ε DR specialists on in-distribution circles within the prescribed range, retains their cross-shape generalisation on unseen geometries, and retains their compact DR target structure with millisecond-scale inference. The ablations and comparisons with Direct-ε-DR and specialist weight interpolation further demonstrate

the value of full DR conditioning for prediction accuracy and interface-width consistency. The conditional K-step rollout error bound we establish separates one-step consistency from slice-wise stability.

6 Limitation and future work

Our discussion of limitations and future work focuses on three aspects: prediction of dynamics involving topological changes with the first-order DR target, architecture optimisation and systematic evaluation across different parameter families, and cross-equation applicability and architecture design.

The first-order DR target may limit prediction performance for dynamics involving topological changes. The dumbbell decomposition experiment in Appendix E provides preliminary evidence that IADR retains the DR operator's ability to predict dynamics involving a simple topological change. However, both the per-ε specialists and IADR perform poorly overall in this experiment. Further work is needed to improve the operator design for such dynamics and to conduct a more systematic evaluation.

Our evaluation focuses on the Allen–Cahn equation under fixed numerical settings, with the interface width ε as the only varying physical parameter. Future work will study joint conditioning on multiple physical parameters and evaluate extensions across different resolutions, time steps, and boundary conditions. We will also explore separating and recombining ε-related and ε-independent features and further study how to incorporate physical constraints into the hypernetwork mechanism, aiming to improve accuracy beyond that of per-ε specialists.

Although this work focuses only on the Allen–Cahn equation, it also provides a path for cross-equation extensions. The hypernetwork conditioning interface can be retained while adapting the target network and conditioning variables to the corresponding splitting or physical decomposition. When a single MLP generates these weights, its output multiplier α should be selected with the weight scales of the constituent operators in mind. We may also explore using two MLPs to reduce reliance on the output multiplier α. For cross-equation extensions, we will first instantiate a target network for a second phase-field PDE, such as the Cahn–Hilliard equation, and independently evaluate the framework's cross-equation transferability.

Acknowledgments and Disclosure of Funding

This research is supported by the National Natural Science Foundation of China (No. 12571435 and No. 11971336). The authors declare no competing interests.

References

[1] Rafael Bischof, Michal Piovarči, Michael A. Kraus, Siddhartha Mishra, and Bernd Bickel. HyPINO: Multi-physics neural operators via HyperPINNs and the method of manufactured solutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[2] Elie Bretin, Roland Denis, Simon Masnou, and Garry Terii. Learning phase field mean curvature flows with neural networks. *Journal of Computational Physics*, 470:111579, 2022. doi: 10.1016/j.jcp.2022.111579.

[3] Yadi Cao, Yuxuan Liu, Liu Yang, Rose Yu, Hayden Schaeffer, and Stanley Osher. ViCoN: Vision in-context operator networks for multi-physics fluid dynamics prediction. *arXiv preprint arXiv:2411.16063*, 2024.

[4] Long-Qing Chen. Phase-field models for microstructure evolution. *Annual Review of Materials Research*, 32:113–140, 2002. doi: 10.1146/annurev.matsci.32.112001.132041.

[5] Long-Qing Chen and Jie Shen. Applications of semi-implicit Fourier-spectral method to phase field equations. *Computer Physics Communications*, 108(2–3):147–158, 1998. doi: 10.1016/S0010-4655(97)00115-X.

[6] Xinfu Chen. Generation and propagation of interfaces for reaction-diffusion equations. *Journal of Differential Equations*, 96(1):116–141, 1992. doi: 10.1016/0022-0396(92)90146-E.

[7] Yuyan Chen, Bin Dong, and Jinchao Xu. Meta-MgNet: Meta multigrid networks for solving parameterized partial differential equations. *Journal of Computational Physics*, 455:110996, 2022. doi: 10.1016/j.jcp.2022.110996.

[8] Woojin Cho, Kookjin Lee, Donsub Rim, and Noseong Park. Hypernetwork-based meta-learning for low-rank physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[9] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022. doi: 10.1007/s10915-022-01939-z.

[10] Piero De Mottoni and Michelle Schatzman. Geometrical evolution of developed interfaces. *Transactions of the American Mathematical Society*, 347(5):1533–1589, 1995. doi: 10.1090/S0002-9947-1995-1672406-4.

[11] Klaus Deckelnick, Gerhard Dziuk, and Charles M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232, 2005. doi: 10.1017/S0962492904000224.

[12] Qiang Du and Xiaobing Feng. The phase field method for geometric moving interfaces and their numerical approximations. In *Geometric Partial Differential Equations, Part I*, volume 21 of *Handbook of Numerical Analysis*, pages 425–508. Elsevier, 2020.

[13] L. C. Evans and J. Spruck. Motion of level sets by mean curvature. I. *Journal of Differential Geometry*, 33(3):635–681, 1991. doi: 10.4310/jdg/1214446559.

[14] David J. Eyre. Unconditionally gradient stable time marching the Cahn–Hilliard equation. In *Computational and Mathematical Models of Microstructural Evolution*, volume 529 of *Materials Research Society Symposium Proceedings*, pages 39–46. Materials Research Society, 1998.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

[16] David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations (ICLR)*, 2017.

[17] Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. *arXiv preprint arXiv:2403.03542*, 2024.

[18] Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. *arXiv preprint arXiv:2405.19101*, 2024.

[19] Tom Ilmanen. Convergence of the Allen–Cahn equation to Brakke's motion by mean curvature. *Journal of Differential Geometry*, 38(2):417–461, 1993. doi: 10.4310/jdg/1214454300.

[20] Pengzhan Jin, Shuai Meng, and Lu Lu. MIONet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022. doi: 10.1137/22M1477751.

[21] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

[22] 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.

[23] Armand Kassaï Koupaï, Jorge Mifsut Benet, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. GEPS: Boosting generalization in parametric PDE neural solvers through adaptive conditioning. *arXiv preprint arXiv:2410.23889*, 2024.

[24] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[25] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[26] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[27] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations (ICLR)*, 2021.

[28] 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. *arXiv preprint arXiv:2111.03794*, 2021.

[29] 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.

[30] Revanth Mattey and Susanta Ghosh. A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations. *Computer Methods in Applied Mechanics and Engineering*, 390:114474, 2022. doi: 10.1016/j.cma.2021.114474.

[31] Robert I. McLachlan and G. R. W. Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002. doi: 10.1017/S0962492902000053.

[32] Luciano Modica and Stefano Mortola. Un esempio di γ^--convergenza. *Bollettino della Unione Matematica Italiana, Serie V*, 14-B:285–299, 1977.

[33] Rudy Morel, Jiequn Han, and Edouard Oyallon. DISCO: Learning to DISCover an evolution operator for multi-physics-agnostic prediction. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[34] 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.

[35] Ingo Steinbach. Phase-field models in materials science. *Modelling and Simulation in Materials Science and Engineering*, 17(7):073001, 2009. doi: 10.1088/0965-0393/17/7/073001.

[36] Colby L. Wight and Jia Zhao. Solving Allen–Cahn and Cahn–Hilliard equations using the adaptive physics informed neural networks. *Communications in Computational Physics*, 29(3): 930–954, 2021.

A Phase-Field Formulation and DR Splitting Schemes

This appendix expands on the phase-field background summarised in Section 3.1, derives the first-order DR splitting scheme of (3), and records the higher-order $K_2 * \rho_2$ variant referenced in Section 3.2.

A.1 Mean-curvature flow and phase-field representation

Mean-curvature flow and the Cahn–Hilliard energy. A smooth domain $\Omega(t) \subset \mathbb{R}^d$ evolves under mean-curvature flow if its inward normal velocity satisfies $V_n(t) = H(t)$, where $H(t)$ is the scalar mean curvature of $\partial\Omega(t)$. This is the L^2-gradient flow of the perimeter functional $P(\Omega(t)) = \int_{\partial\Omega(t)} \mathrm{d}\mathcal{H}^{d-1}$, with $\mathcal{H}^{d-1}$ the $(d-1)$-dimensional Hausdorff measure. In the phase-field approach [4, 35], the perimeter is approximated (up to a constant) by the Cahn–Hilliard energy

$$P_\varepsilon(u) \;=\; \int_{\mathbb{R}^d} \left(\varepsilon \frac{|\nabla u|^2}{2} + \frac{1}{\varepsilon} W(u) \right) \mathrm{d}x,$$

with double-well potential $W(s) = \frac{1}{2} s^2 (1-s)^2$ fixed throughout. The energy Γ-converges to P [32], and its L^2-gradient flow is the Allen–Cahn equation (1).

Exact phase-field representation and well-prepared data. Associated with (1) is the evolving set $\Omega_\varepsilon(t) = \{u_\varepsilon(\cdot, t) \geq \frac{1}{2}\}$, with u_ε the solution under the well-prepared initial data $u_\varepsilon(x, 0) = q(d(x, \Omega(0))/\varepsilon)$ for the optimal profile $q(r) = \frac{1}{2}(1 - \tanh(r/2))$. A formal asymptotic expansion near the interface gives

$$u_\varepsilon(x, t) \;=\; q\Big(\tfrac{d(x, \Omega_\varepsilon(t))}{\varepsilon}\Big) + \mathcal{O}(\varepsilon^2),$$

so that u_ε stays close to the exact representation $\mathbf{q}_\varepsilon$ defined in (2). Before the onset of singularities, the Hausdorff distance between $\Omega_\varepsilon(t)$ and the true MCF evolution $\Omega(t)$ satisfies

$$\mathrm{dist}_{\mathcal{H}}\big(\Omega(t),\, \Omega_\varepsilon(t)\big) \;\leq\; C\, \varepsilon^2 |\log \varepsilon|^2, \tag{8}$$

with C depending on the regularity of $\Omega(t)$ [10, 6], which is the tighter-reference statement used in Section 3.1.

A.2 First-order DR splitting scheme

The first-order DR splitting [2] is a semi-implicit discretisation of (1): given $\delta t > 0$, one defines $(u^n)_{n \geq 0}$ by

$$\frac{u^{n+1} - u^n}{\delta t} \;=\; \Delta u^{n+1} - \frac{1}{\varepsilon^2} W'(u^n), \tag{9}$$

which rearranges to $u^{n+1} = (I - \delta t\, \Delta)^{-1}(u^n - (\delta t/\varepsilon^2) W'(u^n)) = K_1 * \rho_1(u^n)$ with $\rho_1, \widehat{K}_1$ as in (3). The scheme decouples the diffusion and reaction operators and reaches sub-grid accuracy with only a handful of parameters per ε, but depends explicitly on $(\delta t, \varepsilon)$; changing either forces a rebuild.

A.3 Higher-order DR splitting

The second-order DR splitting scheme [2] reads $u^{n+1} = K_2 * \rho_2(u^n)$ with $\rho_2(s) = s - (\delta t/\varepsilon^2)(W'(s) - \alpha s)$ and $\widehat{K}_2(\xi) = 1/(1 + \delta t(4\pi^2|\xi|^2 + \alpha/\varepsilon^2))$ for a fixed shift $\alpha > 0$. The corresponding DR splitting network Φ_2 used by IADR at second order mirrors this scheme by composing two reaction blocks and two spectral-convolution diffusion blocks with a skip connection from the input field to the second reaction block, following the $M = 2$ stage layout [2]. The hypernetwork interface is unchanged: $h_\vartheta(\varepsilon)$ now supplies the concatenated weight vector of the two (ρ, K) stages together with the skip coefficients. Fixed-step rollouts evaluate long-horizon prediction. The temporal accuracy of higher-order models will be assessed through time-step refinement experiments in future work.

B Proofs and Derivations

This appendix gives (i) a linear non-expansive special case of Theorem 1 in which the exact propagator is linear, with a tighter telescoping bound; (ii) the formal version of Theorem 1 together with explicit problem-level and direct hypotheses, and a complete proof; and (iii) supporting material for the formal version, namely a concrete Lipschitz estimate on a spectral-gap subspace of the DR splitting backbone, and the first-order equivalence of the two natural DR splitting orders.

B.1 Linear non-expansive special case

We state a linear non-expansive special case of Theorem 1 in which the exact propagator is linear on a working subspace and the loss is stated through the operator inverse. In this regime the algebraic telescoping identity $\mathcal{P}^K - \mathcal{Q}^K = \sum_{j=0}^{K-1} \mathcal{P}^j (\mathcal{P} - \mathcal{Q}) \mathcal{Q}^{K-1-j}$ yields a tighter bound than the recursive estimate of Theorem 1.

Theorem 2 (Linear non-expansive special case)**.** *Let $\widetilde{\mathcal{X}} = \mathbb{R} \times \mathcal{X}$ equipped with the norm $\|(r, x)\|_{\widetilde{\mathcal{X}}} := (|r|^2 + \|x\|_{\mathcal{X}}^2)^{1/2}$. Assume the exact propagator $\mathcal{P}_{\delta t} : \widetilde{\mathcal{X}} \to \mathcal{X}$ is linear and non-expansive,*

$$\|\mathcal{P}_{\delta t}\|_{\widetilde{\mathcal{X}}^*} := \sup_{g \in \widetilde{\mathcal{X}},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \|\mathcal{P}_{\delta t}\, g\|_{\mathcal{X}} \;\le\; 1. \tag{10}$$

Let $\mathcal{U} \subseteq \widetilde{\mathcal{X}}$ be a closed linear subspace with $\mathcal{P}_{\delta t}\,\mathcal{U} \subseteq \mathcal{U}$, on which $\mathcal{P}_{\delta t}$ is invertible. If the learned linear approximant $\mathcal{P}_{\theta;\delta t}^{\mathrm{NN}}$ satisfies the loss bound

$$\sup_{g \in \mathcal{U},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \left\| \mathcal{P}_{\delta t}^{-1}\, \mathcal{P}_{\theta;\delta t}^{\mathrm{NN}}\, g - g \right\|_{\widetilde{\mathcal{X}}} \;\le\; \epsilon, \tag{11}$$

then for every integer $K \ge 1$,

$$\sup_{g \in \mathcal{U},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \left\| \mathcal{P}_{\delta t}^{K}\, g - (\mathcal{P}_{\theta;\delta t}^{\mathrm{NN}})^K\, g \right\|_{\mathcal{X}} \;\le\; \epsilon\, K\, (1+\epsilon)^K. \tag{12}$$

Moreover, if $\|\mathcal{P}_{\delta t}\|_{\widetilde{\mathcal{X}}^*} \le \eta < 1$ and $\epsilon \le (1-\eta)/2$, then

$$\sup_{g \in \mathcal{U},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \left\| \mathcal{P}_{\delta t}^{K}\, g - (\mathcal{P}_{\theta;\delta t}^{\mathrm{NN}})^K\, g \right\|_{\mathcal{X}} \;\le\; \epsilon\, K \left(\frac{1+\eta}{2} \right)^{K-1}. \tag{13}$$

Proof of Theorem 2. We work on the unit ball of $\mathcal{U}$, $\{g \in \mathcal{U} : \|g\|_{\widetilde{\mathcal{X}}} = 1\}$, throughout. For brevity write $\mathcal{P} := \mathcal{P}_{\delta t}$ and $\mathcal{P}_{\mathrm{NN}} := \mathcal{P}_{\theta;\delta t}^{\mathrm{NN}}$.

Step 1: from inverse-form loss to operator error. For any $g \in \mathcal{U}$ with $\|g\|_{\widetilde{\mathcal{X}}} = 1$,

$$\|\mathcal{P}\, g - \mathcal{P}_{\mathrm{NN}}\, g\|_{\mathcal{X}} \;=\; \left\| \mathcal{P}\,(g - \mathcal{P}^{-1}\, \mathcal{P}_{\mathrm{NN}}\, g) \right\|_{\mathcal{X}} \;\le\; \|\mathcal{P}\|_{\widetilde{\mathcal{X}}^*} \cdot \epsilon \;\le\; \epsilon,$$

where the equality uses linearity of $\mathcal{P}$ together with its invertibility on $\mathcal{U}$, the first inequality uses non-expansiveness (10) and the loss bound (11), and the second uses $\|\mathcal{P}\|_{\widetilde{\mathcal{X}}^*} \le 1$. Taking the supremum,

$$\sup_{g \in \mathcal{U},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \|\mathcal{P}\, g - \mathcal{P}_{\mathrm{NN}}\, g\|_{\mathcal{X}} \;\le\; \epsilon. \tag{14}$$

Step 2: bound on the learned operator's norm. The triangle inequality, non-expansiveness, and (14) give

$$\sup_{g \in \mathcal{U},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \|\mathcal{P}_{\mathrm{NN}}\, g\|_{\mathcal{X}} \;\le\; \|\mathcal{P}\|_{\widetilde{\mathcal{X}}^*} + \epsilon \;\le\; 1 + \epsilon.$$

Step 3: telescoping identity. For linear operators on the invariant subspace $\mathcal{U}$,

$$\mathcal{P}^K - \mathcal{P}_{\mathrm{NN}}^K \;=\; \sum_{j=0}^{K-1} \mathcal{P}^j\, (\mathcal{P} - \mathcal{P}_{\mathrm{NN}})\, \mathcal{P}_{\mathrm{NN}}^{K-1-j}.$$

Applying triangle inequality on the unit ball of $\mathcal{U}$, sub-multiplicativity of operator norms, and the bounds $\|\mathcal{P}^j\|_{\widetilde{\mathcal{X}}^*} \le 1$, $\|\mathcal{P} - \mathcal{P}_{\mathrm{NN}}\| \le \epsilon$ (from (14)), $\|\mathcal{P}_{\mathrm{NN}}^{K-1-j}\| \le (1+\epsilon)^{K-1-j}$ (from Step 2),

$$\sup_{g \in \mathcal{U},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \|\mathcal{P}^K\, g - \mathcal{P}_{\mathrm{NN}}^K\, g\|_{\mathcal{X}} \;\le\; \sum_{j=0}^{K-1} 1 \cdot \epsilon \cdot (1+\epsilon)^{K-1-j} \;\le\; \epsilon\, K\, (1+\epsilon)^K,$$

which proves the first claim.

Step 4: strict-contraction branch. If $\|\mathcal{P}\|_{\widetilde{\mathcal{X}}^*} \le \eta < 1$ and $\epsilon \le (1-\eta)/2$, Step 2 sharpens to $\|\mathcal{P}_{\mathrm{NN}}\| \le \eta + \epsilon \le (1+\eta)/2$. Substituting $\|\mathcal{P}^j\| \le \eta^j$ and $\|\mathcal{P}_{\mathrm{NN}}^{K-1-j}\| \le ((1+\eta)/2)^{K-1-j}$ into the telescoping bound, and noting that each term is dominated by $((1+\eta)/2)^{K-1}$ since $\eta < (1+\eta)/2$,

$$\sup_{g \in \mathcal{U},\, \|g\|_{\widetilde{\mathcal{X}}}=1} \|\mathcal{P}^K\, g - \mathcal{P}_{\mathrm{NN}}^K\, g\|_{\mathcal{X}} \;\le\; \epsilon \sum_{j=0}^{K-1} \eta^j \left(\frac{1+\eta}{2} \right)^{K-1-j} \;\le\; \epsilon\, K \left(\frac{1+\eta}{2} \right)^{K-1},$$

which proves the second claim. □

B.2 Formal version and proof of Theorem 1

This subsection states the formal slice-wise version of Theorem 1 and gives its complete proof. The setting is nonlinear (through the reaction $\rho_{1,\varepsilon}$) and varies with $\varepsilon \in E$, so non-expansiveness and the inverse-form loss of the linear special case (Theorem 2) are replaced by slice-wise Lipschitz stability and a uniform one-step consistency error. This conditional bound describes error propagation under these assumptions.

Setup. We adopt the notation of Section 3.1. Fix $\varepsilon \in E$ and write $\mathcal{T}_\varepsilon := \mathcal{T}_{\delta t,\varepsilon}$ and $\mathcal{N}_\varepsilon := \mathcal{N}_{\theta,\delta t,\varepsilon}$ for the exact and learned one-step propagators at interface width ε. Both are treated as self-maps of the admissible set $\mathcal{A}_\varepsilon \subset \mathcal{X}$ under Assumption 2; the ε-Lipschitz regularity of $\mathcal{T}_\varepsilon$ (Assumption 3) and the coverage of the range by μ_E (Assumption 1) are not used directly in the slice-wise proof but justify passing from sampled to uniform control over E.

Problem-level assumptions. The three problem-level assumptions summarised in Section 3.1 are formalised as follows.

Assumption 1 (Coverage of the range)**.** *The measure μ_E is absolutely continuous with respect to Lebesgue measure on E and has full support, so that a minimiser of* (5) *controls the loss uniformly over E rather than at a finite set of sampled widths.*

Assumption 2 (Slice-wise regularity)**.** *For every $\varepsilon \in E$, the admissible set $\mathcal{A}_\varepsilon$ is bounded in $\mathcal{X}$ and positively invariant under $\mathcal{T}_\varepsilon$; that is, $\mathcal{T}_\varepsilon(\mathcal{A}_\varepsilon) \subseteq \mathcal{A}_\varepsilon$ up to the horizon T.*

Assumption 3 (Lipschitz continuity in ε)**.** *There exists a constant $C_E > 0$ such that for every $\varepsilon, \varepsilon' \in E$ and every $\mathbf{q} \in \mathcal{A}_\varepsilon \cap \mathcal{A}_{\varepsilon'}$, $\|\mathcal{T}_\varepsilon(\mathbf{q}) - \mathcal{T}_{\varepsilon'}(\mathbf{q})\|_{L^2(\Omega_x)} \le C_E\,|\varepsilon - \varepsilon'|$.*

Direct hypotheses of the slice-wise extension. The three direct hypotheses are as follows.

Assumption 4 (Slice-wise Lipschitz stability of the target)**.** *There exists a constant $L_T \ge 0$ such that for every $\varepsilon \in E$ and every $\mathbf{q}, \mathbf{q}' \in \mathcal{A}_\varepsilon$, $\left\|\mathcal{T}_\varepsilon(\mathbf{q}) - \mathcal{T}_\varepsilon(\mathbf{q}')\right\|_{\mathcal{X}} \le L_T\,\|\mathbf{q} - \mathbf{q}'\|_{\mathcal{X}}$.*

Assumption 5 (Slice-wise Lipschitz stability of the learned operator)**.** *There exists a constant $L_N \ge 0$ such that for every $\varepsilon \in E$ and every $\mathbf{q}, \mathbf{q}' \in \mathcal{A}_\varepsilon$, $\left\|\mathcal{N}_\varepsilon(\mathbf{q}) - \mathcal{N}_\varepsilon(\mathbf{q}')\right\|_{\mathcal{X}} \le L_N\,\|\mathbf{q} - \mathbf{q}'\|_{\mathcal{X}}$.*

Assumption 6 (Uniform one-step consistency)**.** *There exists $\tau \ge 0$ such that for every $\varepsilon \in E$ and every $\mathbf{q} \in \mathcal{A}_\varepsilon$, $\left\|\mathcal{N}_\varepsilon(\mathbf{q}) - \mathcal{T}_\varepsilon(\mathbf{q})\right\|_{\mathcal{X}} \le \tau$.*

On the consistency constant τ. The existence of a parameter ϑ realising τ in Assumption 6 is an expressivity statement about the hypernetwork-parameterised family $\{\Phi(\cdot; h_\vartheta(\varepsilon))\}_{\varepsilon \in E}$. In practice it is obtained by a two-step bridging argument: (a) for a continuous target weight map $\varepsilon \mapsto w^\star(\varepsilon) \in \mathbb{R}^p$, the universal approximation theorem for multiple-input operators [20] delivers an MLP h_ϑ approximating $w^\star$ uniformly on $[\varepsilon_{\min}, \varepsilon_{\max}]$; (b) a Lipschitz-in-weights estimate $\|\Phi(\cdot; w) - \Phi(\cdot; \tilde{w})\|_{\mathcal{X}} \le C\,\|w - \tilde{w}\|$ on any bounded input set transfers the weight-space approximation into operator error. The MIONet theorem does not cover the hypernetwork parameterisation $(\varepsilon, \mathbf{q}) \mapsto \Phi(\mathbf{q}; h_\vartheta(\varepsilon))$ directly, so we take Assumption 6 as a hypothesis rather than as a consequence of MIONet.

Theorem 3 (Slice-wise K-step rollout stability)**.** *Under Assumptions 2–6, set $L := \max\{L_T, L_N\}$. Then for every integer $K \ge 1$,*

$$\sup_{\varepsilon \in E}\sup_{\mathbf{q} \in \mathcal{A}_\varepsilon} \left\|\mathcal{T}_\varepsilon^K(\mathbf{q}) - \mathcal{N}_\varepsilon^K(\mathbf{q})\right\|_{\mathcal{X}} \;\le\; \tau \sum_{j=0}^{K-1} L^j.$$

Two regimes follow:

- *(i) if $L \ge 1$, the bound specialises to $\tau K\,L^{K-1}$, which grows linearly in K when $L = 1$ and at geometric rate L^{K-1} when $L > 1$;*
- *(ii) if $L < 1$, the geometric series converges and the bound becomes $\tau/(1 - L)$, uniform in K, with residual decaying to zero at geometric rate L^K.*

Proof of Theorem 3. Fix $\varepsilon \in E$, $\mathbf{q} \in \mathcal{A}_\varepsilon$, and set $u_k := \mathcal{T}_\varepsilon^k(\mathbf{q})$, $v_k := \mathcal{N}_\varepsilon^k(\mathbf{q})$, $e_k := \|u_k - v_k\|_{\mathcal{X}}$. Assumption 2 ensures $u_k, v_k \in \mathcal{A}_\varepsilon$ for every $k \ge 0$, so Assumptions 4–6 apply pointwise to the

iterates. For every $k \geq 0$,

$$\begin{aligned} e_{k+1} &= \|\mathcal{T}_\varepsilon(u_k) - \mathcal{N}_\varepsilon(v_k)\|_{\mathcal{X}} \\ &\leq \|\mathcal{T}_\varepsilon(u_k) - \mathcal{T}_\varepsilon(v_k)\|_{\mathcal{X}} + \|\mathcal{T}_\varepsilon(v_k) - \mathcal{N}_\varepsilon(v_k)\|_{\mathcal{X}} \;\leq\; L_T\, e_k + \tau, \end{aligned}$$

by Assumptions 4 and 6. The symmetric split via Assumptions 5 and 6 gives $e_{k+1} \leq L_N\, e_k + \tau$; either inequality yields $e_{k+1} \leq L\, e_k + \tau$. Iterating from $e_0 = 0$, $e_K \leq \tau \sum_{j=0}^{K-1} L^j$ for every $K \geq 1$. Taking suprema over $\mathbf{q} \in \mathcal{A}_\varepsilon$ and $\varepsilon \in E$ yields the main bound.

For (i), if $L \geq 1$ then $L^j \leq L^{K-1}$ for every $0 \leq j \leq K-1$, so $\tau \sum_{j=0}^{K-1} L^j \leq \tau K\, L^{K-1}$. For (ii), if $L < 1$ the partial sum equals $\tau(1 - L^K)/(1 - L) \leq \tau/(1 - L)$, with residual $\tau L^K/(1 - L) \to 0$ as $K \to \infty$. □

Comparison with Theorem 2. The linear non-expansive special case (Theorem 2) is recovered from Theorem 1 by setting $E = \{\varepsilon\}$, $L_T = 1$ (or η), $L_N = L_T + \epsilon$, $\tau = \epsilon$, with the algebraic telescoping identity on linear operators tightening the bound by a constant factor that is unavailable in the nonlinear setting (Section B.1 above).

B.3 Slice-wise Lipschitz estimate for the DR splitting propagator

We record a concrete upper bound on the slice-wise Lipschitz constant of the DR splitting propagator of Section 3.1, obtained from a spectral-gap analysis on a working subspace where the negative Laplacian is coercive.

Lemma 1 (Slice-wise Lipschitz bound, DR splitting propagator)**.** *Let $\mathcal{S}_{\delta t,\varepsilon}$ denote the DR splitting operator of* (3)*, and suppose the state field is constrained to* $[0, 1]$*. Let $Y \subseteq L^2(\Omega_x)$ be a subspace on which the negative Laplacian has spectral gap $\lambda_1 > 0$ (e.g. $Y = L^2(\Omega_x)$ under homogeneous Dirichlet data, or the zero-mean subspace under periodic or Neumann data). Set $a := \delta t/\varepsilon^2$ and $L_\rho(a) := \sup_{s\in[0,1]} |1 - a\, W''(s)|$. Then*

$$\mathrm{Lip}\big(\mathcal{S}_{\delta t,\varepsilon}|_Y\big) \;\leq\; \frac{L_\rho(a)}{1 + \delta t\, \lambda_1}.$$

Proof. Write $\mathcal{S}_{\delta t,\varepsilon} = \rho_{1,\varepsilon} \circ (I - \delta t\Delta)^{-1}$ by (3), with $\rho_{1,\varepsilon}(s) = s - (\delta t/\varepsilon^2)\, W'(s)$.

Step 1 (reaction step). Since $\rho'_{1,\varepsilon}(s) = 1 - a\, W''(s)$, the mean value theorem gives $|\rho_{1,\varepsilon}(u(x)) - \rho_{1,\varepsilon}(v(x))| \leq L_\rho(a)\, |u(x) - v(x)|$ almost everywhere on Ω_x for u, v taking values in $[0, 1]$. Squaring and integrating yields $\|\rho_{1,\varepsilon}(u) - \rho_{1,\varepsilon}(v)\|_{L^2} \leq L_\rho(a)\, \|u - v\|_{L^2}$.

Step 2 (Helmholtz resolvent). Expand $f \in Y$ in an orthonormal eigenbasis $\{\varphi_j\}$ of $-\Delta|_Y$ with eigenvalues $\mu_j \geq \lambda_1 > 0$, $f = \sum_j c_j \varphi_j$. By the spectral theorem,

$$\|(I - \delta t\Delta)^{-1} f\|_{L^2}^2 \;=\; \sum_j |c_j|^2\, (1 + \delta t\, \mu_j)^{-2} \;\leq\; (1 + \delta t\, \lambda_1)^{-2}\, \|f\|_{L^2}^2,$$

so $\|(I - \delta t\Delta)^{-1}\|_{Y\to Y} \leq (1 + \delta t\, \lambda_1)^{-1}$. Moreover, under periodic, homogeneous Neumann, or homogeneous Dirichlet boundary conditions, $(I - \delta t\Delta)^{-1}$ is a positive contraction with $\mathbf{1}$ as a fixed point, so $u \in [0, 1]$ implies $(I - \delta t\Delta)^{-1} u \in [0, 1]$ as well; the Lipschitz constant $L_\rho(a)$ of $\rho_{1,\varepsilon}$ taken over $[0, 1]$ therefore remains the relevant bound when $\rho_{1,\varepsilon}$ is applied to $(I - \delta t\Delta)^{-1} u$.

Step 3 (composition). By sub-multiplicativity of Lipschitz constants on Y,

$$\mathrm{Lip}\big(\mathcal{S}_{\delta t,\varepsilon}|_Y\big) \;\leq\; \mathrm{Lip}(\rho_{1,\varepsilon}) \cdot \|(I - \delta t\Delta)^{-1}\|_{Y\to Y} \;\leq\; \frac{L_\rho(a)}{1 + \delta t\, \lambda_1}.$$ □

Remark (why the spectral gap is needed). On the full $L^2(\Omega_x)$ under periodic boundary data, the zero mode $\mu_0 = 0$ makes $(I - \delta t\Delta)^{-1}$ only non-expansive. Combined with $\rho'_{1,\varepsilon}(\frac{1}{2}) = 1 + \delta t/(2\varepsilon^2) > 1$ for the double-well $W(s) = \frac{1}{2} s^2 (1 - s)^2$, the Lipschitz constant on the full space exceeds one. Restriction to a spectral-gap subspace Y (homogeneous Dirichlet data, or the zero-mean subspace under periodic or Neumann data) is essential for a contraction estimate of the DR splitting propagator.

B.4 First-order equivalence of DR splitting orders

The first-order DR splitting admits two natural compositional orders. We write

$$\mathcal{S}^{(\rho K)}_{\delta t,\varepsilon}(u) \;:=\; K_1 * \rho_{1,\varepsilon}(u) \qquad \text{(reaction-then-diffusion)},$$

$$\mathcal{S}^{(K\rho)}_{\delta t,\varepsilon}(u) \;:=\; \rho_{1,\varepsilon}(K_1 * u) \qquad \text{(diffusion-then-reaction)}.$$

The first form arises from the semi-implicit Allen–Cahn step $(I - \delta t\Delta)\, u^{n+1} = \rho_{1,\varepsilon}(u^n)$ and is the form stated by Bretin et al. [2]; the second form is the order realised by the reference implementation used in this paper, and matches the closed-form update (3) as written. The following lemma shows that the two orders are first-order equivalent splittings of the Allen–Cahn semigroup, so the Lipschitz analysis of Lemma 1 and the slice-wise rollout bound of Theorem 3 apply to either choice.

Lemma 2 (First-order equivalence of DR splitting orders)**.** *Let $L_\varepsilon u := \Delta u - \varepsilon^{-2} W'(u)$ denote the Allen–Cahn generator. For every $u \in C^2(\Omega_x; [0,1])$ with bounded Laplacian and every sufficiently small $\delta t > 0$,*

$$\mathcal{S}^{(\rho K)}_{\delta t,\varepsilon}(u) \;=\; u + \delta t\, L_\varepsilon u + O(\delta t^2), \qquad \mathcal{S}^{(K\rho)}_{\delta t,\varepsilon}(u) \;=\; u + \delta t\, L_\varepsilon u + O(\delta t^2),$$

both in the $L^2(\Omega_x)$ norm, with $O(\delta t^2)$ constants depending on $\|\Delta u\|_{L^2}$, $\|W''\|_{C^0([0,1])}$, and ε but independent of δt. In particular,

$$\big\|\mathcal{S}^{(\rho K)}_{\delta t,\varepsilon}(u) - \mathcal{S}^{(K\rho)}_{\delta t,\varepsilon}(u)\big\|_{L^2(\Omega_x)} \;=\; O(\delta t^2).$$

Proof. Let $T_D := (I - \delta t\Delta)^{-1}$ and $T_R := \rho_{1,\varepsilon}$. For $u \in C^2(\Omega_x)$ with bounded Laplacian, the Helmholtz resolvent admits the Neumann-series expansion $T_D u = u + \delta t\, \Delta u + O(\delta t^2)$ in $L^2(\Omega_x)$, valid for $\delta t < \|\Delta\|^{-1}$ on the working subspace. The reaction step is exact: $T_R u = u - (\delta t/\varepsilon^2)\, W'(u)$.

Composing in the two orders and using a Taylor expansion of W' around u,

$$\begin{aligned}
\mathcal{S}^{(\rho K)}(u) \;&=\; T_D(T_R u) \;=\; T_R u + \delta t\, \Delta(T_R u) + O(\delta t^2) \\
&=\; u - (\delta t/\varepsilon^2)\, W'(u) + \delta t\, \Delta u + O(\delta t^2) \;=\; u + \delta t\, L_\varepsilon u + O(\delta t^2),
\end{aligned}$$

$$\begin{aligned}
\mathcal{S}^{(K\rho)}(u) \;&=\; T_R(T_D u) \;=\; T_D u - (\delta t/\varepsilon^2)\, W'(T_D u) \\
&=\; u + \delta t\, \Delta u - (\delta t/\varepsilon^2)\, W'\big(u + \delta t\, \Delta u + O(\delta t^2)\big) + O(\delta t^2) \\
&=\; u + \delta t\, \Delta u - (\delta t/\varepsilon^2)\, \big(W'(u) + \delta t\, W''(u)\, \Delta u + O(\delta t^2)\big) + O(\delta t^2) \\
&=\; u + \delta t\, L_\varepsilon u + O(\delta t^2),
\end{aligned}$$

where the cross-term $-(\delta t^2/\varepsilon^2)\, W''(u)\, \Delta u$ is absorbed into the $O(\delta t^2)$ remainder. Subtracting the two expansions yields the stated equivalence. □

Implication for the rollout analysis. Lemma 1 bounds the slice-wise Lipschitz constant of $\mathcal{S}_{\delta t,\varepsilon}$ on a spectral-gap subspace Y by sub-multiplicativity of Lipschitz constants under composition; this is symmetric in the order of composition, so the estimate $L_\rho(a)/(1 + \delta t\, \lambda_1)$ applies to both $\mathcal{S}^{(\rho K)}$ and $\mathcal{S}^{(K\rho)}$. Theorem 3 is stated for an abstract slice-wise propagator $\mathcal{T}_\varepsilon$ with no commitment to its internal sub-step order, and therefore applies to either splitting order verbatim.

C Implementation Details

IADR architecture. The target operator Φ in (6) is the first-order DR splitting network: a spectral convolution parameterised by a 17×17 radial kernel initialised at $\widehat{K}_1$ of (3), followed by a pointwise reaction MLP of hidden widths $[8, 3]$ with Gauss activation. The hypernetwork h_ϑ is a three-hidden-layer MLP of width 256 with GELU activations, reading ε normalised to $[-1, 1]$ and emitting directly the flattened weight vector of Φ (< 400 parameters at any ε). The MLP output is multiplied by a fixed global scalar factor of 0.1 before being injected into Φ via `torch.func.functional_call`, following the weight-injection mechanism of [33]; Φ itself carries no trainable parameters, so the only parameters of IADR are those of h_ϑ, totalling roughly 160K.

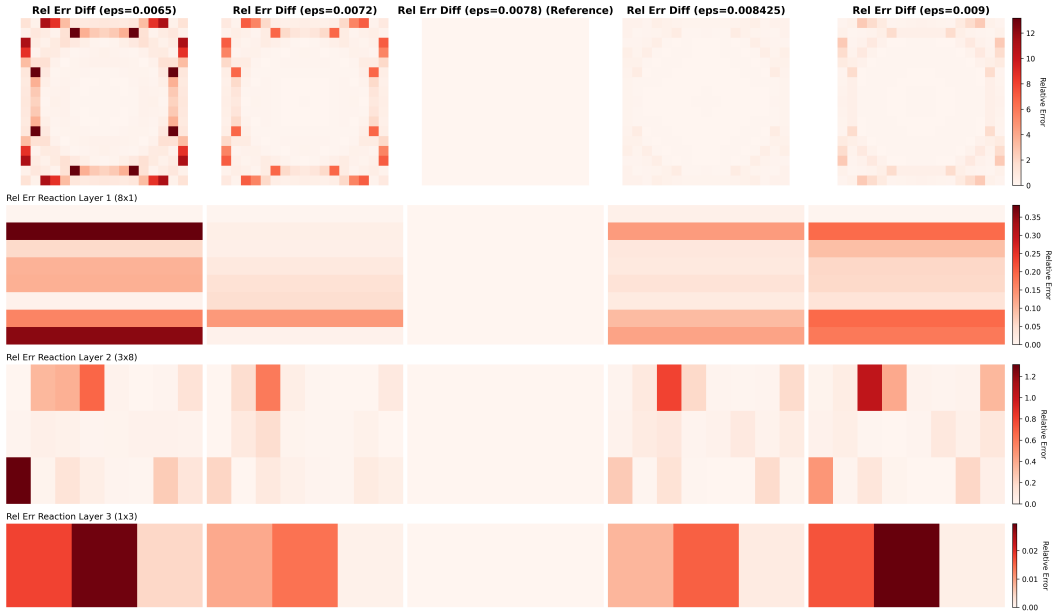


Figure 3: Per-ε DR specialists trained independently at the five training widths $\{0.0065, 0.0072, 0.0078, 0.0084, 0.0090\}$ exhibit non-trivial weight drift across ε. Row 1 shows the relative differences of the diffusion-kernel coefficient vectors; rows 2–4 show the relative differences of the three weight tensors of the reaction MLP (architecture $1{-}3{-}8{-}1$). All rows take $\varepsilon = 0.0078$ as the reference; the off-reference columns are visibly non-zero on structured patterns of coefficients, so a single shared weight vector cannot match every endpoint. We therefore parameterise the family by a learned hypernetwork $h_\vartheta(\varepsilon)$.

Baseline architectures. The U-Net variant keeps the hypernetwork pipeline of IADR (same h_ϑ, same `functional_call` injection, same global scalar damping) and replaces Φ by the four-level U-Net of [33] with base channels 8 and circular padding ($\sim$ 200K target parameters; $\sim$ 25M total once the hypernetwork emits a weight vector of comparable size). The DISCO reproduction follows the original stack: a space-time Transformer encoder on a $T = 16$ frame context, hypernetwork head, ODE-integrated single-step prediction, and DISCO's own U-Net as target; the T-frame context consists of 16 consecutive ground-truth states. At test time, the context starts with 16 ground-truth states; each autoregressive step removes the earliest state and appends the latest prediction. The Per-ε DR specialists share the Φ topology of IADR but are trained independently at each width with no hypernetwork. The Blind DR baseline trains the same Φ on the pooled five-width data without any ε input.

Empirical training objective. The population loss (5) is approximated in training by an empirical objective in which both integrals over $(\varepsilon, \mathbf{q})$ are replaced by sums over a finite training set of N_{train} pairs $\{(\varepsilon^{(n)}, R_0^{(n)}, t^{(n)})\}_{n=1}^{N_{\text{train}}}$, and the spatial $L^2(\Omega_x)$ norm by its pixel-grid Riemann sum on the $H \times W$ grid. With the multipoint unrolling of Bretin et al. [2], each sample contributes K_{mp} auto-regressive predictions $\Phi^{(k)}(\,\cdot\,; h_\vartheta(\varepsilon)) := \Phi(\,\cdot\,; h_\vartheta(\varepsilon)) \circ \cdots \circ \Phi(\,\cdot\,; h_\vartheta(\varepsilon))$ (k-fold composition) compared to the corresponding reference frames, and the discrete training objective takes the form

$$\widehat{\mathcal{L}}(\theta) \;=\; \frac{1}{N_{\text{train}}\, K_{\text{mp}}} \sum_{n=1}^{N_{\text{train}}} \sum_{k=1}^{K_{\text{mp}}} \frac{1}{HW} \sum_{i,j} \Big(\Phi^{(k)}\big(\mathbf{q}^{(n)}; h_\vartheta(\varepsilon^{(n)})\big)(i,j) - \mathbf{q}_{\varepsilon^{(n)}}\big(\cdot,\, t^{(n)} + k\delta t\big)(i,j) \Big)^2, \tag{15}$$

where the inner $\frac{1}{HW}\sum_{i,j}(\cdot)^2$ is the pixel-mean squared error implemented in the training script and approximates $\|\cdot\|^2_{L^2(\Omega_x)}$ on the unit square $\Omega_x = [0,1]^2$ up to the Riemann discretisation error of order Δx_{prod}. The configurations reported here use $K_{\text{mp}} = 1$ (single-step multipoint), with longer unrolls deferred to future runs.

Training schedule and optimiser. All trained models share the same base configuration recorded in the released YAML files: learning rate 1×10^{-4}, fixed random seed 0, and validation every five epochs. The IADR, per-ε DR, U-Net, FNO, Blind DR, Hyper-D, and Hyper-R configurations train for $20{,}000$ epochs. DISCO follows its original implementation with a learning rate of 10^{-4}, 100 training epochs, and default values for the remaining hyperparameters, without an additional large-scale task-specific hyperparameter search. The optimiser, weight decay, and any learning-rate schedule are inherited from the `nnpf` phase-field framework defaults and are released unchanged with the supplemental code package, alongside the configuration files (`DR1_DISCO.yml`, `FullDisco_A.yml`, and analogous configurations for the remaining baselines).

D Dataset and Evaluation Details

This appendix collects the dataset construction (Section D.1) and the discrete metric formulas (Section D.2) of the experimental suite. Inputs to every model are the raw phase fields $\mathbf{q}_\varepsilon \in [0, 1]$ on the 256×256 grid; the scalar ε is normalised to $[-1, 1]$ before being read by the hypernetwork (Appendix C). No further preprocessing or augmentation is applied to the trajectories.

D.1 Dataset configuration

Throughout this appendix we denote the standard training ε grid by $E_{\text{train}} = \{0.0065, 0.0072, 0.0078, 0.0084, 0.0090\}$.

Geometry parameters of the three datasets used in Section 4.1 are described in the three paragraphs below; their ε values, sample counts, and trajectory frame counts are listed in Table 3.

Circles. Each circular initial set is a disk centred near $(0.5, 0.5)$. *(Train / validation.)* Initial radii are drawn from $R_0 \sim \mathcal{U}[0.05, 0.45]$ with the centre fixed at $(0.5, 0.5)$. *(Test.)* For every ε value we draw 100 independent samples with $R_0 \sim \mathcal{U}[0.25, 0.35]$ and centre $(c_x, c_y) \sim \mathcal{U}[0.47, 0.53]^2$, so that bootstrap confidence intervals on circle test cells aggregate over 100 trajectories rather than a single template.

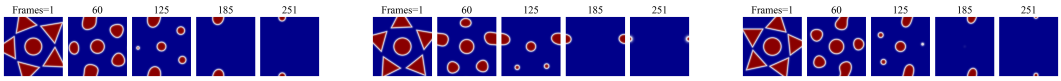
Circle (DISCO variant). DISCO consumes a 16-frame context per sample (the `n_past` $= 16$ setting of Morel et al. [33]), so for the DISCO baseline alone we reshape the circle dataset to provide this context: the train tensor stores 16 consecutive ground-truth past frames followed by one target next frame (giving the second dimension $T = 17$), while the validation tensor stores 16 past frames followed by six future targets used for the rollout error metric (giving $T = 22$). The remaining axes ($N_\varepsilon = 5$ training widths, $N_{\text{sample}} = 100$ trajectories per width, single channel, 256×256 resolution) and the sampling distribution of R_0 are identical to the standard Circle dataset; the test split is reused unchanged.

Spiral disks. Each spiral-disk initial set starts from a disk of base radius $R \in [0.28, 0.32]$ and is deformed by a spiral-ripple displacement $a \sin(k_r\, r + k_\theta\, \theta)$ in polar coordinates, with ripple amplitude $a \in [0.06, 0.10]$, integer radial frequency $k_r \in [9, 11]$, and fixed angular frequency $k_\theta = 5$. The deformed shape is then rotated by a global angle drawn uniformly from $[0, 2\pi]$ and translated to the fixed centre $(0.5, 0.5)$.

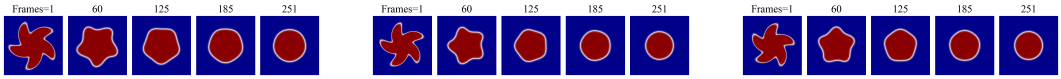
Five-pointed star. Each five-pointed-star initial set is the union of a central disk of radius $\in [0.10, 0.14]$ and five regular triangular arms (each of outer radius $\in [0.18, 0.22]$, placed at distance $\in [0.37, 0.43]$ from the centre, and rotated to the five evenly spaced angular positions $2\pi i/5$, $i = 0, \ldots, 4$). The whole figure is then rotated by a global angle drawn uniformly from $[0, 2\pi]$ and translated to the fixed centre $(0.5, 0.5)$.

D.2 Evaluation metrics

Notation. The formulas below define each metric as the trajectory-mean (over N_{total} samples) of a worst-frame quantity per trajectory; we mark this trajectory mean by an overbar, e.g., $\overline{L^2_{T^\infty}}$, $\overline{V_{T^\infty}}$, $\overline{\text{AIW}}$. The unbarred shorthands L^2-T^∞, V-T^∞, and AIW used in the main text and in every results table refer to these same quantities.



(a) Five-pointed star (`fs`) initial conditions.



(b) Spiral disk (`sd`) initial conditions.

Figure 4: Sample diversity in the held-out test sets at $\varepsilon = 0.0078$. Three independently sampled initial conditions are shown for each shape family. Samples differ in global rotation, ripple parameters (spiral disk), and arm placement (five-pointed star), so each held-out evaluation aggregates over a non-trivial geometric distribution rather than a single template.

Table 3: Dataset configuration. Each *Size* entry is a 6-tuple $(N_\varepsilon, T, N_{\text{sample}}, 1, H, W)$, where N_ε is the number of sampled ε values, T is the total number of time frames per sample, N_{sample} is the number of samples per ε, the fourth entry 1 is the single channel, and $H \times W$ is the spatial resolution.

Dataset	Train set		Validation set		Test set	
	ε	Size	ε	Size	ε	Size
Circle	E_{train}	$(5, 2, 100, 1, 256, 256)$	E_{train}	$(5, 6, 200, 1, 256, 256)$	$E_{\text{train}} \cup \{0.0069, 0.0081, 0.0093\}$	$(8, 700, 100, 1, 256, 256)$
Circle (DISCO variant)	E_{train}	$(5, 17, 100, 1, 256, 256)$	E_{train}	$(5, 22, 100, 1, 256, 256)$	shared with the standard Circle test set	
Spiral disks	—	—	—	—	0.0069, 0.0081, 0.0093	$(3, 251, 100, 1, 256, 256)$
Five-pointed star	—	—	—	—	0.0069, 0.0081, 0.0093	$(3, 251, 100, 1, 256, 256)$

Let $\hat{u}_{t'}^{(n)}, u_{t'}^{(n)} \in \mathbb{R}^{H \times W}$ denote the predicted and reference phase fields for sample n at time-step index t', on the $H \times W$ grid with physical cell area $\Delta x_{\text{prod}} = \Delta x\, \Delta y$. The evaluation window is $t' \in [t_{\text{start}} + 1,\, t_{\text{end}})$, giving $T_{\text{eval}} = t_{\text{end}} - t_{\text{start}} - 1$ one-step predictions per sample across N_{total} samples.

L^2-T^∞ error. The per-frame $L^2(\Omega_x)$ error is computed with the physical cell-area weight,

$$\left\|\hat{u}_{t'}^{(n)} - u_{t'}^{(n)}\right\|_{L^2(\Omega_x)} = \sqrt{\Delta x_{\text{prod}} \sum_{i,j} \left(\hat{u}_{t'}^{(n)}(i,j) - u_{t'}^{(n)}(i,j)\right)^2}.$$

Taking the worst frame per trajectory and then the sample mean,

$$\overline{L^2_{T^\infty}} = \frac{1}{N_{\text{total}}} \sum_{n=1}^{N_{\text{total}}} \max_{t'} \left\|\hat{u}_{t'}^{(n)} - u_{t'}^{(n)}\right\|_{L^2(\Omega_x)}. \tag{16}$$

Volume-T^∞ error. The discrete mass of a phase field u is

$$M[u] = \Delta x_{\text{prod}} \sum_{i,j} u(i,j) \approx \int_{\Omega_x} u(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

with per-frame volume error $\left| M[\hat{u}_{t'}^{(n)}] - M[u_{t'}^{(n)}] \right|$. The reported scalar is the worst-frame mass deviation, averaged over samples,

$$\overline{V_{T^\infty}} = \frac{1}{N_{\text{total}}} \sum_{n=1}^{N_{\text{total}}} \max_{t'} \left| M[\hat{u}_{t'}^{(n)}] - M[u_{t'}^{(n)}] \right|. \tag{17}$$

Under mean-curvature flow, $M[u_\varepsilon(\cdot, t)]$ decays at the theoretical rate $A'(t) = -2\pi$ for a single convex curve, so this metric measures the worst-frame violation of that law.

AIW absolute error. The *average interface width* (AIW) of a phase field u is defined via the discrete coarea formula. Using central differences with replicate boundary padding,

$$\partial_x u(i,j) = \frac{u(i, j{+}1) - u(i, j{-}1)}{2\,\Delta x}, \quad \partial_y u(i,j) = \frac{u(i{+}1, j) - u(i{-}1, j)}{2\,\Delta y},$$

Table 4: Mapping between datasets and tasks. ✓ indicates that the dataset is used for the corresponding split or task; × indicates it is not.

Dataset	Train/Val	Task 1	Task 2	Task 3	Task 4
Circle	✓	✓	✓	×	×
Spiral disks	×	×	×	✓	✓
Five-pointed star	×	×	×	✓	✓

we form the regularised gradient magnitude

$$\|\nabla u\|(i,j) \;=\; \sqrt{(\partial_x u)^2 + (\partial_y u)^2 + \varepsilon_0}, \qquad \varepsilon_0 = 10^{-30}.$$

With fixed thresholds $\ell = 0.02$ and $h = 0.98$, define the interface-band indicator $\mathbf{1}_{\text{band}}(i,j) = \mathbf{1}[\ell < u(i,j) < h]$ and the band area and band gradient integral

$$A_{\text{band}}[u] = \Delta x_{\text{prod}} \sum_{i,j} \mathbf{1}_{\text{band}}(i,j), \quad G_{\text{band}}[u] = \Delta x_{\text{prod}} \sum_{i,j} \mathbf{1}_{\text{band}}(i,j)\, \|\nabla u\|(i,j).$$

The discrete coarea formula yields the interface length estimate $L_{\text{est}}[u] = G_{\text{band}}[u]/(h - \ell + \varepsilon_0)$, and the AIW is the band area divided by this length,

$$\text{AIW}[u] \;=\; \frac{A_{\text{band}}[u]}{L_{\text{est}}[u] + \varepsilon_0}.$$

For a well-prepared $\mathbf{q}_\varepsilon$ profile this equals the theoretical interface width, proportional to ε up to a constant determined by (ℓ, h). Fixing a late-time frame index $t^\star$ (default $t_{\text{start}} + \lfloor 0.8\,(T_{\text{eval}} - 1)\rfloor$), the per-sample single-frame absolute error is $\big|\text{AIW}[\hat{u}_{t^\star}^{(n)}] - \text{AIW}[u_{t^\star}^{(n)}]\big|$, and the reported scalar is its sample mean,

$$\overline{\text{AIW}} \;=\; \frac{1}{N_{\text{total}}} \sum_{n=1}^{N_{\text{total}}} \big|\text{AIW}[\hat{u}_{t^\star}^{(n)}] - \text{AIW}[u_{t^\star}^{(n)}]\big|. \tag{18}$$

Sample bootstrap. Tables 1, 5, and 8 report bootstrap means, with $\mathrm{boot_std}$ additionally shown in Tables 5 and 8. The 95% percentile CIs are computed but not displayed in these tables. These statistics are computed by non-parametric bootstrap over the N_{total} per-trajectory metric values $\{X^{(n)}\}_{n=1}^{N_{\text{total}}}$ (with X one of $\overline{L^2_{T^\infty}}$, $\overline{V_{T^\infty}}$, $\overline{\text{AIW}}$). For $b = 1, \ldots, B$ we draw indices $i_{b,1}, \ldots, i_{b,N_{\text{total}}}$ i.i.d. uniformly from $\{1, \ldots, N_{\text{total}}\}$ with replacement and form the resampled mean

$$\bar{X}^{(b)} \;=\; \frac{1}{N_{\text{total}}} \sum_{k=1}^{N_{\text{total}}} X^{(i_{b,k})}.$$

The reported point estimate is $\frac{1}{B}\sum_{b=1}^{B} \bar{X}^{(b)}$, the standard deviation is $\mathrm{boot_std} = \mathrm{std}(\{\bar{X}^{(b)}\}_{b=1}^{B})$ with Bessel correction, and the 95% CI is the percentile interval $[\hat{q}_{2.5\%},\ \hat{q}_{97.5\%}]$ of the resampled-mean distribution. We use $B = 2000$ resamples and a fixed RNG seed of 42 for every $(\text{model}, \text{shape}, \varepsilon)$ cell, so the bootstrap statistics are reproducible from the released per-sample metric files.

E Additional Results

Additional case study: spiral disk geometry-OOD. Figure 5 provides an additional geometry-OOD case: spiral disk at $\varepsilon = 0.0078$. IADR, per-ε DR, and ground truth follow the same evolution from spiral arms to a rounded disk, while the U-Net target rounds the spiral too quickly and overshoots the contracting disk size at later frames.

Full main-table numbers. Table 5 reproduces Table 1 of the main text with the full $\mathrm{mean} \pm \mathrm{std}$. Column scale factors and row grouping are identical; the table is set in landscape to fit the wider mean ± std cells.

Table 5: Main results, full $\mathrm{mean} \pm \mathrm{std}$. Same row grouping, column scale factors, and per-ε DR matching as Table 1; smallest mean per (dataset, metric) row is set in bold; second smallest is underlined. The DISCO entry at $\varepsilon = 0.0065$ is unavailable.

		L^2-T^∞ ($\times 10^{-3}$)					V-T^∞ ($\times 10^{-3}$)					AIW ($\times 10^{-3}$)				
Shape	ε	IADR	per-ε DR	U-Net	FNO	DISCO	IADR	per-ε DR	U-Net	FNO	DISCO	IADR	per-ε DR	U-Net	FNO	DISCO
Task 1: circles at training widths (in-distribution)																
Circle	0.0065	33.98 ± 1.21	$\mathbf{33.75 \pm 1.25}$	921.67 ± 0.13	518.77 ± 5.40	diverged	3.81 ± 0.17	$\mathbf{3.79 \pm 0.16}$	680.39 ± 0.22	261.42 ± 6.42	diverged	$\mathbf{51.84 \pm 49.53}$	51.87 ± 49.53	69.46 ± 48.00	76.61 ± 46.77	diverged
	0.0072	27.06 ± 1.31	$\mathbf{18.89 \pm 0.80}$	951.33 ± 0.60	510.70 ± 6.52	$6.7\times10^{13} \pm 6.6\times10^{13}$	2.97 ± 0.10	$\mathbf{2.09 \pm 0.12}$	709.87 ± 0.40	269.69 ± 6.03	$7.2\times10^{12} \pm 7.2\times10^{12}$	0.34 ± 0.07	$\mathbf{0.32 \pm 0.06}$	25.62 ± 9.18	24.51 ± 3.49	398.57 ± 36.02
	0.0078	22.35 ± 1.43	$\mathbf{6.93 \pm 0.33}$	959.88 ± 0.73	508.08 ± 5.28	$4.7\times10^{13} \pm 4.6\times10^{13}$	2.37 ± 0.03	$\mathbf{0.92 \pm 0.08}$	728.62 ± 0.83	269.20 ± 6.75	$5.7\times10^{12} \pm 5.6\times10^{12}$	0.48 ± 0.10	$\mathbf{0.37 \pm 0.05}$	17.45 ± 1.56	20.81 ± 3.53	411.65 ± 37.82
	0.0084	14.78 ± 1.45	$\mathbf{4.53 \pm 0.36}$	974.31 ± 0.91	508.02 ± 5.24	$2.5\times10^{12} \pm 2.4\times10^{12}$	1.39 ± 0.08	$\mathbf{0.94 \pm 0.09}$	745.59 ± 1.50	264.67 ± 6.36	$2.3\times10^{11} \pm 2.2\times10^{11}$	0.54 ± 0.19	$\mathbf{0.35 \pm 0.05}$	23.25 ± 1.21	16.86 ± 3.49	419.19 ± 37.11
	0.0090	11.80 ± 1.27	$\mathbf{7.36 \pm 0.69}$	984.91 ± 1.19	509.65 ± 5.04	$1.6\times10^{18} \pm 1.6\times10^{18}$	1.18 ± 0.07	$\mathbf{0.84 \pm 0.04}$	755.55 ± 2.15	268.61 ± 5.68	$3.5\times10^{17} \pm 3.4\times10^{17}$	1.00 ± 0.39	$\mathbf{0.50 \pm 0.07}$	27.56 ± 0.98	14.84 ± 3.41	409.46 ± 34.45
Task 2: circles at non-training widths (ε-OOD)																
Circle	0.0069	28.08 ± 1.22	$\mathbf{25.46 \pm 1.04}$	937.71 ± 0.61	508.98 ± 5.66	$1.1\times10^{10} \pm 5.9\times10^{9}$	3.08 ± 0.12	$\mathbf{2.82 \pm 0.13}$	698.39 ± 0.29	265.67 ± 5.89	$8.7\times10^{8} \pm 4.8\times10^{8}$	$\mathbf{0.37 \pm 0.08}$	0.41 ± 0.07	24.17 ± 9.99	29.90 ± 2.89	396.72 ± 34.43
	0.0081	19.99 ± 1.49	$\mathbf{9.24 \pm 0.40}$	968.03 ± 0.90	500.88 ± 4.97	$7.1\times10^{9} \pm 3.6\times10^{9}$	2.07 ± 0.03	$\mathbf{1.16 \pm 0.10}$	737.03 ± 1.13	265.36 ± 5.98	$7.1\times10^{8} \pm 3.1\times10^{8}$	0.64 ± 0.18	$\mathbf{0.50 \pm 0.08}$	20.79 ± 1.36	20.59 ± 3.59	416.88 ± 38.23
	0.0093	65.95 ± 2.25	$\mathbf{8.27 \pm 0.75}$	778.98 ± 40.87	510.99 ± 5.97	$3.1\times10^{11} \pm 1.6\times10^{11}$	9.97 ± 0.16	$\mathbf{0.95 \pm 0.04}$	549.11 ± 40.59	269.45 ± 6.06	$4.8\times10^{10} \pm 2.6\times10^{10}$	16.76 ± 10.10	$\mathbf{0.85 \pm 0.34}$	40.81 ± 10.95	12.11 ± 3.14	410.05 ± 35.86
Task 3: non-circular geometries at training widths (geometry-OOD)																
Five-pointed star	0.0078	21.85 ± 4.76	$\mathbf{8.80 \pm 5.11}$	757.35 ± 1.92	553.48 ± 4.15	406.12 ± 2.90	3.77 ± 1.09	$\mathbf{1.62 \pm 1.19}$	603.77 ± 2.79	229.99 ± 6.80	224.92 ± 2.43	1.98 ± 1.78	$\mathbf{1.87 \pm 1.78}$	11.52 ± 2.66	19.34 ± 3.24	184.83 ± 3.96
Spiral disk	0.0078	4.03 ± 0.01	$\mathbf{3.65 \pm 0.05}$	119.04 ± 3.41	382.58 ± 6.19	205.19 ± 1.89	1.08 ± 0.01	$\mathbf{0.96 \pm 0.02}$	33.76 ± 1.17	86.77 ± 3.45	16.24 ± 0.07	0.44 ± 0.01	0.35 ± 0.01	$\mathbf{0.23 \pm 0.02}$	18.04 ± 1.16	295.82 ± 4.42
Task 4: non-circular geometries at non-training widths (joint OOD)																
Five-pointed star	0.0069	16.73 ± 5.10	$\mathbf{13.35 \pm 4.99}$	720.87 ± 1.69	556.31 ± 4.54	407.88 ± 2.87	2.81 ± 1.13	$\mathbf{2.25 \pm 1.07}$	547.74 ± 2.43	231.35 ± 7.10	222.74 ± 1.73	$\mathbf{1.70 \pm 1.57}$	1.71 ± 1.58	7.95 ± 1.74	20.80 ± 2.77	180.18 ± 3.38
	0.0093	50.26 ± 5.61	$\mathbf{28.34 \pm 6.64}$	790.93 ± 1.14	550.19 ± 3.96	393.26 ± 3.05	10.75 ± 1.15	$\mathbf{5.23 \pm 1.56}$	659.78 ± 1.78	229.31 ± 6.36	222.89 ± 2.39	$\mathbf{2.33 \pm 2.08}$	2.38 ± 2.07	9.06 ± 2.60	13.43 ± 3.15	186.33 ± 3.37
Spiral disk	0.0069	7.71 ± 0.04	$\mathbf{7.59 \pm 0.05}$	112.61 ± 2.53	373.37 ± 6.56	213.23 ± 1.59	1.97 ± 0.02	$\mathbf{1.95 \pm 0.02}$	27.20 ± 0.92	82.75 ± 3.61	16.06 ± 0.07	0.34 ± 0.02	0.36 ± 0.02	$\mathbf{0.19 \pm 0.02}$	24.03 ± 1.58	286.82 ± 3.70
	0.0093	14.16 ± 0.03	$\mathbf{3.72 \pm 0.04}$	170.10 ± 5.60	382.93 ± 7.89	205.85 ± 1.56	4.24 ± 0.01	$\mathbf{0.64 \pm 0.02}$	58.13 ± 2.23	88.09 ± 3.59	16.00 ± 0.08	0.56 ± 0.02	0.56 ± 0.01	$\mathbf{0.49 \pm 0.03}$	8.02 ± 0.97	293.18 ± 3.48

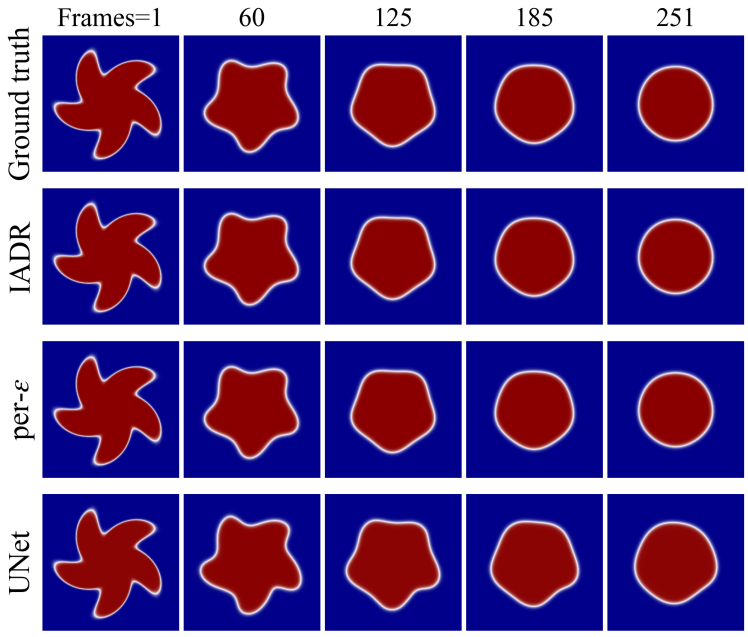


Figure 5: Geometry-OOD spiral disk at $\varepsilon = 0.0078$. Columns are frames $\{1, 60, 125, 185, 251\}$; rows are ground truth, IADR, the per-ε DR specialist, and U-Net.

Wider range. We expand the training interval to $[0.0036, 0.0120]$, approximately 3.3 times the original span, and retain five evenly spaced training widths, including both endpoints. We evaluate circles, five-pointed stars, and spiral disks at $\varepsilon = 0.0047$ (unseen interpolation width), 0.0078 (seen midpoint), and 0.0120 (seen endpoint). Table 6 reports the complete comparison with the corresponding per-ε DR specialists.

At $\varepsilon = 0.0120$, IADR has lower errors than the specialists in all nine geometry–metric combinations. Across the nine AIW comparisons in the wider range, IADR has lower error in five. These results show that the original range is not an architectural limit of IADR. The circle AIW errors are higher at the seen endpoint and unseen interpolation width than at the midpoint. The range should be chosen to match the spatial resolution. The experiment supports single-model coverage of a wider predefined, numerically resolvable range included in training, rather than unrestricted extrapolation beyond that range.

Dumbbell decomposition. The initial configuration consists of two circular lobes joined by a narrow rectangular bridge. During evolution, the bridge breaks and the region separates into multiple connected components. We evaluate the originally trained IADR and per-ε DR specialists at $\varepsilon = 0.0069$ (unseen interpolation width), 0.0078 (interval midpoint), and 0.0090 (interval endpoint), without retraining. The results in Table 7 indicate that IADR retains geometric evolution capability comparable to that of the per-ε DR specialists for this simple topological change. The AIW errors further show that IADR continues to control the phase-field interface width.

F Ablation Results

This appendix reports the per-cell numbers for the Blind DR, Hyper-D and Hyper-R ablations: variants of IADR that remove or restrict ε-conditioning while keeping the DR splitting target fixed. **Blind DR** replaces the hypernetwork with a single ε-agnostic weight vector trained on the pooled five-width data. **Hyper-D** and **Hyper-R** restrict the hypernetwork to emit only the diffusion-block weights $w_D \in \mathbb{R}^{K_D^2}$ or only the reaction-block weights $w_R \in \mathbb{R}^{p_R}$ respectively, with the non-emitted block taking a single ε-agnostic vector jointly trained with h_ϑ. On circles, we also compare Direct-ε-DR, which uses parameter concatenation, at $\varepsilon = 0.0069$, 0.0078 and 0.0090, and direct interpolation of independently trained specialist weights (without further training) at the unseen widths 0.0069 and 0.0081. Tables 9 and 10 report these results; IADR has lower errors on all three metrics in these

Table 6: Wider-range results. Best result per row in bold.

Shape	ε	L^2-T^∞ ($\times 10^{-3}$)		V-T^∞ ($\times 10^{-3}$)		AIW ($\times 10^{-3}$)	
		IADR	per-ε DR	IADR	per-ε DR	IADR	per-ε DR
Task 1: circles at training widths (in-distribution)							
Circle	0.0078	33.17	**7.15**	3.81	**0.71**	0.75	**0.37**
	0.0120	**24.46**	39.72	**6.50**	10.87	**1.91**	7.35
Task 2: circles at non-training widths (ε-OOD)							
Circle	0.0047	96.15	**86.47**	23.22	**12.19**	7.36	**4.55**
Task 3: non-circular geometries at training widths (geometry-OOD)							
Five-pointed star	0.0078	46.81	**12.10**	8.48	**2.44**	1.93	**1.90**
	0.0120	**40.01**	73.28	**7.86**	16.46	**0.54**	0.76
Spiral disk	0.0078	6.48	**2.28**	0.58	**0.26**	**0.11**	0.30
	0.0120	**14.49**	26.61	**5.02**	9.13	**0.85**	1.34
Task 4: non-circular geometries at non-training widths (joint OOD)							
Five-pointed star	0.0047	141.55	**86.73**	33.30	**15.75**	**2.99**	3.26
Spiral disk	0.0047	128.90	**23.39**	29.96	**4.97**	1.39	**0.20**

Table 7: Dumbbell decomposition using the originally trained models.

ε	L^2-T^∞ ($\times 10^{-2}$)		V-T^∞ ($\times 10^{-2}$)		AIW ($\times 10^{-3}$)	
	IADR	Specialist	IADR	Specialist	IADR	Specialist
0.0069	6.548	**3.801**	1.048	**0.5869**	0.8420	**0.4394**
0.0078	10.47	**7.542**	2.167	**1.696**	2.363	**2.198**
0.0090	10.52	**8.329**	2.060	**1.634**	2.233	**2.052**

comparisons. Row grouping and column scale factors mirror Table 1; smallest mean per (dataset, metric) row across the four ablation columns is set in bold; second smallest is underlined.

Although Hyper-R appears competitive on most cells of Table 8, its AIW at $\varepsilon = 0.0072$ rises to 72.38×10^{-3}, two orders of magnitude above IADR's 0.34×10^{-3} and even higher than Blind DR (36.84) and Hyper-D (5.33) at the same cell. The width $\varepsilon = 0.0072$ is neither the boundary of the training interval ($\varepsilon = 0.0065$) nor an out-of-range value ($\varepsilon = 0.0093$); it is a width the model was trained on, in the interior of the training set. This result shows the value of joint diffusion and reaction conditioning for interface-width consistency and, together with the Direct-ε-DR and specialist weight interpolation comparisons in Tables 9 and 10, supports full DR conditioning.

G Reproducibility

Code release. A lightweight, runnable anonymised code package covering the IADR architecture, the per-ε DR specialist baseline, the dataset generation script, and the training and evaluation entry points is bundled with the supplemental material of this submission. A public GitHub release with the full experimental codebase, including all baselines and the bootstrap evaluation pipeline, will be provided upon acceptance.

Random seeds. All training runs reported in Table 1 and Table 8 use a single fixed random seed per model; seed values and the seed-dependent components (parameter initialisation, mini-batch sampling, data-augmentation noise) are recorded in the configuration files released with the code. Reported metric variability comes from a non-parametric trajectory bootstrap ($B = 2000$) at evaluation time rather than from training-seed resampling; the bootstrap procedure and its assumptions are described in Appendix D.2.

Runtime and hardware. All training was carried out on a single NVIDIA V100 GPU with 100 epochs for DISCO and 20,000 epochs for all other models. Per-run wall-clock times are: 16 h for

Table 8: Blind DR / Hyper-D / Hyper-R ablations, full $\mathrm{mean} \pm \mathrm{std}$ at the same (shape, ε) cells as Table 1.

Shape	ε	L^2-T^∞ ($\times 10^{-3}$)				V-T^∞ ($\times 10^{-3}$)				AIW ($\times 10^{-3}$)			
		IADR	Blind DR	Hyper-D	Hyper-R	IADR	Blind DR	Hyper-D	Hyper-R	IADR	Blind DR	Hyper-D	Hyper-R
Task 1: circles at training widths (in-distribution)													
Circle	0.0065	$\underline{33.98} \pm \underline{1.21}$	322.62 ± 6.85	177.20 ± 3.78	$\mathbf{10.22 \pm 0.86}$	$\underline{3.81} \pm \underline{0.17}$	114.81 ± 2.49	40.18 ± 0.70	$\mathbf{1.03 \pm 0.06}$	$\underline{51.84} \pm \underline{49.53}$	82.20 ± 48.14	53.23 ± 48.15	$\mathbf{51.81 \pm 49.53}$
	0.0072	$\underline{27.06} \pm \underline{1.31}$	324.67 ± 6.64	87.07 ± 5.72	$\mathbf{18.66 \pm 0.74}$	$\underline{2.97} \pm \underline{0.10}$	117.41 ± 2.50	13.77 ± 0.70	$\mathbf{2.25 \pm 0.12}$	$\mathbf{0.34 \pm 0.07}$	36.84 ± 0.96	$\underline{5.33} \pm \underline{2.79}$	72.38 ± 69.27
	0.0078	$\mathbf{22.35 \pm 1.43}$	327.27 ± 6.66	65.00 ± 1.12	$\underline{33.13} \pm \underline{1.50}$	$\mathbf{2.37 \pm 0.03}$	120.55 ± 2.61	9.47 ± 0.58	$\underline{4.13} \pm \underline{0.11}$	$\mathbf{0.48 \pm 0.10}$	41.32 ± 1.20	18.65 ± 18.13	$\underline{7.85} \pm \underline{7.12}$
	0.0084	$\mathbf{14.78 \pm 1.45}$	341.33 ± 6.78	158.59 ± 1.67	$\underline{48.62} \pm \underline{1.94}$	$\mathbf{1.39 \pm 0.08}$	133.54 ± 2.80	33.21 ± 0.80	$\underline{6.76} \pm \underline{0.15}$	$\mathbf{0.54 \pm 0.19}$	46.01 ± 1.44	22.55 ± 8.73	$\underline{3.36} \pm \underline{2.66}$
	0.0090	$\mathbf{11.80 \pm 1.27}$	360.34 ± 6.02	205.08 ± 2.21	$\underline{54.13} \pm \underline{2.13}$	$\mathbf{1.18 \pm 0.07}$	150.91 ± 2.48	51.18 ± 1.17	$\underline{7.95} \pm \underline{0.15}$	$\mathbf{1.00 \pm 0.39}$	50.29 ± 1.70	46.32 ± 15.83	$\underline{3.53} \pm \underline{2.63}$
Task 2: circles at non-training widths (ε-OOD)													
Circle	0.0069	$\mathbf{28.08 \pm 1.22}$	323.69 ± 6.76	131.63 ± 4.96	$\underline{36.79} \pm \underline{1.16}$	$\mathbf{3.08 \pm 0.12}$	116.35 ± 2.57	25.37 ± 0.72	$\underline{4.58} \pm \underline{0.22}$	$\mathbf{0.37 \pm 0.08}$	34.65 ± 0.84	5.02 ± 2.59	$\underline{2.40} \pm \underline{2.00}$
	0.0081	$\mathbf{19.99 \pm 1.49}$	331.30 ± 6.60	115.58 ± 1.44	$\underline{48.97} \pm \underline{1.83}$	$\mathbf{2.07 \pm 0.03}$	124.22 ± 2.69	20.34 ± 0.62	$\underline{6.75} \pm \underline{0.17}$	$\mathbf{0.64 \pm 0.18}$	43.61 ± 1.32	11.65 ± 4.15	$\underline{3.03} \pm \underline{2.32}$
	0.0093	$\underline{65.95} \pm \underline{2.25}$	366.49 ± 5.91	212.02 ± 2.26	$\mathbf{39.37 \pm 2.08}$	$\underline{9.97} \pm \underline{0.16}$	156.89 ± 2.62	54.44 ± 1.22	$\mathbf{5.37 \pm 0.06}$	$\underline{16.76} \pm \underline{10.10}$	52.66 ± 1.84	51.79 ± 18.82	$\mathbf{12.11 \pm 10.64}$
Task 3: non-circular geometries at training widths (geometry-OOD)													
Five-pointed star	0.0078	21.85 ± 4.76	327.30 ± 3.90	$\underline{16.95} \pm \underline{4.77}$	$\mathbf{12.33 \pm 5.12}$	3.77 ± 1.09	145.59 ± 2.67	$\underline{2.95} \pm \underline{1.11}$	$\mathbf{2.75 \pm 1.21}$	$\underline{1.98} \pm \underline{1.78}$	38.82 ± 1.09	$\mathbf{1.92 \pm 1.79}$	2.04 ± 1.77
Spiral disk	0.0078	$\mathbf{4.03 \pm 0.01}$	331.22 ± 2.63	26.29 ± 0.22	$\underline{7.00} \pm \underline{0.02}$	$\mathbf{1.08 \pm 0.01}$	14.53 ± 0.65	7.19 ± 0.09	$\underline{1.99} \pm \underline{0.01}$	$\underline{0.44} \pm \underline{0.01}$	41.47 ± 0.03	$\mathbf{0.29 \pm 0.01}$	0.44 ± 0.02
Task 4: non-circular geometries at non-training widths (joint OOD)													
Five-pointed star	0.0069	$\underline{16.73} \pm \underline{5.10}$	325.14 ± 3.81	163.30 ± 1.74	$\mathbf{15.52 \pm 5.33}$	$\underline{2.81} \pm \underline{1.13}$	141.47 ± 2.54	43.35 ± 0.60	$\mathbf{2.07 \pm 1.24}$	$\mathbf{1.70 \pm 1.57}$	33.16 ± 0.54	4.76 ± 2.24	$\underline{1.74} \pm \underline{1.57}$
	0.0093	$\underline{50.26} \pm \underline{5.61}$	348.38 ± 3.65	177.55 ± 2.80	$\mathbf{24.82 \pm 5.34}$	$\underline{10.75} \pm \underline{1.15}$	166.93 ± 2.71	54.46 ± 1.41	$\mathbf{5.70 \pm 1.42}$	$\mathbf{2.33 \pm 2.08}$	51.70 ± 1.30	270.10 ± 136.07	$\underline{2.75} \pm \underline{2.05}$
Spiral disk	0.0069	$\mathbf{7.71 \pm 0.04}$	332.76 ± 2.19	46.14 ± 0.47	$\underline{10.74} \pm \underline{0.07}$	$\mathbf{1.97 \pm 0.02}$	19.20 ± 0.70	3.23 ± 0.15	$\underline{2.86} \pm \underline{0.03}$	$\underline{0.34} \pm \underline{0.02}$	34.49 ± 0.02	$\mathbf{0.23 \pm 0.02}$	0.39 ± 0.02
	0.0093	$\underline{14.16} \pm \underline{0.03}$	334.50 ± 2.03	73.80 ± 0.46	$\mathbf{6.29 \pm 0.02}$	$\underline{4.24} \pm \underline{0.01}$	14.02 ± 0.81	21.17 ± 0.06	$\mathbf{1.94 \pm 0.00}$	$\mathbf{0.56 \pm 0.02}$	53.15 ± 0.04	2.00 ± 0.02	$\underline{0.94} \pm \underline{0.01}$

Table 9: Comparison with Direct-ε-DR on circles. All errors are in units of $\times 10^{-3}$.

ε	Model	L^2-T^∞	V-T^∞	AIW
0.0069	IADR	**28.08**	**3.08**	**0.37**
	Direct-ε-DR	34.07	4.29	2.06
0.0078	IADR	**22.35**	**2.37**	**0.48**
	Direct-ε-DR	33.50	5.37	8.03
0.0090	IADR	**11.80**	**1.18**	**1.00**
	Direct-ε-DR	47.24	7.74	13.92

Table 10: Comparison with specialist weight interpolation on circles at the unseen widths $\varepsilon = 0.0069$ and 0.0081. All errors are in units of $\times 10^{-3}$.

ε	Model	L^2-T^∞	V-T^∞	AIW
0.0069	IADR	**28.08**	**3.08**	**0.37**
	Specialist weight interpolation	68.71	9.56	23.71
0.0081	IADR	**19.99**	**2.07**	**0.64**
	Specialist weight interpolation	91.85	15.48	78.35

IADR, 3 h for each of the eight per-ε DR specialists, 60 h for U-Net, 65 h for FNO, 72 h for DISCO, and 18 h, 15 h, 14 h respectively for the Hyper-D, Hyper-R, and Blind DR ablations, summing to roughly 284 GPU-hours for the training runs listed here. Single-step inference timings on a 256×256 grid are reported in Table 2.

Reference solver. Ground-truth trajectories on circles use the closed-form law $R(t) = \sqrt{R_0^2 - 2t}$. Ground-truth trajectories on non-circular geometries (spiral disks and five-pointed stars) are produced by the higher-order DR splitting solver of Appendix A.3 at the same time step $\delta t = 6.10 \times 10^{-5}$ as the trained models. The solver configuration and dataset-construction details are in Appendix D.1.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The three contributions listed in Section 1 match the slice-wise rollout result stated as Theorem 1 in Section 3.3, with its formal version and complete proof given as Theorem 3 in Appendix B.2 and a tighter linear non-expansive special case (Theorem 2) in Appendix B.1, together with the experimental results reported in Section 4. The scope is limited to two-dimensional phase-field mean-curvature flow over the interface-width range $[\varepsilon_{\min}, \varepsilon_{\max}]$ introduced in Section 3.1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 states the scope of the current validation and the planned extensions. Section 3.3 distinguishes the conditional rollout bound from the empirical long-horizon results.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 1 (slice-wise informal) is stated in Section 3.3; its formal version (Theorem 3) and complete proof are given in Appendix B.2, with its three direct hypotheses (Assumptions 4–6) recorded in the same appendix. A tighter linear non-expansive special case is stated and proved as Theorem 2 in Appendix B.1. The problem-level assumptions are formalised as Assumptions 1–3 in Appendix B.2, and the slice-wise Lipschitz estimate used in the proof is recorded as Lemma 1 in Appendix B.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The neural operator architecture, hypernetwork design, and training objective are described in Section 3; the training set, four evaluation tasks, time step, grid resolution, interface-width range, and reference solver are described in Section 4.1, with full per-shape sample counts and trajectory budgets given in Appendix D.1 (Table 3). Architecture and main training settings are described in Appendix C, with further optimisation settings specified in the configuration files referenced there, and the discrete metric formulas used to score every model are given in Appendix D.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A lightweight, runnable anonymised code package covering the IADR architecture, the per-ε DR specialist baseline, the dataset generation script, and the training/evaluation entry points is included in the supplemental material accompanying this submission. A public GitHub release with the full experimental codebase (including all baselines and the bootstrap evaluation pipeline) will be provided upon acceptance and publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 states the training/evaluation split (training on circles at five widths E_{train}; evaluation on the four-task suite covering in-distribution, ε-OOD, geometry-OOD, and joint-OOD settings), the training budgets (100 epochs for DISCO and 20,000 epochs for all other models), and the reference-solution provenance for each shape. Appendix C provides architecture and training details for IADR and the baselines (target-network kernel sizes, hypernetwork depth/width, learning rate, and rollout-training setting), with further optimisation settings specified in the configuration files referenced there, and Appendix D.1 fixes the dataset construction (sample counts, trajectory budgets, initial-set parameters).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The errors in Tables 1, 5, and 8 are bootstrap means, with their associated 1-sigma standard deviations reported alongside the means in Tables 5 and 8. These statistics are computed from $B = 2000$ bootstrap resamples of per-trajectory metric values and reflect trajectory-level sampling variability at a fixed model checkpoint, shape, and conditioning ε. Table 2 reports the mean and standard deviation of single-step inference time across ε within each shape.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All training runs were carried out on a single NVIDIA V100 GPU. The wall-clock time per run is 16 h for IADR, 3 h for each of the eight per-ε DR specialists, 60 h for U-Net, 65 h for FNO, 72 h for DISCO, 18 h for Hyper-D, 15 h for Hyper-R, and 14 h for Blind DR, summing to roughly 284 GPU-hours for the training runs listed here. The full research project, including preliminary architecture searches, baseline-tuning iterations, and failed runs not reported in the paper, consumed more than ten times this amount (above 2,900 GPU-hours of total V100 compute).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms in every respect to the NeurIPS Code of Ethics. The work does not involve human subjects, personal data, or assets scraped from the web.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper proposes a numerical surrogate for a geometric partial differential equation (phase-field mean-curvature flow). It has no direct application to decision-making, user-facing systems, or other downstream deployments, and we do not identify a specific positive or negative societal impact beyond its contribution to scientific computing.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper releases neither pretrained language models, image generators, nor scraped datasets. The released artifacts (training and inference code for a phase-field mean-curvature surrogate) pose no identifiable risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We adapt the official DISCO codebase [33] as the meta-learning baseline, retaining DISCO's own U-Net target and adapting the dataloader to the phase-field trajectories. Our DR splitting baseline is implemented following the description and reference code of Bretin et al. [2]; the FNO baseline [27] is re-implemented from the published paper. These works are credited at first use in Sections 2–4.1, and the original licenses of the upstream repositories are respected and preserved in the supplemental code package. The four evaluation datasets (circles, spiral disks, five-pointed stars, and dumbbells) are constructed from scratch and introduce no third-party data.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the IADR training and evaluation code together with the dataset generation scripts (circles, spiral disks, and five-pointed stars at the interface-width range E, with analytic reference trajectories for circles and higher-order DR splitting reference trajectories for spiral disks and five-pointed stars). Both are bundled in the anonymised

supplemental code package described in Q5, accompanied by a `README` that lists the entry-point commands, expected runtime, and the per-task hyperparameter configurations of Appendix C. Trained model checkpoints are not released; the released scripts are sufficient to reproduce them under the protocol of Section 4.1.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects; no IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Large language models did not contribute to the core methodology, the experimental design, the theoretical analysis, or any algorithmic component of this research. Any LLM usage was limited to incidental writing and editing assistance, which falls outside the scope of this declaration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.