Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Geometry-aware LegONet for PDE Learning on Arbitrary Domains

Jiahao Zhang
Purdue University System
Yueqi Wang
Purdue University System
Guang Lin
guanglin@purdue.edu
Purdue University System

Article

Keywords:

Posted Date: September 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10749436/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Geometry-aware LegONet for PDE Learning on Arbitrary Domains

Jiahao Zhang[1†], Yueqi Wang[1†], Guang Lin[1,2*†]

[1]Department of Mathematics, Purdue University, West Lafayette, IN, 47907, USA.
[2]School of Mechanical Engineering, Purdue University, West Lafayette, IN, 47907, USA.

*Corresponding author(s). E-mail(s): guanglin@purdue.edu;
†These authors contributed equally to this work.

Abstract

Learned PDE solvers often entangle governing operators with the geometry, boundary conditions, and discretization used for training. This limits reuse when the same physics is posed on new domains, and it also makes physical-law discovery geometry-dependent. We introduce Geometry-aware LegONet (gLegONet), a boundary-manifold extension of Lego-like operator learning. Physical mechanisms are pretrained once as modular variational blocks on an ambient spectral domain. For a target geometry, sampled boundary constraints define an affine admissible manifold. Its mass-orthonormal tangent coordinates are used to evolve the dynamics and evaluate candidate law-discovery features directly. Changing the domain therefore changes only an algebraic coordinate interface, not the learned operator blocks. This converts arbitrary-domain PDE learning from geometry-specific retraining or soft penalty enforcement into boundary-guaranteed assembly of reusable mechanisms. In forward simulations and sparse identification tests on unseen domains, the method maintains boundary residuals near the algebraic tolerance and yields predictive governing laws from short-time observations.

Introduction

Learned solvers for partial differential equations have advanced rapidly, including physics-informed networks, neural operators and learned simulators[1–6]. Their use in

scientific computing, however, is still limited by geometry. A physical mechanism such as diffusion, transport or reaction can appear on many domains, but many learned solvers are trained on a fixed family of geometries. When the domain changes from a regular grid to a disk, channel or mixed-boundary region, the model may require new training data, architectural changes or an additional boundary treatment. This coupling between learned physics and geometric realization makes it difficult to build reusable learned solvers for related physical problems.

Classical numerical methods make this separation explicit. The PDE specifies the interior operator, while the mesh, trial space, quadrature and boundary constraints determine how that operator is realized on a particular domain [7–9]. Neural solvers often mix these roles inside one trained model. Physics-informed neural networks can handle irregular domains through collocation, and some trial functions can impose selected boundary conditions exactly [2,10,11]. However, the optimization is generally repeated for each target problem, and exact trial constructions become increasingly specialized for complex, mixed or derivative-valued boundary conditions.

Neural operators address this problem by amortizing inference across families of inputs. Foundational DeepONet, Fourier neural operator and U-shaped neural operator architectures were primarily developed on fixed domains or structured representations [3–5,12]. Recent works have made neural operators more geometry-aware. Geo-FNO, DAFNO and GINO develop deformations, domain indicators or graph-to-Fourier interfaces [13–15]. GNOT and Transolver use attention on irregular discretizations, whereas ArGEnT uses a Transformer-based DeepONet trunk to encode geometry from point-cloud representations [16–18]. PI-GANO combines a geometry encoder with physics-informed operator training, and boundary-focused methods such as BENO explicitly condition solution operators on complex geometries and boundary data [19–21]. Geo-NeW instead jointly learns a geometry-conditioned differential operator and compatible reduced finite-element spaces [22]. These methods greatly expand geometric flexibility by allowing the target geometry and, where applicable, boundary data to condition a learned solution operator or discretization, typically trained on prescribed families of domains or boundary configurations. Our objective differs in what is learned and reused: we keep the pretrained ambient mechanism blocks fixed and realize them on each new domain, without geometry-specific retraining, through a deterministic boundary-adapted Galerkin interface.

Our previous Lego-like variational operator learning framework [23] addressed one part of this problem: *physics modularity*. Rather than learning a monolithic PDE solver, it decomposes an equation into reusable variational operator blocks, such as diffusion, transport and reaction, and composes them by structure-aware time integration [23–25]. That formulation makes physical mechanisms reusable, but its most accurate implementation still relied on boundary-compatible bases tailored to particular baseplates. In other words, the physics became modular, but the geometry remained built into the representation.

This work supplies the missing geometry interface. The key idea is to reuse a frozen ambient mechanism library on a new domain by changing coordinates, rather than by retraining the library or correcting the boundary after each step. Let $\mathcal{B}_\theta^Q = \{\mathbf{F}_i^\theta\}_{i=1}^{N_{\mathrm{lib}}}$ be a library pretrained once on a regular domain Q. We ask whether this library

can be reused on new embedded target domains while enforcing new linear trace constraints algebraically. A tempting alternative is to evolve an unconstrained ambient state and project it back to the admissible space after each time step. This projection–evolution splitting is generally not equivalent to integrating the M_Ω-Galerkin vector field on the admissible tangent space, unless the projection metric, tangent space and time integrator are specifically matched[25]. We instead build the sampled boundary constraints into the coordinates before time integration begins.

For a target domain $\Omega \subset Q$, consider a PDE decomposed into physical mechanisms as

$$\partial_t u(\mathbf{x}, t) = \sum_{i=1}^{N_{\mathrm{blk}}} c_i L_i^\Omega(u)(\mathbf{x}, t), \qquad \mathbf{x} \in \Omega,$$

where L_i^Ω is the ith target-domain mechanism and c_i is its equation-specific scalar coefficient. Sampling the boundary operator in the ambient coefficient space gives

$$C\mathbf{a}(t) = \mathbf{d}(t), \qquad \mathbf{a}(t) = \mathbf{a}_{\mathrm{bc}}(t) + N_\Omega \mathbf{z}(t).$$

Here $\mathbf{a}_{\mathrm{bc}}(t)$ is an affine lift of the prescribed boundary data. The columns of N_Ω span the homogeneous boundary-compatible directions and are orthonormal in the M_Ω inner product. Thus the state $\mathbf{a}(t)$ satisfies the sampled boundary constraints by construction. For each mechanism L_i^Ω, the frozen ambient block is realized on Ω as a reduced response $\mathbf{q}_i^\Omega(\mathbf{z}, t) \in \mathbb{R}^r$. The boundary-adapted dynamics are therefore

$$\dot{\mathbf{z}}(t) = \sum_{i=1}^{N_{\mathrm{blk}}} c_i \, \mathbf{q}_i^\Omega(\mathbf{z}(t), t) - N_\Omega^\top M_\Omega \dot{\mathbf{a}}_{\mathrm{bc}}(t).$$

For static boundary data, $\dot{\mathbf{a}}_{\mathrm{bc}}(t) = \mathbf{0}$. Thus this equation keeps the same physical coefficients c_i, but replaces each mechanism by its boundary-adapted reduced response. Geometry and boundary conditions change only the algebraic realization layer $(C, \mathbf{a}_{\mathrm{bc}}, N_\Omega, M_\Omega)$. The neural parameters of the ambient mechanism library remain fixed.

This formulation clarifies the role of geometry transfer. Boundary conditions are not imposed by adding a penalty to the loss; they are built into the coordinates. The neural blocks are not retrained for each new domain; they are realized through a new algebraic interface. It is also distinct from post-step state projection, because the reduced dynamics are derived and integrated directly in the boundary-adapted coordinates. As a result, the sampled boundary constraints are satisfied to the tolerance of the null-space construction throughout the rollout. The main remaining errors come from the expressiveness of the boundary-adapted space, the accuracy of the frozen block responses, and the time discretization.

These boundary-adapted coordinates also provide a natural space for physical-law identification. Traditional sparse discovery methods such as SINDy and PDE-FIND identify governing terms from candidate libraries[26–28]. In our setting, the candidate library is not rebuilt from raw spatial derivatives on each new geometry. Instead, frozen mechanism blocks and analytic terms are first realized through the same boundary-adapted Galerkin interface and then regressed in the reduced coordinate $\mathbf{z}$. Once their

coefficients are identified, the discovered law already defines a boundary-constrained rollout model on the target domain.

Fig. 1 summarizes the resulting pretrain–realize–rollout pipeline. We evaluate this idea on unseen geometries with diverse boundary conditions, nonlinear transport, obstacle flow, higher-order operator composition and short-time inverse discovery. Across these tests, the boundary constraints are controlled by the algebraic coordinate construction, while the same pretrained mechanisms are reused without geometry-specific neural retraining. Geometry-aware LegONet therefore connects modular neural mechanism learning with the classical principle that interior physics and boundary-constrained realization should be separated.

Results

Reusable mechanism blocks are trained once and frozen

All experiments reuse ambient libraries pretrained on the square $Q = [-1, 1]^2$ before any target geometry or boundary condition is introduced. For each Fourier cutoff K, we train one resolution-matched library $\mathcal{B}_\theta^Q = \{\mathbf{F}_\Delta^\theta, \mathbf{F}_x^\theta, \mathbf{F}_y^\theta\}$, containing diffusion Δu and the two directional quadratic-transport mechanisms uu_x, uu_y. Once trained, the neural parameters are frozen. All subsequent target problems—regardless of geometry, boundary type or rollout equation—use the same pretrained blocks. Unless stated otherwise, all experiments use $K = 22$. Held-out block errors remain low and stable over the tested resolutions (Fig. 2).

Frozen blocks transfer across diverse geometries and boundary conditions

We first ask whether the same frozen library can be realized on unseen domains and boundary operators without geometry-specific retraining. Five manufactured-solution benchmarks provide exact trajectories and therefore reflect boundary realization and reduced-space rollout: Dirichlet reaction–diffusion on a rosette, Neumann reaction–diffusion on a crescent, nonhomogeneous Robin dynamics on a bunny-shaped domain, a coupled annular-star system with mixed Dirichlet–Neumann constraints, and Burgers transport on a pinwheel domain (Fig. 3). All five tests use the same $K = 22$ library; only C, $\mathbf{a}_{\mathrm{bc}}$, M_Ω and N_Ω are rebuilt.

PINN is optimized separately for each target, whereas FNO and UNO are trained on supervised trajectories from nearby manufactured families. Geometry-aware LegONet uses neither target-specific neural training nor target-family trajectories. Across the five tests, it achieves 10^{-4}–10^{-3}-level final errors with runtimes of 2.12–19.5 s, compared with hundreds to thousands of seconds for the baselines (Table 1). Homogeneous traces remain near the algebraic tolerance, and nonhomogeneous Robin data are imposed through an affine lift. Boundary leakage is therefore suppressed below the dominant interior-approximation, retained-rank and time-integration errors. Full benchmark and baseline specifications are provided in Supplementary Information.

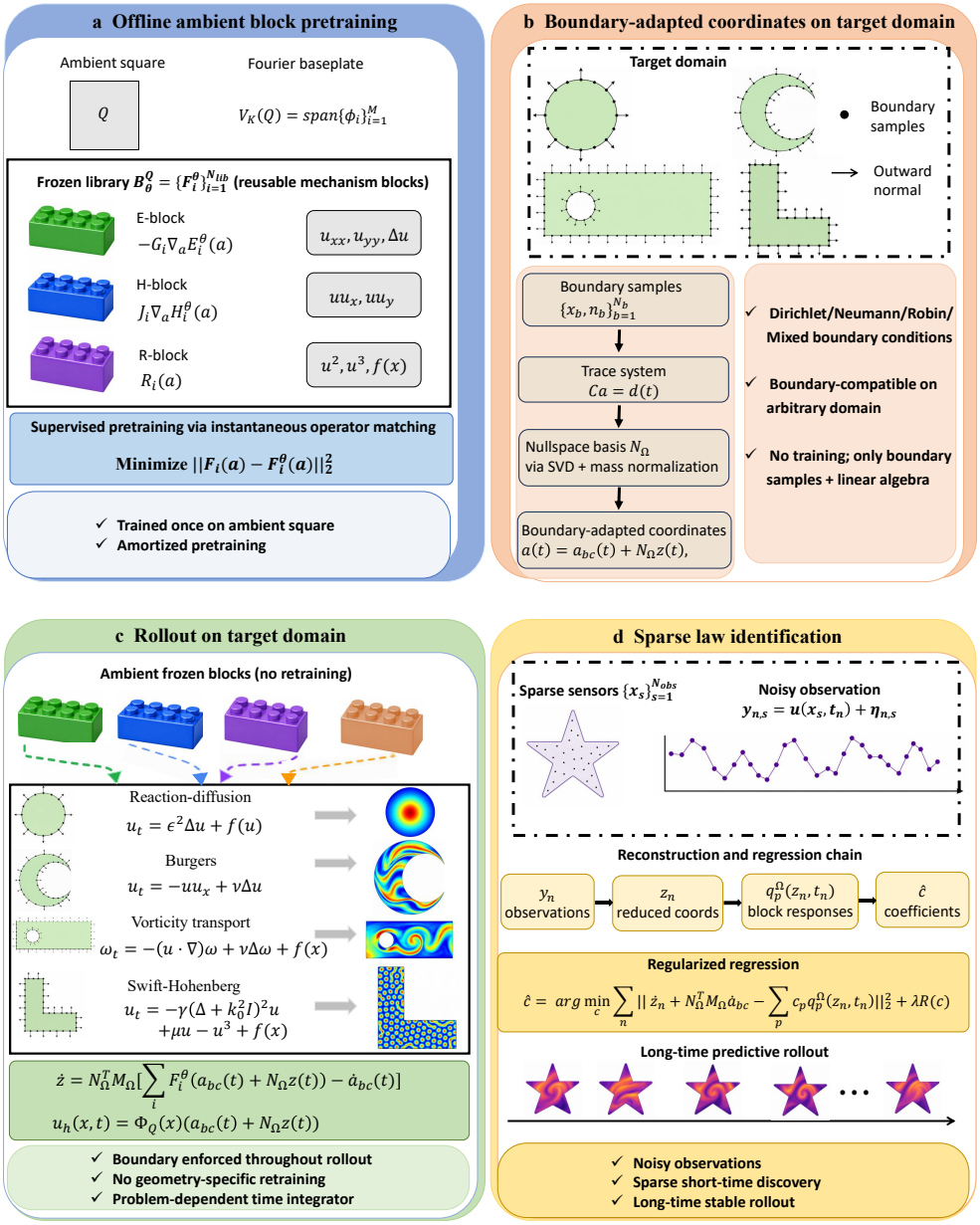


Fig. 1: Workflow of Geometry-aware LegONet. **a**, Single-mechanism blocks are pretrained once on the ambient square Q and frozen as $\mathcal{B}_\theta^Q$. **b**, Boundary samples define $C\mathbf{a} = \mathbf{d}(t)$; null-space and mass normalization yield admissible coordinates $\mathbf{a} = \mathbf{a}_{\mathrm{bc}} + N_\Omega \mathbf{z}$. **c**, The frozen blocks are assembled through the Ω-Galerkin dynamics and rolled out on new domains without neural retraining. **d**, The same coordinates map sparse short-time observations to reduced states, enabling mechanism identification and direct long-time prediction.

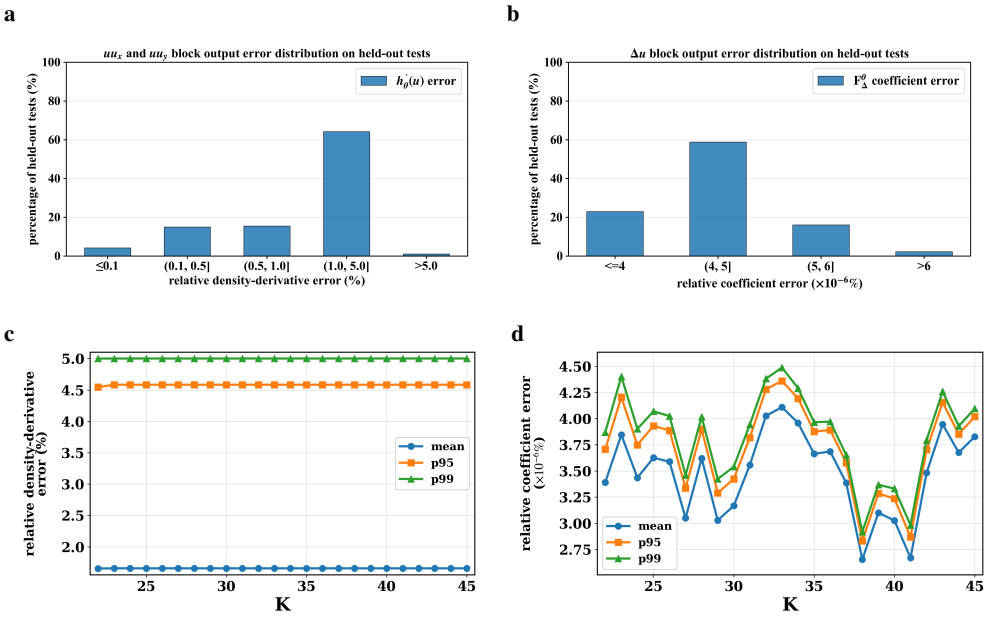


Fig. 2: Held-out diagnostics and resolution sensitivity of the reusable ambient blocks. Panels (a,b) show held-out error distributions for the learned transport primitive derivative $h'_\theta(u)$ and the Laplace block coefficient output, respectively; bars report the percentage of held-out samples in each error range. Panels (c,d) show the resolution sensitivity of the same two ambient blocks under separate pretraining in $\mathcal{V}_K(Q)$ for different Fourier cutoffs K. The diagnostics remain stable as the ambient resolution increases over the tested range.

Geometry-aware LegONet preserves constrained dynamics

We next ask whether gLegONet preserves the structure of a nonlinear constrained system, rather than only matching field values. The benchmark is the volume-constrained Allen–Cahn equation on the disk $\Omega_{\text{disk}} = \{\mathbf{x} : x^2 + y^2 \le 0.4^2\}$, with $\varepsilon^2 = 5 \times 10^{-3}$ and final time $T = 4.0$:

$$\begin{cases} u_t = \varepsilon^2 \Delta u + u - u^3 - \lambda(t), & \mathbf{x} \in \Omega_{\text{disk}}, \\ \partial_{\mathbf{n}} u = 0, & \mathbf{x} \in \partial\Omega_{\text{disk}}, \end{cases} \qquad \lambda(t) = \frac{1}{|\Omega_{\text{disk}}|} \int_{\Omega_{\text{disk}}} (u - u^3)\, d\mathbf{x}.$$

Here $\lambda(t)$ preserves the spatial mean, making this a joint test of field accuracy, boundary enforcement and global conservation. We use $N_b = 1600$ Neumann boundary samples and retain $r = 412$ boundary-adapted coordinates. An independent P_1 finite-element solution is used as reference. We report $D^{\max}_{\text{mass}} = \max_t |\bar{u}_h(t) - \bar{u}_h(0)|$, where $\bar{u}_h$ is the quadrature-weighted spatial mean.

Fig. 4a,b show that gLegONet reproduces the FEM two-phase state with final relative error 2.98×10^{-2}. The transferred coordinates also preserve the constrained

a

Test	Domain	Equation	Boundary condition	N_b	r
MMS-I	Rosette	$u_t = \varepsilon^2 \Delta u + \mu u - \beta u^3 + f^{\Omega}$	$u = 0$	420	207
MMS-II	Crescent	$u_t = \varepsilon^2 \Delta u + \mu u - \beta u^3 + f^{\Omega}$	$\partial_{\mathbf{n}} u = 0$	620	220
MMS-III	Bunny	$u_t = D \Delta u + r_g u(1 - u/K_c) + f^{\Omega}$	$\partial_{\mathbf{n}} u + \kappa u = g_R$	420	843
MMS-IV	Annular star	$\begin{cases} u_t = D_u \Delta u + a - u + u^2 v + f_u^{\Omega}, \\ v_t = D_v \Delta v + b - u^2 v + f_v^{\Omega} \end{cases}$	$\begin{cases} u = v = 0 \text{ on } \Gamma_{\text{out}}, \\ \partial_{\mathbf{n}} u = \partial_{\mathbf{n}} v = 0 \text{ on } \Gamma_{\text{in}} \end{cases}$	720	832
MMS-V	Pinwheel shell	$u_t = \nu \Delta u - u u_x - u u_y + f^{\Omega}$	$u = 0$	420	644

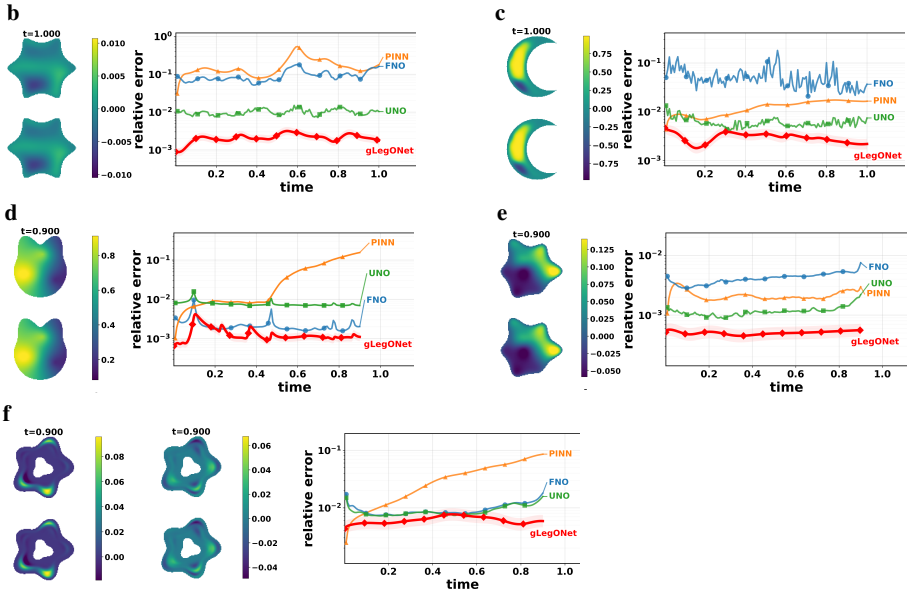


Fig. 3: Manufactured-solution benchmarks on unseen embedded domains. **a**, Summary of the governing equation, boundary condition, number of boundary samples N_b, and retained reduced dimension r for the five benchmarks. **b**, MMS-I on the rosette domain. **c**, MMS-II on the crescent domain. **d**, MMS-III on the bunny-shaped domain. **e**, MMS-V on the pinwheel-shell domain. **f**, MMS-IV on the annular-star domain, including the u- and v-components. Panels **b**–**f** combine representative reference and gLegONet snapshots with relative $L^2(\Omega)$ error histories against PINN, FNO and UNO. Full multi-time rollout visualizations and benchmark-specific numerical details are provided in Supplementary Information.

structure: the maximum mass drift is 3.63×10^{-5} (Fig. 4c), and the dense Neumann residual is 3.47×10^{-9} (Table 2). PINN is the closest baseline in final field error, but its error is 1.6 times larger and its mass drift is 2.4×10^2 times larger; FNO and UNO are less accurate. Panels (d,e) show that increasing r reduces the error until saturation, whereas additional boundary samples have little effect once the smooth Neumann trace is resolved. Reduced-space capacity, rather than boundary sampling, is therefore the dominant limitation. Numerical settings are provided in Supplementary Information.

Transferred reduced bases match geometry-specific bases

We next test whether a reduced basis obtained by boundary transfer can match a basis constructed directly for the target geometry in a nonlinear transport problem. On the embedded square $\Omega = [-h, h]^2$, with $h = 0.55$, $\nu = 10^{-2}$ and final time $T = 0.5$, we solve

$$\begin{cases} u_t + uu_x + vu_y = \nu\Delta u, \\ v_t + uv_x + vv_y = \nu\Delta v, \end{cases} \qquad \mathbf{x} \in \Omega, \qquad u = v = 0 \text{ on } \partial\Omega.$$

This benchmark requires diffusion, directional transport and cross-advection to be composed in one Dirichlet-compatible coordinate space. The same boundary-adapted matrix N_Ω is used for both velocity components. It is constructed from $N_b = 1000$ boundary samples and retains $r = 833$ coordinates. The frozen diffusion and transport blocks are reused through this interface, while cross-advection is assembled by analytic weak projection. An independent 201×201 upwind finite-volume solution provides the reference.

A direct inner-square block solver from our previous work[23] serves as a geometry-specific control, because its basis is built for the target boundary. gLegONet reaches the same accuracy without that tailored basis (Table 2): the blocks are reused as frozen ambient mechanisms, and only the boundary-adapted realization changes. As shown in Fig. 5a–c, both velocity components track the reference, and the transferred and direct solvers have nearly overlapping error histories below the neural baselines. The ablations in Fig. 5d–f clarify the expected operating regime: the embedded square should remain sufficiently inside the ambient box, boundary-sample refinement has little effect once the constraints are resolved, and increasing r improves accuracy until the reduced space saturates. Thus the transfer limit is ambient-resolution and reduced-space capacity, not boundary leakage. Numerical settings are provided in Supplementary Information.

Frozen blocks compose into complex operator assemblies

We next test whether frozen second-order mechanisms can be assembled into more complex operator systems. The first case is Navier–Stokes vorticity transport past an embedded cylinder:

$$\begin{cases} \partial_t\omega + (\mathbf{u}_{\mathrm{bg}} + \nabla^\perp\psi)\cdot\nabla\omega = \nu\Delta\omega + f, \\ -\Delta\psi = \omega, \end{cases} \qquad \omega = \psi = 0 \text{ on } \partial\Omega.$$

The embedded channel is $\Omega = [-0.95, 0.95] \times [-0.2375, 0.2375] \setminus B_{0.0523}\big((-0.6769, 0)\big)$, where $B_R(\boldsymbol{\xi}_0)$ denotes the open disk of radius R centred at $\boldsymbol{\xi}_0$. We use $\nu = 10^{-2}$ and $T = 1.8$. This benchmark combines a multiply connected boundary, an auxiliary Poisson solve for ψ and advective wake transport. With $K = 45$ and $r = 6099$, gLegONet achieves final relative error 3.85×10^{-2}, and the dense boundary residual is 1.59×10^{-12} (Fig. 6a). The K-ablation shows that boundary accuracy is retained while interior errors decrease as the ambient representation is enriched (Table 3).

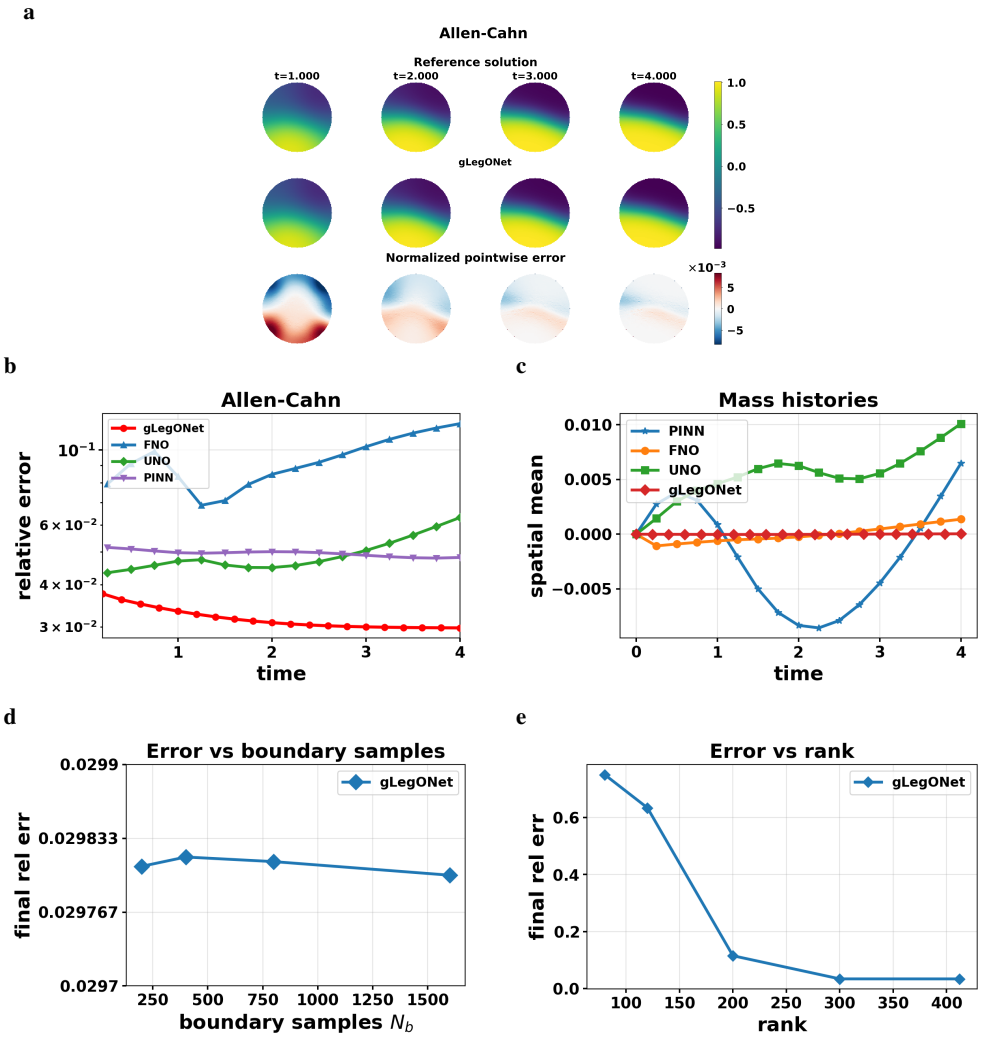


Fig. 4: Boundary-adapted gLegONet rollout and ablations for volume-constrained Allen–Cahn dynamics on the disk. Panel (a) compares the finite-element reference with the gLegONet prediction. Panels (b,c) report the relative field error and spatial mean, respectively. Panels (d,e) vary the number of boundary samples and the retained reduced rank. The error is weakly sensitive to N_b once the Neumann trace is resolved, but decreases sharply with r before saturating, showing that reduced-space capacity rather than boundary sampling is the dominant limitation.

The second case is the clamped Swift–Hohenberg equation on a heart-shaped embedded domain,

$$\partial_t u = -\gamma(\Delta + k_0^2 I)^2 u + \mu u - u^3 + f, \qquad u = \partial_{\mathbf{n}} u = 0 \text{ on } \partial\Omega_{\text{heart}}.$$

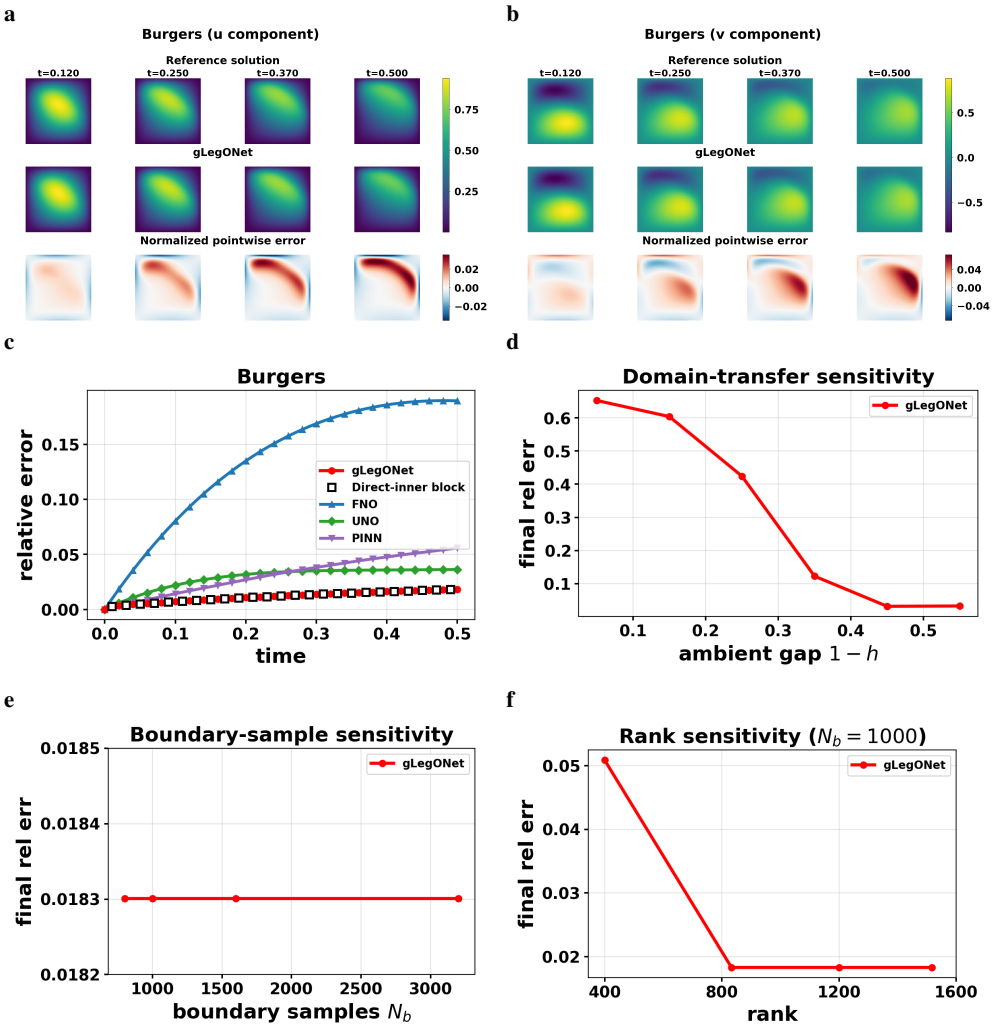


Fig. 5: gLegONet prediction for nonlinear vector Burgers dynamics. Panels (a,b) compare the reference and gLegONet rollouts for the u- and v-components. Panel (c) compares the relative error histories with the neural and geometry-specific baselines. Panel (d) varies the ambient-boundary gap $1 - h$, while panels (e,f) vary the number of boundary samples and the retained rank. The error rises as the target domain approaches the ambient boundary, changes little once the Dirichlet trace is sufficiently sampled, and decreases with r until saturation. The limiting factors are therefore ambient resolution and reduced-space capacity rather than boundary leakage.

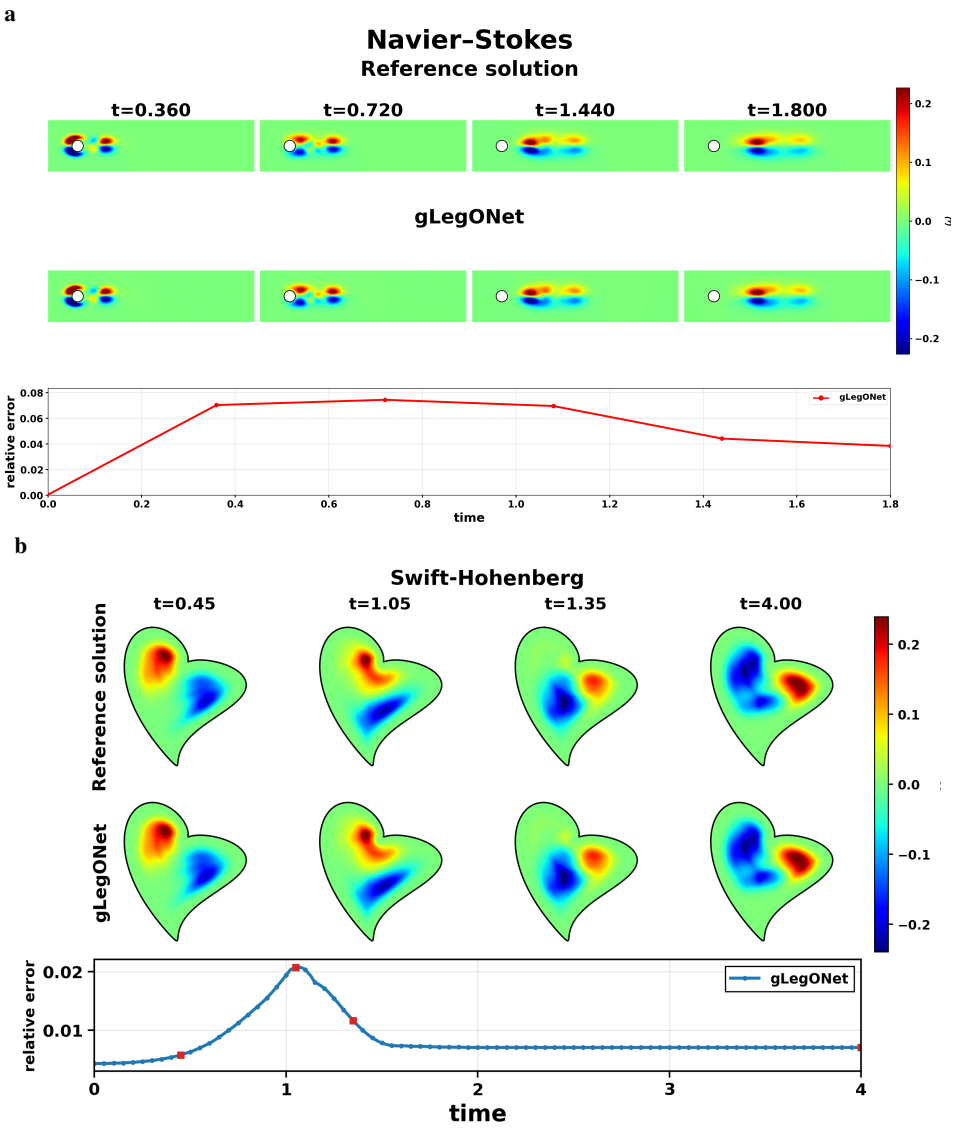


Fig. 6: gLegONet stress tests for boundary transfer and operator composition. Panel (a) shows vorticity transport past an embedded cylinder, where the solver must preserve homogeneous traces on both the channel wall and the obstacle while propagating a wake interaction. Panel (b) shows the clamped Swift–Hohenberg benchmark on a heart-shaped domain, where the solver simultaneously enforces $u = 0$ and $\partial_{\mathbf{n}} u = 0$, and composes the reusable second-order Laplace block into a fourth-order mechanism without training a fourth-order block. Detailed configurations, manufactured forcing, Fourier-resolution ablations and boundary diagnostics are given in Supplementary Information.

We use $\gamma = 4.0 \times 10^{-3}$, $k_0 = 5.3$, $\mu = 0.05$, $T = 4.0$, $\Delta t = 10^{-3}$, and a $K = 30$ boundary-adapted space with retained dimension $r = 795$. No fourth-order neural block is trained. Instead, $(\Delta + k_0^2 I)^2$ is assembled by composing the pretrained Laplace block inside the boundary-adapted space. This controlled test prescribes a smooth manufactured path. Fig. 6b shows that the composed fourth-order rollout tracks the reference evolution with sub-10^{-2} final relative error and peak error below 2.1×10^{-2}. Both clamped traces are retained, with dense value- and normal-trace residuals below 2×10^{-8}. The Fourier-resolution ablation in Table 3 confirms that increasing K primarily improves interior resolution rather than boundary realization.

Together, these tests show that the same interface supports multiply connected domains, auxiliary solves, multiple boundary traces and higher-order operator composition. Full configurations are provided in Supplementary Information.

Sparse physical-law discovery on unseen domains

Finally, we test whether gLegONet can use the same boundary-transfer layer for inverse discovery. To isolate the effect of the candidate mechanisms from the size of the reduced representation, all identification experiments use the same rank-80 homogeneous Dirichlet boundary-adapted space on each unseen domain. The candidate library is

$$\mathcal{D}_\theta^\Omega = \{\mathbf{q}_\Delta^\Omega, \mathbf{q}_x^\Omega, \mathbf{q}_y^\Omega, \mathbf{q}_u^\Omega, \mathbf{q}_{u^2}^\Omega, \mathbf{q}_{u^3}^\Omega\},$$

combining realized frozen blocks with analytic polynomial maps. No neural parameter is updated during identification; only the sparse coefficients multiplying these realized mechanisms are fitted from short-time observations.

Identification is deliberately data limited. We use $N_{\mathrm{traj}} = 16$ independent short trajectories over $T_{\mathrm{id}} = 0.012$, sampled every $\Delta t_{\mathrm{obs}} = 5 \times 10^{-5}$, from either $N_{\mathrm{obs}} = 1000$ or 240 fixed random interior sensors. The recovered law is then rolled out to $T_{\mathrm{roll}} = 1.0$, more than 80 times longer than the identification window. The noise level σ denotes relative additive observation noise.

The peanut dumbbell tests reaction–diffusion through a narrow bottleneck, with true coefficient vector

$$\mathbf{c}^\star_{\mathrm{peanut}} = (0.04, 0, 0, 0.06, 0, -0.35).$$

The oblique sinusoidal channel combines nonlinear transport, reaction terms and a thin anisotropic geometry, with

$$\mathbf{c}^\star_{\mathrm{channel}} = (0.02, -0.06, 0.09, -0.03, 0.025, -0.30).$$

With $N_{\mathrm{obs}} = 1000$ and no observation noise, gLegONet recovers the peanut law almost exactly, giving $e_{\mathrm{coef}} = 6.89 \times 10^{-5}$ and $e_{\mathrm{roll}} = 3.17 \times 10^{-5}$. Even at $\sigma = 10^{-2}$, the long-time rollout remains stable with final error 4.53×10^{-2}. The channel is more demanding because the active law contains both directional transport terms. Nevertheless, at zero noise gLegONet gives $e_{\mathrm{coef}} = 1.25 \times 10^{-2}$ and $e_{\mathrm{roll}} = 3.44 \times 10^{-2}$, and remains predictive under low-to-moderate noise before degrading at the highest noise level.

The ablations show that the reduced boundary-adapted representation is essential. Replacing it by a rank-matched ambient Fourier basis fails: coefficient errors exceed 10^2 and the rollout errors are near one on both domains. Sparse finite differences also fail under the same sensor budget, because scattered observations do not provide stable derivative features. The sparse FD coefficient error on the peanut domain reaches 9.33×10^2, and the channel rollout becomes unstable. Dense finite differences are competitive only when given the full masked grid, which is a much stronger observation setting than the sparse sensor data used by gLegONet. Reducing the sensor count to $N_{\mathrm{obs}} = 240$ still gives a predictive peanut rollout ($e_{\mathrm{roll}} = 1.83 \times 10^{-4}$), while the channel becomes less accurate but remains finite.

Fig. 7a summarizes coefficient recovery and long-time prediction under noise and sensor ablations. Panels (b,c) and (d,e) show representative peanut and channel rollouts, respectively. Together, these results show that gLegONet turns sparse observations into predictive physical laws by evaluating frozen mechanism blocks in a geometry-compatible boundary-adapted coordinate system.

Discussion

This work introduces Geometry-aware LegONet as a geometry-transfer layer for reusable PDE learning. The central idea is to separate two roles that are often entangled in neural PDE solvers: learning interior physical mechanisms and realizing those mechanisms on a particular geometry. The physical mechanisms are represented by Lego-like variational blocks pretrained once in an ambient coefficient space. The geometry and boundary condition are represented separately through algebraic constraints that define boundary-adapted coordinates. In this formulation, changing the domain changes the coordinate interface, not the learned physics.

The results show that this separation is useful across both forward and inverse problems. In the manufactured benchmarks, one frozen ambient block library transfers across several boundary operators without target-domain neural retraining. In the Allen–Cahn disk problem, hard Neumann coordinates help preserve a constrained nonlinear evolution against an independent finite-element reference. In the vector Burgers experiment, nonlinear transport blocks transfer with little loss relative to a geometry-specific block construction when the target domain is resolved by the ambient space. The cylinder-wake and clamped Swift–Hohenberg examples further show that the same boundary interface can support Poisson coupling and higher-order operators assembled from lower-order reusable blocks. Finally, the law-identification tests show that boundary-adapted coordinates provide stable candidate features for discovering governing equations from short-time data on unseen domains.

A useful interpretation is that the method changes the unit of reuse in scientific machine learning. In many neural-operator approaches, the reusable object is a full solution map on a prescribed geometry or mesh distribution. In the original LegONet framework, the reusable object is a physical mechanism block. In the present work, these reusable mechanism blocks are connected to new geometries by an algebraic boundary interface. This suggests a route toward scientific machine-learning libraries in which physical mechanisms are pretrained once and deployed on new domains

a

Setting	Data	Peanut e_{coef}	Peanut e_{roll}	Channel e_{coef}	Channel e_{roll}
Noise robustness, gLegONet with $N_{\mathrm{obs}} = 1000$					
gLegONet	$\sigma = 0$	$6.89\times10^{-5} \pm 5.02\times10^{-5}$	$3.17\times10^{-5} \pm 1.14\times10^{-5}$	$1.25\times10^{-2} \pm 3.74\times10^{-4}$	$3.44\times10^{-2} \pm 3.66\times10^{-4}$
gLegONet	$\sigma = 10^{-4}$	$3.90\times10^{-3} \pm 3.90\times10^{-3}$	$4.68\times10^{-4} \pm 1.52\times10^{-4}$	$1.31\times10^{-2} \pm 6.55\times10^{-4}$	$3.17\times10^{-2} \pm 5.10\times10^{-3}$
gLegONet	$\sigma = 10^{-3}$	$3.96\times10^{-2} \pm 3.95\times10^{-2}$	$4.58\times10^{-3} \pm 1.59\times10^{-3}$	$2.23\times10^{-2} \pm 2.39\times10^{-2}$	$6.78\times10^{-2} \pm 1.97\times10^{-2}$
gLegONet	$\sigma = 10^{-2}$	$4.16\times10^{-1} \pm 4.01\times10^{-1}$	$4.53\times10^{-2} \pm 1.38\times10^{-2}$	$1.72\times10^{-1} \pm 2.35\times10^{-1}$	$2.84\times10^{-1} \pm 1.62\times10^{-1}$
Zero-noise sparsity and basis ablation					
gLegONet	$N_{\mathrm{obs}} = 240$	$6.20\times10^{-4} \pm 4.44\times10^{-4}$	$1.83\times10^{-4} \pm 5.64\times10^{-5}$	$5.04\times10^{-2} \pm 2.66\times10^{-2}$	$2.69\times10^{-1} \pm 1.30\times10^{-1}$
Ambient Fourier	$N_{\mathrm{obs}} = 1000$	$6.42\times10^{2} \pm 3.63\times10^{2}$	$1.0174 \pm 6.58\times10^{-1}$	$1.77\times10^{2} \pm 8.11\times10^{1}$	$9.98\times10^{-1} \pm 5.01\times10^{-1}$
Zero-noise finite-difference cost study					
FD dense	full masked grid	$3.12\times10^{-2} \pm 3.26\times10^{-2}$	$3.05\times10^{-4} \pm 1.04\times10^{-4}$	$8.02\times10^{-3} \pm 4.60\times10^{-4}$	$8.39\times10^{-3} \pm 1.24\times10^{-3}$
FD sparse	$N_{\mathrm{obs}} = 1000$	$9.33\times10^{2} \pm 9.77\times10^{2}$	$7.57\times10^{-1} \pm 3.24\times10^{-1}$	$4.77\times10^{2} \pm 2.67\times10^{2}$	unstable

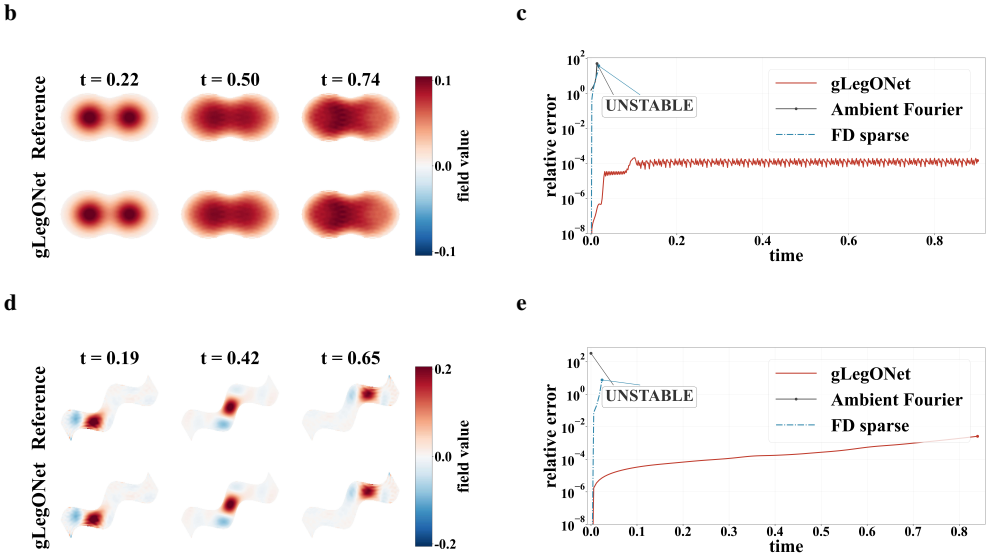


Fig. 7: Boundary-adapted physical-law identification on unseen domains. Panel (a) shows active-coefficient error e_{coef} and final rollout error e_{roll} under observation noise, reduced sensor counts, an ambient Fourier representation and finite-difference baselines. Entries with $\pm$ report the mean and standard deviation over three randomized seeds. Panels (b)–(e) show representative zero-noise rollout visualizations from fixed illustrative initial conditions chosen to make the transported structures clearly visible; these panels are single-seed examples and are separate from the randomized statistics in Panel (a). Panels (b) and (c) show the identified rollout and relative $L^2(\Omega)$ error history on the peanut dumbbell, respectively; Panels (d) and (e) show the corresponding results on the oblique sinusoidal channel. In Panels (c) and (e), *Ambient Fourier* uses an ambient Fourier representation of the same rank without encoding the target boundary; and *FD sparse* denotes the finite-difference discovery baseline using sparse observations. The label *UNSTABLE* indicates that the corresponding baseline diverges before reaching the final rollout time. Here e_{coef} is the relative error of the active mechanism coefficients and e_{roll} is the final relative $L^2(\Omega)$ error at $T_{\mathrm{roll}} = 1.0$, using only data from $T_{\mathrm{id}} = 0.012$.

by boundary assembly, rather than by retraining a monolithic solver for every new configuration.

The framework also clarifies where errors come from. Because the boundary condition is embedded in the coordinate system, boundary leakage is largely removed as a dominant failure mode. When the boundary residual remains near numerical precision but the field error increases, the cause is not weak boundary enforcement. It is instead associated with the expressiveness of the ambient basis, the retained rank of the boundary-adapted space, the accuracy of the pretrained blocks, quadrature or time integration. This separation is difficult to obtain in monolithic neural solvers, where geometry, boundary constraints and learned dynamics are usually mixed inside one model.

The method has several practical considerations. It inherits the resolution limits of the chosen ambient trial space: if the target geometry contains fine boundary features or requires modes beyond the selected cutoff, the boundary can remain satisfied while the interior approximation error grows. The deployment cost also differs from a fully amortized black-box neural time-stepper. A new target geometry requires boundary sampling, null-space construction, reduced operator assembly and coefficient-space rollout. However, these are algebraic and numerical assembly costs rather than neural retraining costs. Once the ambient block library has been trained, new geometries can be handled by rebuilding the boundary-coordinate interface, which is precisely the intended advantage of the framework. The formulation is not restricted to two dimensions, but large-scale three-dimensional arbitrary-boundary applications will require efficient implementations of boundary sampling, quadrature, rank selection and sparse reduced algebra. Strongly nonlocal, stochastic, multiscale or geometry-dependent mechanisms may also require additional reusable blocks or hybrid coupling with classical solvers.

These considerations clarify the role of Geometry-aware LegONet. The method is most useful when the same physical mechanisms appear across many geometries and boundary conditions, and when hard boundary satisfaction, interpretability and mechanism reuse are important. It is not designed merely to be the fastest black-box predictor for one fixed geometry. Rather, it provides a reusable mechanism library together with an algebraic interface for deploying that library on new domains. In this sense, the work extends Lego-like operator learning from physics modularity to geometry modularity. The first step was to make physical mechanisms reusable; the step taken here is to make those mechanisms deployable on new domains through boundary-adapted coordinates. This turns arbitrary-domain PDE learning from a geometry-specific retraining problem into an algebraic assembly problem over reusable scientific operators.

Methods

Ambient representation and reusable mechanism blocks

Geometry-aware LegONet does not learn a new trajectory model for each geometry or boundary condition. It pretrains reusable physical mechanism blocks once on an ambient domain, using a shared coefficient representation, and then realizes the frozen

blocks on each target domain through boundary-adapted Galerkin coordinates. Thus geometry and boundary data change the linear realization layer, while the neural block parameters remain fixed.

We first place all target problems inside a fixed regular base domain. In two dimensions, we use the ambient square $Q = [-1, 1]^2$, the standard tensor-product box for Fourier representation and quadrature. A physical domain Ω_{phys} is mapped into Q by an affine rescaling $T_\Omega : \Omega_{\mathrm{phys}} \to \Omega \subset Q$, and the rescaled PDE is posed on the embedded target domain Ω. The mapping consists only of translation, isotropic scaling and, when needed, a rigid rotation, so it does not change the physical mechanism. In Supplementary Information, we specify this physical-to-embedded rescaling before block pretraining and transfer are described.

In all experiments, the ambient trial space is the truncated real Fourier space on Q. Let $\mathcal{I}_K = \{(k, \ell) \in \mathbb{Z}^2 : k^2 + \ell^2 \le K^2\}$ and let $\mathcal{I}_K^+ = \{(k, \ell) \in \mathcal{I}_K : k > 0 \text{ or } (k = 0, \ell > 0)\}$, which selects one representative from each nonzero pair (k, ℓ) and $(-k, -\ell)$. We use the normalized real Fourier basis

$$\begin{aligned}\mathcal{V}_K(Q) &= \operatorname{span}\Big(\{1\} \cup \{\sqrt{2}\cos(\pi(kx + \ell y)), \sqrt{2}\sin(\pi(kx + \ell y)) : (k, \ell) \in \mathcal{I}_K^+\}\Big) \\ &:= \operatorname{span}\{\phi_1, \dots, \phi_M\}.\end{aligned}$$

The basis is orthonormal with respect to the normalized inner product $\langle u, v\rangle_Q = |Q|^{-1} \int_Q u(x)v(x)\,dx$. The choice of Fourier functions is deliberate. They provide a complete orthonormal coordinate system on the regular ambient box Q, allow analytic differentiation of basis functions, and give explicit coefficient projection. Let P_K be the orthogonal projection onto $\mathcal{V}_K(Q)$ under the normalized inner product. The key property we use is density in $L^2(Q)$[9,29]: for every $u \in L^2(Q)$,

$$\|u - P_K u\|_{L^2(Q)} \to 0 \qquad \text{as } K \to \infty.$$

Consequently, if a target-domain field $u \in L^2(\Omega)$ is represented by an ambient extension $\tilde{u} \in L^2(Q)$ with $\tilde{u}|_\Omega = u$, then the restricted Fourier approximation also converges on the embedded domain Ω. Thus the Fourier baseplate supplies a universal ambient coefficient interface, while the actual target geometry and boundary condition are imposed later by boundary-adapted coordinates.

Remark 1 The same realization principle is not tied to Fourier functions. One may replace $\mathcal{V}_K(Q)$ by another finite-dimensional trial space. The essential requirement is an approximation-complete trial family: the spaces $\mathcal{V}_K(Q)$ should be dense, as $K \to \infty$, in the function space in which the target PDE is posed. Polynomial spectral bases, finite-element bases and wavelet bases are examples satisfying this requirement in their standard settings[8,30,31]. The Fourier choice in this work is therefore a controlled baseplate for testing reusable block transfer, rather than an empirical claim that all admissible bases behave identically.

We now use this ambient approximation space as the coefficient language for mechanism learning. A coefficient vector $\mathbf{a} \in \mathbb{R}^M$ represents the ambient field

$$u_{\mathbf{a}}(\mathbf{x}) = \sum_{j=1}^{M} a_j \phi_j(\mathbf{x}) = \Phi_Q(\mathbf{x})\mathbf{a},$$

where $\Phi_Q(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x}))$. We denote the reconstruction map by $\mathcal{R}_Q : \mathbf{a} \mapsto u_{\mathbf{a}}$, and the coefficient projection by $\mathcal{P}_Q : u \mapsto (\langle u, \phi_1 \rangle_Q, \dots, \langle u, \phi_M \rangle_Q)$. Thus every ambient field used by the method is first represented in the finite coefficient space $\mathcal{V}_K(Q)$.

Let $\Omega \subset Q$ be an embedded target domain. A target PDE is posed on Ω and decomposed into physical mechanisms as

$$\partial_t u(\mathbf{x}, t) = \sum_{i=1}^{N_{\mathrm{blk}}} c_i L_i^{\Omega}(u)(\mathbf{x}, t), \qquad \mathbf{x} \in \Omega.$$

Here L_i^{Ω} is the ith mechanism acting on fields defined on Ω, and c_i is the scalar coefficient. In our coefficient representation, a target-domain state is represented by the restriction of an ambient Fourier field $u_{\mathbf{a}}$ to Ω. We define the ambient counterpart L_i^Q by using the same interior differential expression on Q, i.e., $L_i^Q(u)\Big|_{\Omega}$ agrees with $L_i^{\Omega}(u|_{\Omega})$ in the interior of Ω. Boundary conditions are not part of this ambient mechanism. They enter only through the coordinate interface in the next subsection. For $\mathbf{a} \in \mathbb{R}^M$, the ambient mechanism L_i^Q induces the coefficient-space vector field

$$\mathbf{F}_i(\mathbf{a}) = \mathcal{P}_Q \left[L_i^Q(\mathcal{R}_Q \mathbf{a}) \right] \in \mathbb{R}^M.$$

The pretraining objective is to construct a neural block $\mathbf{F}_i^{\theta} : \mathbb{R}^M \to \mathbb{R}^M$ that approximates $\mathbf{F}_i$. This block is therefore a coefficient-space representation of one mechanism on Q: for instance, a Laplace block maps the coefficients of $u_{\mathbf{a}}$ to the Fourier coefficients of $\Delta u_{\mathbf{a}}$. Note that, at this stage, the block does not encode a target geometry, a boundary condition, or a complete trajectory model. It only learns the reusable action of a single physical mechanism on the shared ambient coefficient state.

Following our previous structure-preserving LegONet design[23], a mechanism block can be written in the generator form

$$\mathbf{F}_i^{\theta}(\mathbf{a}) = -G_i \nabla_{\mathbf{a}} E_i^{\theta}(\mathbf{a}) + J_i \nabla_{\mathbf{a}} H_i^{\theta}(\mathbf{a}) + \mathbf{R}_i(\mathbf{a}).$$

Here G_i is a symmetric positive-semidefinite structure operator, $J_i^{\top} = -J_i$, $E_i^{\theta} : \mathbb{R}^M \to \mathbb{R}$ is a dissipative generator, $H_i^{\theta} : \mathbb{R}^M \to \mathbb{R}$ is a conservative generator, and $\mathbf{R}_i : \mathbb{R}^M \to \mathbb{R}^M$ contains residual effects. This formula is a template inherited from our previous work[23]. In the present library, blocks are deliberately single-purpose: a diffusion or relaxation mechanism is represented as an E-block, a transport or Hamiltonian mechanism as an H-block, and a forcing, reaction or closure mechanism as an

R-block. This modular design is what makes the library plug-and-play: mechanisms can be selected, scaled and recombined without retraining a monolithic solver.

The block parameters are obtained by supervised pretraining[23]. For each mechanism, training data are generated from either an analytic expression or a trusted reference discretization, and θ is optimized by an L^2-type supervised loss between the predicted and reference mechanism outputs. The exact sampling rules, reference mechanisms and losses used for each benchmark are reported in the Supplementary Information.

After pretraining, the ambient library on Q is frozen:

$$\mathcal{B}_\theta^Q = \{\mathbf{F}_i^\theta\}_{i=1}^{N_{\mathrm{lib}}}.$$

A target PDE, however, is posed only on the embedded domain $\Omega \subset Q$. The next subsection explains how the boundary condition on $\partial\Omega$ defines an admissible coordinate interface inside the ambient coefficient space, so that the frozen library $\mathcal{B}_\theta^Q$ can be realized on Ω without retraining its neural parameters.

Boundary-adapted coordinates on target domains

The frozen library $\mathcal{B}_\theta^Q$ acts on Fourier coefficients on the ambient square Q, whereas a target problem is posed on an embedded domain $\Omega \subset Q$ with its own boundary operator. Geometry-aware LegONet resolves this mismatch by changing coordinates rather than retraining the neural blocks. We construct an affine coordinate space in which every numerical state satisfies the sampled target boundary condition before any block composition or time integration is applied.

Let $\{(\mathbf{x}_b, \mathbf{n}_b)\}_{b=1}^{N_b}$ be samples on $\partial\Omega$, where $\mathbf{n}_b$ is the outward unit normal. We consider Dirichlet, Neumann and Robin trace operators $\mathcal{B}_D u = u$, $\mathcal{B}_N u = \partial_{\mathbf{n}} u$, and $\mathcal{B}_R u = \partial_{\mathbf{n}} u + \kappa u$, with prescribed data $\mathcal{B}_\star u = g_\star(\mathbf{x}, t)$. Here $\partial_{\mathbf{n}} u = \mathbf{n} \cdot \nabla u$, and κ is the Robin coefficient.

After stacking all scalar trace constraints, boundary sampling gives

$$C\mathbf{a} = \mathbf{d}(t), \tag{1}$$

where $C \in \mathbb{R}^{N_b \times M}$, $\mathbf{d}(t) \in \mathbb{R}^{N_b}$. For a row ℓ generated from the sample $\mathbf{x}_b$, the entries are

$$C_{\ell,:} = \begin{cases} \Phi_Q(\mathbf{x}_b), & \star = D, \\ \partial_{\mathbf{n}} \Phi_Q(\mathbf{x}_b), & \star = N, \\ \partial_{\mathbf{n}} \Phi_Q(\mathbf{x}_b) + \kappa(\mathbf{x}_b)\Phi_Q(\mathbf{x}_b), & \star = R, \end{cases} \qquad d_\ell(t) = g_\star(\mathbf{x}_b, t),$$

where $\partial_{\mathbf{n}} \Phi_Q(\mathbf{x}_b) = \mathbf{n}_b^\top \nabla \Phi_Q(\mathbf{x}_b) \in \mathbb{R}^{1 \times M}$. Mixed conditions are handled by assigning different trace types to different boundary segments; clamped conditions are obtained by stacking both value and normal-derivative rows.

The sampled boundary-admissible coefficient set is

$$\mathcal{M}_{\partial\Omega}(t) = \left\{\mathbf{a} \in \mathbb{R}^M : C\mathbf{a} = \mathbf{d}(t)\right\}.$$

Its tangent space is $\ker C$. We compute a numerical basis for this nullspace from the full singular value decomposition (SVD)

$$C = U\Sigma V^\top. \tag{2}$$

Let $p = \min(N_b, M)$, $V = [\mathbf{v}_1, \dots, \mathbf{v}_M] \in \mathbb{R}^{M\times M}$, and order the singular values as $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$. Using the numerical rank threshold $r_C = \#\{j \leq p : \sigma_j > \tau_C \max(\sigma_1, 1)\}$, we define

$$Z = [\mathbf{v}_{r_C+1}, \dots, \mathbf{v}_M] \in \mathbb{R}^{M\times m}, \qquad m = M - r_C.$$

Then $CZ \approx \mathbf{0}$, so the columns of Z span the coefficient directions left unconstrained by the sampled boundary operator[32]. Unless otherwise stated, we use $\tau_C = 10^{-10}$.

The Galerkin metric is defined on the physical domain Ω. Let $\{(\mathbf{x}_q, w_q)\}_{q=1}^{N_\mathrm{q}}$ be volume quadrature points and weights, define $(\Phi_\Omega)_{qj} = \phi_j(\mathbf{x}_q)$, and set $W_\Omega = \mathrm{diag}(w_1, \dots, w_{N_\mathrm{q}})$ and $M_\Omega = \Phi_\Omega^\top W_\Omega \Phi_\Omega$. Here N_q denotes the number of volume quadrature points, whereas N_Ω below denotes the boundary-adapted coordinate matrix. The raw nullspace basis Z may contain directions with negligible $L^2(\Omega)$ mass or poor scaling. We therefore apply the standard Gram-matrix normalization[32,33]. Form $G = Z^\top M_\Omega Z \in \mathbb{R}^{m\times m}$ and compute $G = P\Lambda P^\top$, where $P = [\mathbf{p}_1, \dots, \mathbf{p}_m]$ is orthogonal and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_m)$, with $\lambda_1 \geq \cdots \geq \lambda_m \geq 0$. We retain the eigenvalues satisfying $\lambda_j > \tau_M \max(\lambda_1, 1)$. If r directions remain, let $P_r = [\mathbf{p}_1, \dots, \mathbf{p}_r]$ and $\Lambda_r = \mathrm{diag}(\lambda_1, \dots, \lambda_r)$. The boundary-adapted coordinate matrix is

$$N_\Omega = ZP_r\Lambda_r^{-1/2} \in \mathbb{R}^{M\times r}. \tag{3}$$

Here r is the retained reduced-space dimension reported in the numerical experiments. We use $\tau_M = 10^{-11}$ unless otherwise stated. By construction,

$$CN_\Omega \approx \mathbf{0}, \qquad N_\Omega^\top M_\Omega N_\Omega = I_r.$$

Thus every homogeneous state $\mathbf{a} = N_\Omega \mathbf{z}$, with $\mathbf{z} \in \mathbb{R}^r$, satisfies the sampled boundary condition to the numerical rank tolerance and is orthonormal in the physical-domain mass inner product.

For nonhomogeneous data, we compute an affine lift $\mathbf{a}_{\mathrm{bc}}(t) \in \mathbb{R}^M$ satisfying $C\mathbf{a}_{\mathrm{bc}}(t) \approx \mathbf{d}(t)$. Every admissible numerical state is then represented as

$$\mathbf{a}(t) = \mathbf{a}_{\mathrm{bc}}(t) + N_\Omega \mathbf{z}(t), \qquad \mathbf{z}(t) \in \mathbb{R}^r.$$

For homogeneous data, $\mathbf{a}_{\mathrm{bc}}(t) = \mathbf{0}$. The target boundary condition is therefore built into the state coordinates: frozen ambient blocks are evaluated only at boundary-admissible coefficient states, and their responses are restricted to the same tangent space in the next subsection.

This construction uses only boundary samples, volume quadrature and numerical linear algebra. No neural parameter is updated for a new domain or boundary condition.

Galerkin realization and rollout of frozen blocks

The preceding subsections provide two ingredients: the frozen ambient library $\mathcal{B}_\theta^Q = \{\mathbf{F}_i^\theta\}_{i=1}^{N_{\mathrm{lib}}}$ and the boundary-adapted coordinate map $\mathbf{a}(t) = \mathbf{a}_{\mathrm{bc}}(t) + N_\Omega \mathbf{z}(t)$. We now realize the reusable mechanisms directly on the target domain.

Recall that the target equation is decomposed as

$$\partial_t u(\mathbf{x}, t) = \sum_{i=1}^{N_{\mathrm{blk}}} c_i L_i^\Omega(u)(\mathbf{x}, t), \qquad \mathbf{x} \in \Omega,$$

where c_i is the prescribed scalar weight of the ith mechanism.

At the admissible state $\mathbf{a}_\star(\mathbf{z}, t) = \mathbf{a}_{\mathrm{bc}}(t) + N_\Omega \mathbf{z}$, a default block returns the complete ambient coefficient response $\mathbf{F}_i^\theta(\mathbf{a}_\star) \in \mathbb{R}^M$. Let $\mathbf{q}_i^\Omega(\mathbf{z}, t) \in \mathbb{R}^r$ denote the reduced coordinates of its M_Ω-orthogonal projection onto range(N_Ω). Equivalently, $N_\Omega \mathbf{q}_i^\Omega(\mathbf{z}, t)$ is defined by the Galerkin orthogonality condition

$$N_\Omega^\top M_\Omega \left[N_\Omega \mathbf{q}_i^\Omega(\mathbf{z}, t) - \mathbf{F}_i^\theta\left(\mathbf{a}_\star(\mathbf{z}, t)\right) \right] = \mathbf{0}.$$

Using $N_\Omega^\top M_\Omega N_\Omega = I_r$ gives

$$\mathbf{q}_i^\Omega(\mathbf{z}, t) = N_\Omega^\top M_\Omega \mathbf{F}_i^\theta\left(\mathbf{a}_{\mathrm{bc}}(t) + N_\Omega \mathbf{z}\right) \in \mathbb{R}^r. \tag{4}$$

Thus Eq. (4) is the unique Galerkin coordinate vector of the ambient block response in the boundary-adapted tangent space.

For each mechanism, let $\mathbf{q}_i^\Omega(\mathbf{z}, t)$ denote its realized target-domain response. Since Galerkin restriction is linear, these responses add with the same coefficients c_i as the target PDE. Moreover, $\dot{\mathbf{a}}(t) = \dot{\mathbf{a}}_{\mathrm{bc}}(t) + N_\Omega \dot{\mathbf{z}}(t)$. Projecting this coefficient velocity onto the boundary-adapted space gives

$$N_\Omega^\top M_\Omega \left[\dot{\mathbf{a}}_{\mathrm{bc}}(t) + N_\Omega \dot{\mathbf{z}}(t) \right] = \sum_{i=1}^{N_{\mathrm{blk}}} c_i \, \mathbf{q}_i^\Omega(\mathbf{z}(t), t),$$

which yields the reduced dynamics

$$\dot{\mathbf{z}}(t) = \sum_{i=1}^{N_{\mathrm{blk}}} c_i \, \mathbf{q}_i^\Omega(\mathbf{z}(t), t) - N_\Omega^\top M_\Omega \dot{\mathbf{a}}_{\mathrm{bc}}(t). \tag{5}$$

Equation (5) is the finite-dimensional system advanced by Geometry-aware LegONet. The boundary condition is enforced by the coordinates throughout the rollout, rather than by a penalty or post-step correction.

Time integration is problem dependent. The default LegONet strategy composes the realized mechanisms as subflows, with Strang splitting as the standard second-order symmetric composition[24]. Explicit Runge–Kutta, implicit or Crank–Nicolson treatment of stiff linear terms, and IMEX updates may be used when required by the target dynamics.

Note that the same interface supports operator composition. Higher-order, coupled and auxiliary mechanisms are assembled from reusable responses $\mathbf{q}_i^{\Omega}$, rather than learned as new monolithic networks. Frozen blocks can therefore be selected, scaled, reassembled and composed on new domains without geometry-specific neural retraining.

Algorithm 1 Geometry-aware realization and rollout of frozen LegONet blocks

Require: Ambient domain Q, cutoff K, mechanism expressions $\{L_i^Q\}_{i=1}^{N_{\mathrm{lib}}}$, target domain Ω, and initial condition on Ω.

Ensure: Boundary-admissible rollout $u_h(\mathbf{x}, t)$ on Ω.

Offline: train reusable mechanisms on Q

Construct the ambient basis $\mathcal{V}_K(Q) = \mathrm{span}\{\phi_1, \ldots, \phi_M\}$.
for $i = 1, \ldots, N_{\mathrm{lib}}$ **do**
    Supervise the mechanism L_i^Q and train $\mathbf{F}_i^{\theta} : \mathbb{R}^M \to \mathbb{R}^M$.
end for
Freeze the ambient library $\mathcal{B}_{\theta}^{Q} = \{\mathbf{F}_i^{\theta}\}_{i=1}^{N_{\mathrm{lib}}}$.

Online: realize the target boundary

Assemble the boundary system in Eq. (1).
Compute N_{Ω} using Eqs. (2) and (3).
Compute $\mathbf{a}_{\mathrm{bc}}(t)$ and represent $\mathbf{a}(t) = \mathbf{a}_{\mathrm{bc}}(t) + N_{\Omega}\mathbf{z}(t)$.

Rollout: evolve only admissible coordinates

Project or fit the initial condition to obtain $\mathbf{z}(0)$.
for each time step **do**
    Evaluate the required frozen blocks $\mathbf{F}_i^{\theta}(\mathbf{a}_{\mathrm{bc}}(t) + N_{\Omega}\mathbf{z}(t))$.
    Advance the reduced ODE in Eq. (5).
end for
Reconstruct

$$u_h(\mathbf{x}, t) = \Phi_Q(\mathbf{x})\left(\mathbf{a}_{\mathrm{bc}}(t) + N_{\Omega}\mathbf{z}(t)\right), \qquad \mathbf{x} \in \Omega.$$

Algorithm 1 separates ambient pretraining, target-domain realization and boundary-admissible rollout. A standard stability argument decomposes the resulting error into boundary-adapted approximation, mechanism-response mismatch and time-discretization terms; see Supplementary Information.

Sparse law identification in boundary-adapted coordinates

We now consider the inverse problem of identifying an unknown interior law on a known target domain Ω with a prescribed boundary condition. Let $\{L_p^{\Omega}\}_{p=1}^{P}$ be a candidate set of physical mechanisms. We assume that the unknown dynamics can be approximated by

$$\partial_t u(\mathbf{x}, t) \approx \sum_{p=1}^{P} c_p L_p^{\Omega}(u)(\mathbf{x}, t), \qquad \mathbf{x} \in \Omega,$$

with a subset of the coefficients $\mathbf{c} = (c_1, \ldots, c_P)^\top \in \mathbb{R}^P$ active. Sparse observations are available as $y_{n,s} = u(\mathbf{x}_s, t_n) + \eta_{n,s}$, where $\{\mathbf{x}_s\}_{s=1}^{N_{\mathrm{obs}}} \subset \Omega$ are interior sensors, $\{t_n\}_{n=1}^{N_t} \subset [0, T_{\mathrm{id}}]$ is a short identification window, and $\eta_{n,s}$ denotes observation noise. The goal is to recover the active mechanisms and their coefficients, then use the identified law for prediction beyond T_{id}.

Let $\mathcal{D}_\theta^Q$ denote the frozen ambient dictionary corresponding to the P candidate mechanisms. Each candidate is realized on Ω using the default coefficient response in Eq. (4) or the mechanism-dependent procedure in Remark 2. This gives the target-domain response dictionary

$$\mathcal{D}_\theta^\Omega = \left\{ \mathbf{q}_p^\Omega : \mathbb{R}^r \times [0, T_{\mathrm{id}}] \to \mathbb{R}^r \right\}_{p=1}^P .$$

The unknown law in boundary-adapted coordinates is therefore

$$\dot{\mathbf{z}}(t) + N_\Omega^\top M_\Omega \dot{\mathbf{a}}_{\mathrm{bc}}(t) \approx \sum_{p=1}^P c_p \, \mathbf{q}_p^\Omega(\mathbf{z}(t), t).$$

To infer the reduced trajectory, define the sensor matrix by $(\Phi_{\mathrm{obs}})_{sj} = \phi_j(\mathbf{x}_s)$, and let $\mathbf{y}_n = (y_{n,1}, \ldots, y_{n,N_{\mathrm{obs}}})^\top$. Each observation is mapped into the boundary-adapted coordinates through

$$\mathbf{z}_n = \arg\min_{\mathbf{z} \in \mathbb{R}^r} \left\| \Phi_{\mathrm{obs}} \left(\mathbf{a}_{\mathrm{bc}}(t_n) + N_\Omega \mathbf{z} \right) - \mathbf{y}_n \right\|_2^2 . \tag{6}$$

Temporal derivatives $\dot{\mathbf{z}}_n$ are estimated from the short reduced trajectory. The mechanism coefficients are then identified by a regularized linear least-squares problem,

$$\widehat{\mathbf{c}} = \arg\min_{\mathbf{c} \in \mathbb{R}^P} \sum_{n=1}^{N_t} \left\| \dot{\mathbf{z}}_n + N_\Omega^\top M_\Omega \dot{\mathbf{a}}_{\mathrm{bc}}(t_n) - \sum_{p=1}^P c_p \, \mathbf{q}_p^\Omega(\mathbf{z}_n, t_n) \right\|_2^2 + \lambda_{\mathrm{ridge}} \mathcal{R}(\mathbf{c}). \tag{7}$$

Here $\lambda_{\mathrm{ridge}} \geq 0$ controls the regularization penalty $\mathcal{R}(\mathbf{c})$. If the active mechanisms are prescribed, we solve the least-squares problem on that subset. If the active subset is unknown, we use sequential thresholded ridge least squares, with column normalization and threshold–refit iterations[26,27,34] specified in Supplementary Information.

The identified coefficients immediately define a predictive reduced model:

$$\dot{\mathbf{z}}(t) = \sum_{p=1}^P \hat{c}_p \, \mathbf{q}_p^\Omega(\mathbf{z}(t), t) - N_\Omega^\top M_\Omega \dot{\mathbf{a}}_{\mathrm{bc}}(t). \tag{8}$$

No separate geometry-specific discretization is required after regression. Identification and rollout use the same boundary-adapted coordinates and the same realized mechanism responses.

This formulation targets sparse, randomly distributed and noisy observations over a short time interval. Sensors are used only to infer $\mathbf{z}_n$; the spatial mechanism features are supplied by the frozen dictionary. By contrast, the pointwise finite-difference baseline requires dense geometry-matched observations to construct stable derivative features. The numerical experiments test this distinction directly.

Remark 2 (Dictionary content) The dictionary contains candidate physical mechanisms. Its entries may be frozen learned blocks or fixed analytic Galerkin maps. All candidates enter the regression through the same reduced response $\mathbf{q}_p^{\Omega}$. The same ambient dictionary can therefore be reused across target domains while only its deterministic realization changes.

Algorithm 2 Sparse law identification in boundary-adapted coordinates

Require: Frozen ambient dictionary $\mathcal{D}_\theta^Q$, precomputed target interface N_Ω, M_Ω, affine lift $\mathbf{a}_{\mathrm{bc}}(t)$, observations $\{\mathbf{y}_n\}_{n=1}^{N_t}$.
Ensure: Identified coefficients $\hat{\mathbf{c}}$ and target-domain rollout.
1: Build the sensor matrix Φ_{obs}.
2: **for** $n = 1, \ldots, N_t$ **do**
3: Infer $\mathbf{z}_n$ using Eq. (6).
4: **end for**
5: Estimate $\dot{\mathbf{z}}_n$ from the short reduced trajectory.
6: **for** $n = 1, \ldots, N_t$ and $p = 1, \ldots, P$ **do**
7: Assemble $\mathbf{q}_p^{\Omega}(\mathbf{z}_n, t_n)$ through the target-domain response interface.
8: **end for**
9: Solve Eq. (7) for $\hat{\mathbf{c}}$.
10: Roll out the identified law using Eq. (8).
11: Reconstruct $u_h(\mathbf{x}, t) = \Phi_Q(\mathbf{x})(\mathbf{a}_{\mathrm{bc}}(t) + N_\Omega \mathbf{z}(t))$ on Ω.

Algorithm 2 converts sparse short-time observations into a predictive law using the same boundary-adapted mechanism responses as the forward solver.

Relation to classical Galerkin and geometry-aware neural operators

Having specified the forward and inverse formulations, we now clarify how the proposed framework relates to classical null-space Galerkin methods and modern geometry-aware neural operators. Geometry-aware LegONet combines a classical boundary-compatible coordinate construction with a reusable library of learned physical mechanisms. The null-space idea itself is not new. When the governing PDE is fully known, one may construct a boundary-compatible basis, assemble the required operators in that basis and solve the resulting Galerkin system. For a single prescribed problem, such a classical solver may be the most direct choice.

The difference lies in how the interior mechanisms are represented and reused. In a generic boundary-compatible basis, an energy-based Galerkin discretization commonly takes the form

$$\dot{\mathbf{z}} = -\nabla_{\mathbf{z}} E(\mathbf{z}),$$

where $\nabla_{\mathbf{z}} E(\mathbf{z})$ denotes the energy gradient in coefficient space. Its evaluation still requires deriving and assembling the variational response associated with the chosen basis. For a large, dense or poorly conditioned basis, this step can be costly or numerically sensitive. Classical methods can reduce this cost through orthogonalization or matrix factorization, but the operator construction remains tied to the specified equation and representation.

In gLegONet, the physical mechanisms are pretrained in a shared ambient coefficient space using generator-based blocks,

$$\mathbf{F}_i^{\theta}(\mathbf{a}) = -G_i \nabla_{\mathbf{a}} E_i^{\theta}(\mathbf{a}) + J_i \nabla_{\mathbf{a}} H_i^{\theta}(\mathbf{a}) + \mathbf{R}_i(\mathbf{a}).$$

Their dissipative or conservative structure is encoded through the positive-semidefinite and skew-symmetric operators, while automatic differentiation evaluates the learned generator gradients. For each target geometry, the null-space basis is mass-orthonormalized so that

$$N_{\Omega}^{\top} M_{\Omega} N_{\Omega} = I.$$

After this one-time geometry-dependent construction, the frozen mechanisms produce their reduced responses directly. No repeated reduced mass-matrix solve or new geometry-specific derivation of the learned variational response is required. The main advantage is therefore not the null-space algebra alone, but its use as an interface through which learned mechanisms can be reused, composed and evaluated on new domains.

Modern geometry-aware neural operators address a complementary problem. They usually learn an end-to-end solution map conditioned on geometry or boundary information, using training data sampled from a prescribed distribution of domains. The present experiments instead test whether a fixed mechanism library can be realized on previously unseen domains without geometry-specific neural training. Introducing a geometry-aware neural-operator baseline would require defining and training on a geometry distribution, and would therefore test geometry-conditioned amortization rather than the zero-shot realization of a fixed mechanism library studied here. We do not claim that gLegONet is universally superior to these approaches. Geometry-aware neural operators amortize solution prediction over a learned family of geometries, whereas gLegONet amortizes the learning of physical mechanisms and handles each new geometry through deterministic boundary-adapted assembly.

The framework therefore draws complementary strengths from neural and classical computation. The Galerkin coordinate construction provides algebraic boundary enforcement and efficient reduced evolution. The learned blocks provide data-driven, structure-aware mechanism representations that can be reused when analytic derivation or repeated geometry-specific assembly is difficult. The same realized mechanisms can then support both forward simulation and physical-law identification.

Use of AI-assisted tools

A large language model, ChatGPT, was used during manuscript preparation to assist with language editing, consistency checks and the identification of passages that could benefit from further clarification. All suggested revisions were critically reviewed by the authors. The authors take full responsibility for the scientific claims, mathematical derivations, numerical results and final content of the manuscript. The AI tool was not used to generate research data or perform the reported numerical experiments.

Data availability

Data used to generate the numerical results in this study are available in the GitHub repository: https://github.com/Yooki-YueqiWang/gLegONet.

Code availability

The code used in this study, including Python scripts for data analysis, is available at https://github.com/Yooki-YueqiWang/gLegONet.

References

[1] Sirignano, J. & Spiliopoulos, K. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics* **375**, 1339–1364 (2018).

[2] Raissi, M., Perdikaris, P. & Karniadakis, G. E. 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).

[3] Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence* **3**, 218–229 (2021).

[4] Li, Z. *et al.* Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv preprint arXiv:2010.08895* (2020).

[5] Kovachki, N. *et al.* Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research* **24**, 1–97 (2023).

[6] Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A. & Battaglia, P. W. Learning mesh-based simulation with graph networks. *International Conference on Learning Representations* (2021).

[7] LeVeque, R. J. *Finite Difference Methods for Ordinary and Partial Differential Equations* (SIAM, Philadelphia, PA, 2007).

[8] Brenner, S. C. & Scott, L. R. *The Mathematical Theory of Finite Element Methods* (Springer, New York, NY, 2008).

[9] Boyd, J. P. *Chebyshev and Fourier Spectral Methods* Second edn (Dover Publications, Mineola, NY, 2001).

[10] Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nature Reviews Physics* **3**, 422–440 (2021).

[11] Sukumar, N. & Srivastava, A. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering* **389**, 114333 (2022).

[12] Rahman, M. A., Ross, Z. E. & Azizzadenesheli, K. U-NO: U-shaped Neural Operators. *Transactions on Machine Learning Research* (2023).

[13] Li, Z., Huang, D. Z., Liu, B. & Anandkumar, A. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research* **24**, 1–26 (2023).

[14] Liu, N., Jafarzadeh, S. & Yu, Y. Domain agnostic fourier neural operators. *Advances in neural information processing systems* **36**, 47438–47450 (2023).

[15] Li, Z. *et al.* Geometry-informed neural operator for large-scale 3D PDEs. *Advances in Neural Information Processing Systems* **36**, 35836–35854 (2023).

[16] Hao, Z. *et al.* GNOT: A general neural operator transformer for operator learning. *Proceedings of Machine Learning Research* **202**, 12556–12569 (2023).

[17] Wu, H., Luo, H., Wang, H., Wang, J. & Long, M. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366* (2024).

[18] Chen, W., Fu, Y., Penwarden, M., Roy, P. & Stinis, P. ArGEnT: Arbitrary Geometry-encoded Transformer for Operator Learning. *arXiv preprint arXiv:2602.11626* (2026).

[19] Zhong, W. & Meidani, H. Physics-informed geometry-aware neural operator. *Computer Methods in Applied Mechanics and Engineering* **434**, 117540 (2025).

[20] Wang, H., Li, J., Dwivedi, A., Hara, K. & Wu, T. BENO: Boundary-embedded neural operators for elliptic pdes. *arXiv preprint arXiv:2401.09323* (2024).

[21] Mousavi, S., Mishra, S. & De Lorenzis, L. Imposing Boundary Conditions on Neural Operators via Learned Function Extensions. *arXiv preprint arXiv:2602.04923* (2026).

[22] Shaffer, B. D., Koohy, S., Kinch, B., Hsieh, M. A. & Trask, N. Structure-Preserving Learning Improves Geometry Generalization in Neural PDEs. *arXiv*

preprint arXiv:2602.02788 (2026).

[23] Zhang, J., Wang, Y. & Lin, G. LegONet: Plug-and-play structure-preserving neural operator blocks for compositional PDE learning. *arXiv preprint arXiv: 2603.07882* (2026).

[24] Strang, G. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis* **5**, 506–517 (1968).

[25] Hairer, E., Lubich, C. & Wanner, G. *Geometric Numerical Integration* (Springer, New York, NY, 2006).

[26] Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences* **113**, 3932–3937 (2016).

[27] Rudy, S. H., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Data-driven discovery of partial differential equations. *Science advances* **3**, e1602614 (2017).

[28] Messenger, D. A. & Bortz, D. M. Weak SINDy for partial differential equations. *Journal of Computational Physics* **443**, 110525 (2021).

[29] Trefethen, L. N. *Spectral methods in MATLAB* (SIAM, Philadelphia, PA, 2000).

[30] Canuto, C., Hussaini, M. Y., Quarteroni, A. & Zang, T. A. *Spectral methods* Vol. 285 (Springer, New York, NY, 2006).

[31] Cohen, A. *Numerical analysis of wavelet methods* Vol. 32 (Elsevier, Amsterdam, 2003).

[32] Golub, G. H. & Van Loan, C. F. *Matrix computations* (JHU press, Baltimore, MD, 2013).

[33] Trefethen, L. N. & Bau, D. *Numerical linear algebra* (SIAM, Philadelphia, PA, 2022).

[34] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **58**, 267–288 (1996).

Acknowledgements

This work was supported by the National Science Foundation (DMS-2533878, DMS-2053746, DMS-2134209, ECCS-2328241, CBET-2347401 and OAC-2311848), and U.S. Department of Energy (DOE) Office of Science Advanced Scientific Computing Research program, under the "Uncertainty Quantification for Multifidelity Operator Learning (MOLUcQ)" project (Project No. 81739), DE-SC0023161, the

SciDAC LEADS Institute, and DOE–Fusion Energy Science, under grant number: DE-SC0024583.

Author contributions

J.Z., Y.W. and G.L. conceived the study and developed the overall methodology. J.Z. contributed to the mathematical formulation, theoretical analysis, algorithmic development and improvement, development of the computational framework, and analysis and interpretation of the numerical results. Y.W. contributed to the mathematical formulation, theoretical analysis and methodological development, developed the numerical implementation, designed and conducted the computational experiments, analysed the results and prepared the figures. G.L. supervised the research, contributed to the methodological development and interpretation of the results, and acquired funding. J.Z. and Y.W. wrote the initial manuscript. All authors discussed the results, reviewed and revised the manuscript, and approved the final version. J.Z., Y.W. and G.L. contributed equally to this work.

Competing interests

The authors declare no competing interests.

Table 1: Baseline comparison on the five manufactured-solution benchmarks. The time column reports the target-stage cost in seconds: for PINN, FNO and UNO this is the target training time, whereas for gLegONet it is the target-domain assembly-and-rollout time, including boundary null-space construction, reduced operator assembly and time integration, with no target-domain neural training. Final, mean and max are relative $L^2(\Omega)$ errors over the rollout. $E_\partial^{\mathrm{rms}}$ is the RMS residual of the prescribed boundary operator, and E_∂^{sc} is the corresponding solution-scaled boundary residual.

Test	Method	Time (s)	Final error	Mean error	Max error	$E_\partial^{\mathrm{rms}}$	E_∂^{sc}
MMS-I	**gLegONet**	**2.12**	$\mathbf{1.76 \times 10^{-3}}$	$\mathbf{2.12 \times 10^{-3}}$	$\mathbf{3.16 \times 10^{-3}}$	$\mathbf{9.29 \times 10^{-11}}$	$\mathbf{2.97 \times 10^{-8}}$
	PINN	5.36×10^{3}	1.77×10^{-1}	1.65×10^{-1}	5.37×10^{-1}	8.43×10^{-6}	2.69×10^{-3}
	FNO	6.20×10^{2}	1.59×10^{-1}	9.20×10^{-2}	1.79×10^{-1}	3.17×10^{-4}	1.01×10^{-1}
	UNO	1.96×10^{3}	1.03×10^{-2}	1.02×10^{-2}	2.42×10^{-2}	2.07×10^{-4}	6.62×10^{-2}
MMS-II	**gLegONet**	**2.67**	$\mathbf{2.19 \times 10^{-3}}$	$\mathbf{2.98 \times 10^{-3}}$	$\mathbf{4.78 \times 10^{-3}}$	$\mathbf{1.01 \times 10^{-7}}$	$\mathbf{2.60 \times 10^{-7}}$
	PINN	4.64×10^{3}	1.64×10^{-2}	1.22×10^{-2}	1.73×10^{-2}	1.05×10^{-2}	2.68×10^{-2}
	FNO	1.49×10^{3}	3.65×10^{-2}	5.39×10^{-2}	2.01×10^{-1}	7.43×10^{-1}	1.91×10^{0}
	UNO	2.19×10^{3}	7.42×10^{-3}	6.27×10^{-3}	2.70×10^{-2}	7.39×10^{-1}	1.90×10^{0}
MMS-III	**gLegONet**	$\mathbf{1.13 \times 10^{1}}$	$\mathbf{1.11 \times 10^{-3}}$	$\mathbf{1.36 \times 10^{-3}}$	$\mathbf{4.27 \times 10^{-3}}$	$\mathbf{5.33 \times 10^{-9}}$	$\mathbf{1.04 \times 10^{-8}}$
	PINN	6.55×10^{3}	1.54×10^{-1}	4.10×10^{-2}	1.54×10^{-1}	3.40×10^{-2}	6.63×10^{-2}
	FNO	9.42×10^{2}	2.03×10^{-3}	2.25×10^{-3}	9.59×10^{-3}	1.63×10^{-2}	3.19×10^{-2}
	UNO	9.44×10^{2}	6.95×10^{-3}	7.67×10^{-3}	1.58×10^{-2}	8.94×10^{-2}	1.75×10^{-1}
MMS-IV	**gLegONet**	**8.89**	$\mathbf{5.77 \times 10^{-3}}$	$\mathbf{6.12 \times 10^{-3}}$	$\mathbf{7.70 \times 10^{-3}}$	$\mathbf{4.51 \times 10^{-10}}$	$\mathbf{1.93 \times 10^{-8}}$
	PINN	8.11×10^{3}	8.54×10^{-2}	3.47×10^{-2}	8.54×10^{-2}	1.41×10^{-3}	8.57×10^{-2}
	FNO	4.52×10^{3}	1.79×10^{-2}	9.60×10^{-3}	2.65×10^{-2}	9.37×10^{-4}	9.73×10^{-2}
	UNO	3.32×10^{3}	1.45×10^{-2}	8.43×10^{-3}	1.46×10^{-2}	1.00×10^{-3}	5.30×10^{-2}
MMS-V	**gLegONet**	$\mathbf{1.95 \times 10^{1}}$	$\mathbf{5.60 \times 10^{-4}}$	$\mathbf{5.06 \times 10^{-4}}$	$\mathbf{5.71 \times 10^{-4}}$	$\mathbf{3.91 \times 10^{-11}}$	$\mathbf{8.17 \times 10^{-10}}$
	PINN	6.56×10^{3}	1.50×10^{-3}	1.61×10^{-3}	4.28×10^{-3}	4.61×10^{-5}	7.92×10^{-4}
	FNO	2.25×10^{2}	1.14×10^{-2}	4.71×10^{-3}	1.14×10^{-2}	3.85×10^{-3}	1.22×10^{-1}
	UNO	2.26×10^{3}	1.39×10^{-2}	6.65×10^{-3}	1.39×10^{-2}	3.86×10^{-3}	1.22×10^{-1}

Table 2: Downstream baseline comparison on nonlinear target-domain rollouts. The time column reports the target-stage cost in seconds: for PINN, FNO and UNO this is the target training time, whereas for gLegONet it is the target-domain assembly-and-rollout time without target-domain neural training. Final, mean and max are relative L^2 errors against the independent FEM reference for Allen–Cahn and the finite-volume reference for Burgers. $E_\partial^{\mathrm{rms}}$ is the dense boundary residual. The diagnostic column reports the structure-related quantity: $D_{\mathrm{mass}}^{\max}$ for mass conservation in Allen–Cahn and R_E^{final} for the final kinetic-energy budget residual in Burgers.

Method	Time (s)	Final error	Mean error	Max error	$E_\partial^{\mathrm{rms}}$	Diagnostic
Volume-constrained Allen–Cahn on a disk						
gLegONet	$\mathbf{3.05\times10^{1}}$	$\mathbf{2.98\times10^{-2}}$	$\mathbf{3.18\times10^{-2}}$	$\mathbf{5.83\times10^{-2}}$	$\mathbf{3.47\times10^{-9}}$	$\mathbf{3.63\times10^{-5}}$
PINN	8.22×10^{2}	4.81×10^{-2}	4.94×10^{-2}	7.61×10^{-2}	2.49×10^{-3}	8.56×10^{-3}
FNO	1.24×10^{4}	1.19×10^{-1}	9.31×10^{-2}	1.62×10^{-1}	7.57×10^{-3}	1.37×10^{-3}
UNO	1.06×10^{4}	6.31×10^{-2}	5.39×10^{-2}	1.62×10^{-1}	3.14×10^{-2}	1.01×10^{-2}
Inner-square vector Burgers						
gLegONet	$\mathbf{3.88\times10^{1}}$	$\mathbf{1.83\times10^{-2}}$	$\mathbf{1.17\times10^{-2}}$	$\mathbf{1.83\times10^{-2}}$	$\mathbf{1.91\times10^{-13}}$	$\mathbf{3.63\times10^{-7}}$
PINN	1.31×10^{3}	5.58×10^{-2}	3.10×10^{-2}	5.58×10^{-2}	4.23×10^{-3}	4.21×10^{-2}
FNO	1.61×10^{4}	1.89×10^{-1}	1.34×10^{-1}	1.90×10^{-1}	3.88×10^{-3}	1.44×10^{-2}
UNO	1.19×10^{4}	3.62×10^{-2}	2.88×10^{-2}	3.62×10^{-2}	2.10×10^{-3}	2.55×10^{-2}
Direct inner-square	3.25×10^{1}	1.83×10^{-2}	1.17×10^{-2}	1.83×10^{-2}	1.17×10^{-14}	2.59×10^{-8}

Table 3: Resolution ablations for the Navier–Stokes cylinder-wake and clamped Swift–Hohenberg benchmarks. Final, mean and maximum errors are relative $L^2(\Omega)$ errors against the corresponding reference solutions. $E_\partial^{\mathrm{rms}}$ denotes the RMS dense-boundary residual over saved rollout states, and Rank is the retained boundary-adapted dimension.

Resolution	Final error	Mean error	Maximum error	$E_\partial^{\mathrm{rms}}$	Rank
Navier–Stokes					
$K = 22$	5.36×10^{-2}	1.07×10^{-1}	1.77×10^{-1}	1.41×10^{-11}	1375
$K = 31$	4.04×10^{-2}	5.58×10^{-2}	8.80×10^{-2}	7.17×10^{-12}	2811
$K = 39$	3.84×10^{-2}	4.99×10^{-2}	7.54×10^{-2}	1.31×10^{-12}	4545
$K = 40$	3.83×10^{-2}	4.97×10^{-2}	7.50×10^{-2}	1.44×10^{-12}	4789
$K = 45$	3.85×10^{-2}	4.96×10^{-2}	7.44×10^{-2}	1.59×10^{-12}	6099
Swift–Hohenberg					
$K = 22$	1.47×10^{-2}	1.76×10^{-2}	4.62×10^{-2}	1.53×10^{-8}	445
$K = 24$	1.08×10^{-2}	1.33×10^{-2}	3.66×10^{-2}	8.87×10^{-9}	522
$K = 26$	9.43×10^{-3}	1.15×10^{-2}	2.98×10^{-2}	3.44×10^{-9}	610
$K = 28$	8.39×10^{-3}	9.87×10^{-3}	2.43×10^{-2}	4.06×10^{-9}	698
$K = 30$	7.01×10^{-3}	8.33×10^{-3}	2.08×10^{-2}	3.51×10^{-9}	795

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- gLegONetforsupplementary.pdf