ANALYTICAL NEURAL OPERATOR (ANNO): LEARNING REUSABLE ANALYTICAL STRUCTURES FOR TIME EVOLUTION IN PDE FAMILIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators have emerged as an efficient framework for learning solution operators of PDE families. For time-dependent PDEs, existing learning-based solvers typically learn temporal responses from transient trajectories, either by predicting full spatiotemporal solution fields or by learning recursive propagators. For systems governed by known PDEs, however, the temporal evolution is already constrained by the governing equations. We propose the Analytical Neural Operator (ANNO), which learns reusable analytical structures from steady-state problems and reuses them for time evolution within PDE families. ANNO explicitly represents the lifting function, spatial eigenmodes, and modal coefficients, and constrains the learned modes through the corresponding spatial eigenproblem. After steady-state training, the Learnable Mode Network and Learnable Coefficient Network are frozen and reused for time-dependent PDEs in the same family, while the dynamic modal coefficients are advanced solely according to the target PDE dynamics. This requires neither transient training trajectories nor network retraining. Across five experiments, ANNO accurately recovers steady-state solutions and spatial eigenstructure, and reuses the learned analytical structure for 1D diffusion, 2D wave, and 2D engineering thermal dynamics under nonzero initial conditions, spatiotemporally varying sources, time-varying boundaries, and varying physical parameters. On these transient tasks, ANNO achieves mean relative L_2 errors of 0.2531%, 1.8646%, and 1.0506%, respectively, while substantially outperforming fully supervised transient baselines trained on complete trajectories. Ablation studies further support the effectiveness and computational efficiency of the proposed analytical structure. These results show that reusable analytical structures learned from steady-state problems can support accurate and data-efficient time evolution within PDE families.

1 INTRODUCTION

Neural operators learn mappings between function spaces and provide an efficient framework for solving families of PDEs Kovachki et al. (2023). Representative methods, such as DeepONet Lu et al. (2021) and the Fourier Neural Operator (FNO) Li et al. (2021), have shown strong approximation capability across varying inputs and problem settings. Physics-informed methods, including PINNs Raissi et al. (2019) and deep energy methods Nguyen-Thanh et al. (2020), further incorporate governing equations, boundary conditions, or variational principles to reduce reliance on labeled solution data. For time-dependent PDEs, existing learning-based solvers usually learn temporal responses from transient trajectories. Some methods directly learn mappings from initial and boundary conditions, forcing, and problem parameters to full spatiotemporal solution fields Li et al. (2021); Diab & Al Kobaisi (2025); Yu (2026), while others learn temporal propagators and recursively advance the system state Liu & Cai (2022); Brandstetter et al. (2023). In both cases, time evolution is treated as part of the learned mapping.

For systems governed by known PDEs, however, the temporal dynamics are already specified by the equations. Classical PDE theory provides an explicit *analytical structure* that separates spatial representation from temporal evolution Evans (2022); Strauss (2007). For a class of linear boundary-value problems, the solution can be written as a lifting function plus a modal expansion, where the

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

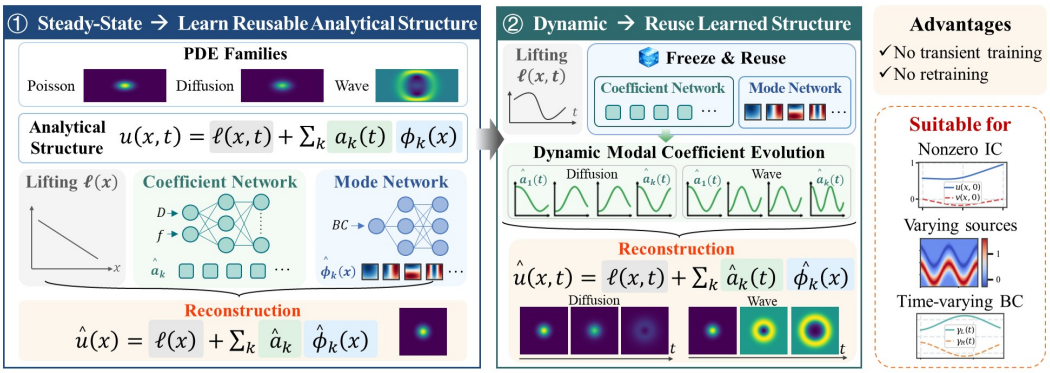

Figure 1: **Overview of ANNO.** ANNO learns reusable analytical structures from steady-state PDEs, including the lifting function, modal coefficients, and spatial eigenmodes. The learned networks are then frozen and reused for time-dependent PDEs, while the modal coefficients are evolved according to the known target-PDE dynamics. The framework supports nonzero initial conditions, varying sources, and time-varying boundary conditions.

lifting function enforces nonhomogeneous boundary conditions and the spatial eigenmodes form the modal basis. PDEs with the same computational domain, spatial operator, and homogeneous boundary-condition space can share this spatial analytical structure, while their modal coefficients follow equation-specific temporal dynamics. This motivates learning reusable analytical structures and evolving them according to the known dynamics of the target PDE.

We define PDEs sharing the same computational domain, spatial operator, and homogeneous boundary-condition space as a *PDE family*. Within a family, steady-state, diffusion, and wave equations can reuse the same spatial modal structure, while different temporal dynamics produce distinct transient responses. The learning target thus shifts from a PDE-specific solver to reusable analytical structures shared within the family. The PDE families considered here include diffusion- and wave-type dynamics, covering heat transfer, mass transfer, wave propagation, and vibration.

Based on this formulation, we propose the **Analytical Neural Operator (ANNO)**, whose overall framework is illustrated in Fig. 1. ANNO represents the solution as $\hat{u}(x) = \ell(x) + \sum_{k=1}^{K} \hat{a}_k \hat{\phi}_k(x)$, where $\ell(x)$ is an analytically derived lifting function, $\hat{\phi}_k(x)$ is produced by the **Learnable Mode Network** and represents a spatial eigenmode, and $\hat{a}_k$ is predicted by the **Learnable Coefficient Network**. With eigenstructure constraints, ANNO learns components with explicit mathematical meaning. The learned spatial modes and Learnable Coefficient Network are reused across equations within the same PDE family, while the modal coefficients are evolved according to the dynamics of the target PDE.

We evaluate ANNO on one-dimensional Poisson problems, transfer from Poisson to diffusion problems, two-dimensional wave problems, two-dimensional heat diffusion problems, and ablation studies. Our main contributions are:

1. **A unified analytical structure for time evolution in PDE families.** We define PDEs sharing the same computational domain, spatial operator, and homogeneous boundary-condition space as a PDE family, and decompose their solution process into reusable analytical structures and equation-specific temporal dynamics. This formulation allows steady-state, diffusion, and wave equations to share spatial modes and steady-state modal representations, while distinct temporal dynamics generate their corresponding transient responses. It therefore shifts the learning target from solving individual PDEs to learning reusable structures at the PDE-family level. The PDE families considered here cover heat transfer, mass transfer, wave propagation, and vibration.
2. **Analytical Neural Operator for learning reusable analytical structures.** We propose the Analytical Neural Operator (ANNO), which explicitly embeds the analytical roles of the lifting function, spatial eigenmodes, and modal coefficients into a neural operator. With eigenstructure constraints, ANNO learns representations with explicit mathematical meaning that can be reused across equations in the same PDE family.

3. **Reuse of analytical structures for time evolution in PDE families.** ANNO reuses the Learnable Mode Network and Learnable Coefficient Network trained on steady-state problems, and advances the dynamic modal coefficients solely according to the target PDE dynamics to solve transient problems in the same family. This requires neither transient data preparation nor network retraining, thereby reducing the cost of time-dependent PDE solving.

2 THEORETICAL FOUNDATION

2.1 PDE FAMILIES AND PROBLEM FORMULATION

We consider PDEs that share a common spatial structure but differ in their temporal dynamics. We define PDEs with the same computational domain Ω, spatial operator, and homogeneous boundary-condition space as a *PDE family*. Within a family, the source term, initial condition, and temporal dynamics may vary, while the spatial analytical structure remains shared. The Poisson, diffusion, and wave equations considered in this work take the representative forms

$$-D\Delta u^{\mathrm{P}}(x) = f^{\mathrm{P}}(x), \qquad x \in \Omega, \tag{1}$$

$$\frac{\partial u^{\mathrm{D}}(x,t)}{\partial t} - D\Delta u^{\mathrm{D}}(x,t) = f^{\mathrm{D}}(x,t), \qquad x \in \Omega, \tag{2}$$

$$\rho\frac{\partial^2 u^{\mathrm{W}}(x,t)}{\partial t^2} + c\frac{\partial u^{\mathrm{W}}(x,t)}{\partial t} - D\Delta u^{\mathrm{W}}(x,t) = f^{\mathrm{W}}(x,t), \qquad x \in \Omega. \tag{3}$$

Although these equations have different temporal forms, they share the same spatial analytical structure when they belong to the same PDE family. We use the Poisson-diffusion pair as the main example below; the wave equation and two-dimensional extension are given in Appendices B.4 and B.5.

2.2 BOUNDARY HOMOGENIZATION AND SHARED EIGENSTRUCTURE

For nonhomogeneous boundary conditions, we introduce a lifting function ℓ satisfying the original boundary conditions and write

$$u(x,t) = \ell(x,t) + v(x,t), \tag{4}$$

where v belongs to the corresponding homogeneous boundary space. Let B_0 denote the homogeneous boundary operator, so that $B_0 v = 0$. The lifting function is obtained analytically from the boundary parameters; the 1D and 2D constructions are given in Appendices A.2 and A.3.

After homogenization, residual solutions within the same PDE family belong to the same homogeneous boundary space and therefore share the eigenvalue problem

$$-\Delta\phi_k = \lambda_k\phi_k, \qquad B_0\phi_k = 0, \tag{5}$$

where ϕ_k and λ_k are the kth spatial eigenmode and eigenvalue, respectively. Since this eigenproblem depends only on the spatial operator, domain, and homogeneous boundary conditions, the same spatial eigenmodes can be reused across equations within the family.

2.3 ANALYTICAL STRUCTURE AND TEMPORAL DYNAMICS

Using the shared spatial eigenmodes, the analytical structure of the Poisson and diffusion residual solutions can be written as

$$v^{\mathrm{P}}(x) = \sum_{k=1}^{K} a_k^{\mathrm{P}}\phi_k(x), \tag{6}$$

$$v^{\mathrm{D}}(x,t) = \sum_{k=1}^{K} a_k^{\mathrm{D}}(t)\phi_k(x). \tag{7}$$

Substituting these expansions into the governing equations and using Eq. 5 gives

$$D\lambda_k a_k^{\mathrm{P}} = q_k^{\mathrm{P}}, \tag{8}$$

$$\frac{da_k^{\mathrm{D}}}{dt} + D\lambda_k a_k^{\mathrm{D}} = q_k^{\mathrm{D}}(t), \tag{9}$$

where q_k denotes the projection of the effective residual forcing onto the kth eigenmode.

The Poisson coefficient gives the instantaneous steady-state modal response under the current forcing. Over $[t_n, t_{n+1}]$, let $a^{\mathrm{P}}_{k,n+\frac{1}{2}}$ denote the corresponding instantaneous steady-state coefficient. The diffusion coefficient can then be advanced analytically as

$$a^{\mathrm{D}}_{k,n+1} = e^{-D\lambda_k \Delta t} a^{\mathrm{D}}_{k,n} + \left(1 - e^{-D\lambda_k \Delta t}\right) a^{\mathrm{P}}_{k,n+\frac{1}{2}}. \tag{10}$$

Thus, the spatial eigenmodes and steady-state modal representation form a reusable analytical structure, while only the modal coefficients are advanced according to the target PDE dynamics. For wave equations, the same spatial structure leads to second-order modal dynamics; the analytical update and 2D extension are given in Appendices B.4 and B.5.

3 Analytical Neural Operator

Overall Architecture. As shown in Fig. 1, ANNO embeds the analytical structure established in Section 2 into a neural operator:

$$\hat{u}(x) = \ell(x) + \sum_{k=1}^{K} \hat{a}_k \hat{\phi}_k(x), \tag{11}$$

where $\ell(x)$ is computed by the **Analytical Lifting Module**, $\hat{\phi}_k(x)$ by the **Learnable Mode Network**, and $\hat{a}_k$ by the **Learnable Coefficient Network**. These components correspond to the lifting function, spatial eigenmodes, and modal coefficients in the analytical solution.

Analytical Lifting Module. Following Section 2, the Analytical Lifting Module analytically removes the nonhomogeneous boundary contribution and obtains the homogeneous residual solution v through Eq. 4. It contains no learnable parameters. The constructions are given in Appendix A.

Learnable Mode Network. The Learnable Mode Network learns spatial eigenmodes from homogeneous boundary information. For 1D problems, the kth mode is parameterized as

$$\hat{\phi}_k(x) = \hat{A}_k \cos(\hat{\omega}_k \bar{x}) + \hat{B}_k \sin(\hat{\omega}_k \bar{x}), \tag{12}$$

where $\bar{x} = (x - x_L)/(x_R - x_L)$, and $\hat{A}_k$, $\hat{B}_k$, and $\hat{\omega}_k$ are learned parameters. This form follows the trigonometric structure of the 1D eigenproblem. To avoid mode overlap, ANNO assigns increasing frequency bands to successive modes and normalizes the resulting modes. Details in Appendix C.1.

Learnable Coefficient Network. The Learnable Coefficient Network predicts the steady-state modal coefficients. For each learned mode, the source projection $\hat{q}_k = \langle q, \hat{\phi}_k \rangle$ represents the corresponding modal component of the source term. The network takes the normalized source projections together with the broadcast diffusion coefficient and outputs $\{\hat{a}_k\}_{k=1}^{K}$. Details in Appendix C.2.

Reconstruction Module. The Reconstruction Module contains no learnable parameters and combines the lifting function, learned modes, and modal coefficients according to Eq. 11.

Structure-Constrained Learning. Accurate steady-state prediction alone does not ensure that the learned modes satisfy the eigenstructure in Section 2. ANNO therefore optimizes

$$\mathcal{L} = \mathcal{L}_{\mathrm{data}} + \lambda_{\mathrm{eig}} \mathcal{L}_{\mathrm{eig}}, \tag{13}$$

where $\mathcal{L}_{\mathrm{data}}$ is the relative L_2 loss of the residual field and $\mathcal{L}_{\mathrm{eig}}$ constrains $-\Delta\hat{\phi}_k \approx \hat{\lambda}_k \hat{\phi}_k$. Coefficient-based importance weighting is further applied to emphasize modes that contribute more to reconstruction. The full loss and staged training strategy are given in Appendix C.3.

Reuse of Analytical Structures for Time Evolution. After steady-state training, the Learnable Mode Network and Learnable Coefficient Network are frozen and reused for time-dependent PDEs within the same family. At each time step, the instantaneous steady-state modal representation is obtained and evolved according to the known target-PDE dynamics: diffusion follows Eq. 10, while wave problems update modal displacement and velocity according to the corresponding second-order dynamics (Appendix B.4). The transient field is reconstructed from the updated coefficients, reused modes, and current lifting function, without retraining on transient trajectories.

Table 1: Overview of the experimental settings and objectives.

Exp.	PDE / task	Grid	Main variations	Experimental objective
1	Poisson	$(128, -, -)$	D, S, BC	Steady-state prediction and analytical-structure learning
2	Poisson $\rightarrow$ diffusion	$(128, -, -) \rightarrow (128, -, 257)$	D, S^t, BC^t, IC	Reuse for 1D diffusion
3	Poisson $\rightarrow$ wave	$(72, 48, -) \rightarrow (72, 48, 129)$	D, S^t, BC, IC	Reuse for 2D wave
4	Poisson $\rightarrow$ diffusion	$(72, 48, -) \rightarrow (72, 48, 129)$	D, S^t, BC^t, IC	Reuse for 2D engineering thermal dynamics
5	Exp. 4 ablation	Same as Exp. 4	Components	Component analysis

Grid entries follow (N_x, N_y, N_t). D, S, BC, and IC denote variations in the spatial-operator coefficient, source or excitation, boundary condition, and initial condition. Superscript t denotes temporal variation, and the arrow indicates reuse from steady-state to time-dependent problems.

4 Experiments

We evaluate ANNO on five experiments covering steady-state learning, reuse for time evolution, and ablation analysis, as summarized in Table 1. For steady-state problems, we compare ANNO with FNO and DeepONet of comparable parameter counts; for transient problems, we use a fully supervised space-time Fourier Neural Operator (ST-FNO) and POD-DeepPropNet as transient baselines. Dataset generation, model configurations, and training details are provided in Appendices D-H.

4.1 Can ANNO Accurately Recover Steady-State Solutions and Their Analytical Structure?

Experiment 1 evaluates ANNO on 1D Poisson problems with varying diffusion coefficients, source terms, and boundary conditions. We assess both solution accuracy and analytical-structure learning. Detailed settings are provided in Appendix D, with results shown in Fig. 2. Figures 2(a-b) show that ANNO maintains low errors across all eight boundary-condition combinations, with median relative L_2 errors between 0.0476% and 0.1210%, lower than those of FNO and DeepONet. Figure 2(c) compares the learned modes with numerical eigenmodes. For the hardest RR case shown, the first four modal correlations are 1.000, 0.999, 0.993, and 0.989, while the corresponding energy spectra also agree closely with the numerical reference. These results show that ANNO achieves high steady-state accuracy while recovering the underlying analytical structure, including accurate spatial modes and modal representations, which is reused in the following time-dependent experiments.

4.2 Can Learned Analytical Structure Be Reused for Diffusion Dynamics?

Experiment 2 evaluates whether the analytical structure learned from the steady-state Poisson problems in Experiment 1 can be reused for diffusion dynamics without transient training. We introduce six dynamic scenarios: **Static-0** with zero initial residual and static source and boundary conditions; **Static-NZ** with a nonzero initial condition; **Modulated** with a time-varying source amplitude; **Moving** with a spatially moving source; **Time-BC** with time-varying boundary values; and **Combined** with a nonzero initial condition, time-varying source, and time-varying boundaries. Together with eight boundary-condition combinations and varying diffusion coefficients, these scenarios form 48 types of transient problems. We compare ANNO with the fully supervised ST-FNO and POD-DeepPropNet; detailed settings are provided in Appendix E.

Figure 3(a) shows four representative cases. In **Static-NZ**, the initial field is nonzero; in **Moving**, the source moves continuously across the spatial domain; in **Time-BC**, the left and right boundary values vary with time; and in **Combined**, the initial state, source, and boundary conditions vary within the same trajectory. Despite these different dynamic variations, ANNO closely matches the numerical ground truth throughout the spatiotemporal domain with small pointwise errors. Figure 3(b) shows an overall mean relative L_2 error of 0.2531% for ANNO, compared with 22.9472% for ST-FNO and 45.5992% for POD-DeepPropNet. Figure 3(c) further shows that the ANNO error remains near the 10^{-3} level throughout the full evolution interval. Notably, ANNO uses no transient training trajectories, whereas ST-FNO and POD-DeepPropNet are trained on 1,536 complete transient trajectories, corresponding to 394,752 stored solution fields.

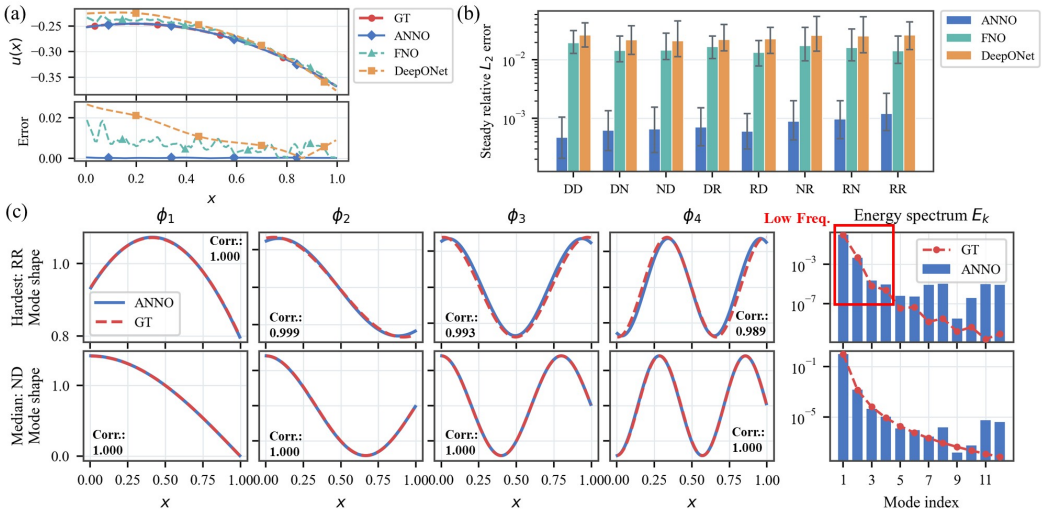

Figure 2: **Steady-state prediction and analytical-structure learning on 1D Poisson problems.** (a) Ground truth and predictions from ANNO, FNO, and DeepONet for a representative test sample. (b) Relative L_2 errors across eight boundary-condition combinations. (c) First four GT and learned eigenmodes for the hardest RR case and a median ND case, together with their modal energy spectra.

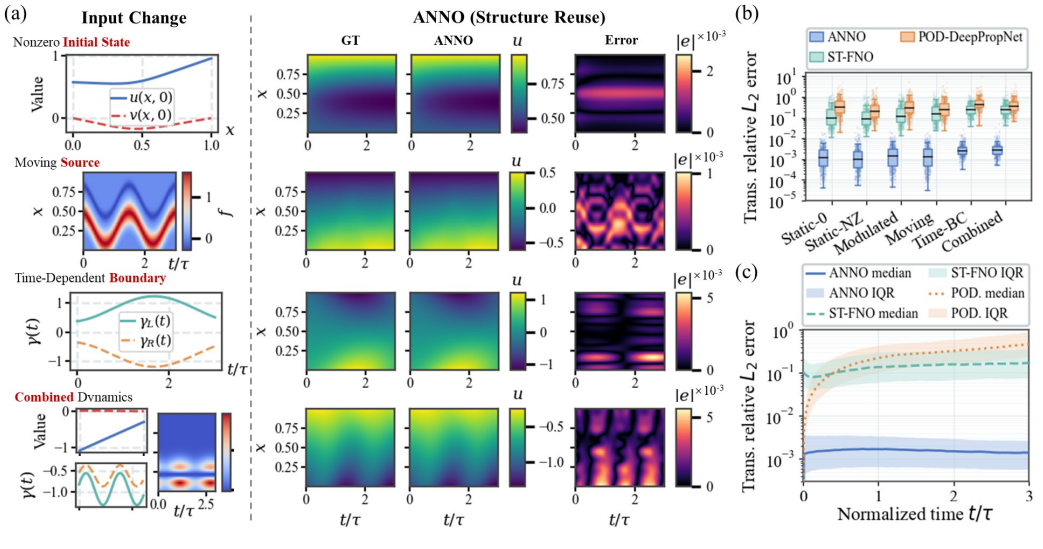

Figure 3: **Reuse of steady-state analytical structure for 1D diffusion dynamics.** (a) Four representative dynamic cases; changing inputs are shown on the left, followed by GT, ANNO prediction, and pointwise absolute error. (b) Full-trajectory relative L_2 error comparison. (c) Relative L_2 error over normalized time; curves show the median and shaded regions denote the interquartile range.

4.3 Can Learned Analytical Structure Be Reused for 2D Wave Dynamics?

Experiment 3 extends ANNO to 2D space and second-order wave dynamics. We consider a tensioned membrane with elastic supports, governed by the steady-state and transient equations

$$-T\nabla^2 w(x,y) = p(x,y), \tag{14}$$

$$\rho\frac{\partial^2 w(x,y,t)}{\partial t^2} + c\frac{\partial w(x,y,t)}{\partial t} - T\nabla^2 w(x,y,t) = p(x,y,t), \tag{15}$$

where w is the membrane displacement, T is the membrane tension, ρ and c are the areal density and damping coefficient, and p is the external load. The steady-state dataset varies the membrane

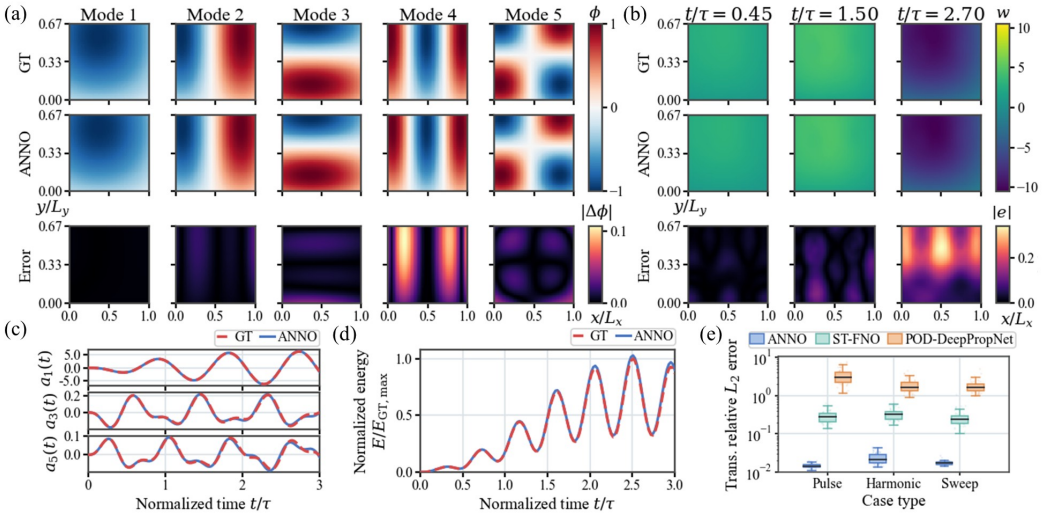

Figure 4: **2D wave dynamics with reused analytical structure.** (a) First five GT and learned spatial eigenmodes with pointwise absolute differences. (b) GT, ANNO prediction, and pointwise absolute error at three representative times. (c) Evolution of modal coefficients for Modes 1, 3, and 5. (d) Normalized mechanical-energy evolution. (e) Full-trajectory relative L_2 errors of ANNO, ST-FNO, and POD-DeepPropNet for localized-pulse, harmonic-excitation, and frequency-sweep cases.

tension, elastic supports, and spatial load distribution, while the dynamic problems include localized pulses, harmonic excitation, and frequency sweeps. Detailed settings are provided in Appendix F.

Figure 4(a) shows that the learned 2D spatial modes closely match the corresponding numerical eigenmodes. Figure 4(b) shows representative displacement fields at three times, where ANNO closely follows the numerical ground truth with small pointwise errors. The modal coefficients in Fig. 4(c) and the normalized mechanical energy in Fig. 4(d) also follow the reference evolution, showing that the reused spatial structure supports the second-order modal dynamics. Figure 4(e) compares full-trajectory errors across the three excitation cases. The overall mean relative L_2 error is 1.8646% for ANNO, compared with 30.0239% for ST-FNO and 232.2216% for POD-DeepPropNet. ANNO is trained on 5,000 steady-state solution fields and uses no transient trajectories for network training, whereas each baseline is trained on 96 complete wave trajectories with 129 stored fields per trajectory, corresponding to 12,384 transient solution fields. These results show that the analytical structure learned from steady-state problems can be reused for 2D second-order wave evolution.

4.4 Can Learned Analytical Structure Be Reused for Thermal Dynamics?

Experiment 4 evaluates ANNO on a 2D battery bottom cooling plate with nonuniform heat generation and complex cooling boundaries. The steady-state and transient temperature fields satisfy

$$-k\nabla^2 T(x,y) = Q(x,y), \tag{16}$$

$$C_v \frac{\partial T(x,y,t)}{\partial t} - k\nabla^2 T(x,y,t) = Q(x,y,t), \tag{17}$$

where T is temperature, k is thermal conductivity, C_v is volumetric heat capacity, and Q is the equivalent volumetric heat source. Water-cooled edges satisfy $k\frac{\partial T}{\partial n} + h(t)(T - T_c(t)) = 0$, where h is the heat-transfer coefficient and T_c is coolant temperature; the remaining edges are adiabatic.

The steady-state dataset varies the thermal conductivity, cooling topology and orientation, heat-transfer coefficient, coolant temperature, and cell heat-generation pattern. Five transient cases are considered: **Cold**, **Pulse**, **Periodic**, **Cooling**, and **Combined**, covering cold start, temporary power increase, periodic power variation, changing cooling conditions, and simultaneous source and cooling variations, respectively. Detailed settings are provided in Appendix G.

Figure 5(a) shows that ANNO achieves low steady-state errors across the three cooling topologies and is more accurate than FNO and DeepONet. Figure 5(b) compares transient prediction across

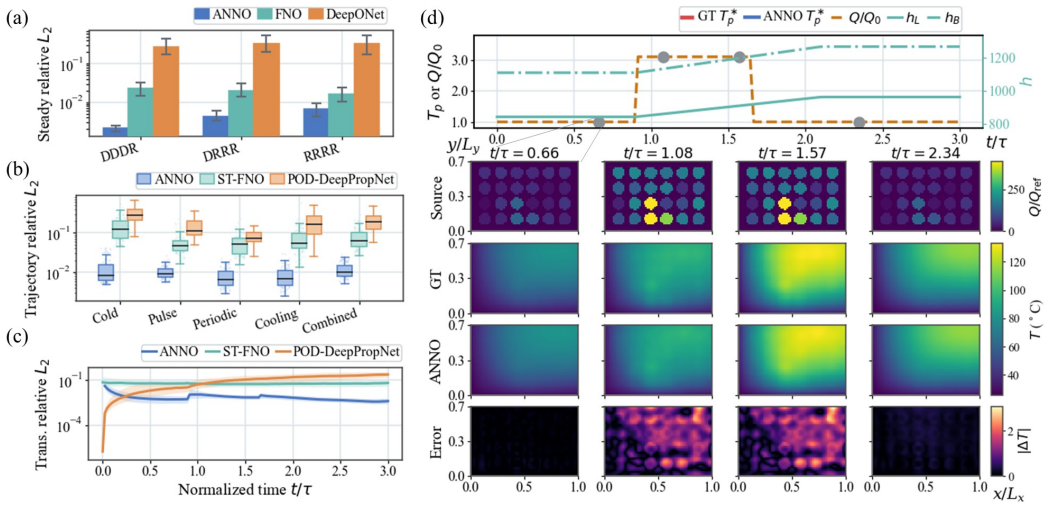

Figure 5: **ANNO on 2D engineering thermal dynamics.** (a) Steady-state relative L_2 errors under three cooling topologies. (b) Full-trajectory relative L_2 errors for five dynamic cases. (c) Relative L_2 error over normalized time. (d) Representative Combined case showing the time-varying heat source and cooling conditions, together with numerical ground truth, ANNO prediction, and pointwise absolute error at four times.

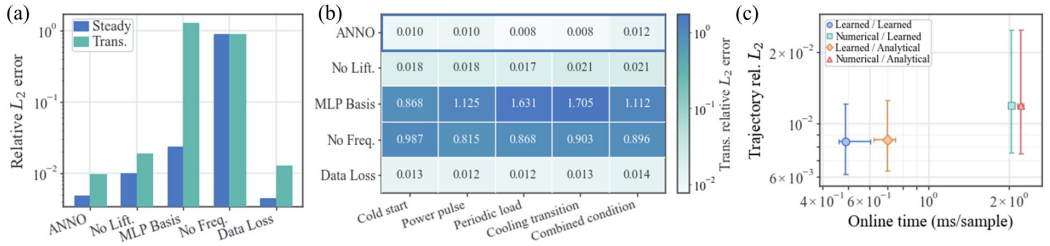

Figure 6: **ANNO ablation and modal analysis.** (a) Steady-state and full-trajectory relative L_2 errors for ablations. (b) Case-wise full-trajectory relative L_2 errors across five conditions. (c) Trajectory relative L_2 error versus online computation time for four mode–coefficient configurations.

the five dynamic cases. ANNO maintains low trajectory errors in all cases, including the **Cooling** and **Combined** cases with time-varying cooling conditions, while ST-FNO and POD-DeepPropNet exhibit substantially larger errors. Figure 5(c) further shows that the ANNO error remains low throughout the full evolution interval. Figure 5(d) visualizes a representative **Combined** case. The heat source undergoes a temporary increase while the cooling conditions change continuously, producing a strongly time-dependent temperature field. ANNO closely follows the GT at all four representative times, including during the coupled source and boundary transition. These results show that the analytical structure learned from steady-state problems can be reused for 2D engineering thermal dynamics with heterogeneous heat sources and time-varying cooling boundaries.

4.5 Ablation Study

Experiment 5 uses the Experiment 4 dataset to evaluate ANNO's main structural components and compare learned and numerical modal components. We consider the full ANNO together with **No Lift.**, which removes the Analytical Lifting Module; **MLP Basis**, which replaces the trigonometric modal representation with an MLP; **No Freq.**, which removes the frequency-band strategy; and **Data Loss**, which removes the eigenstructure constraint. We also construct a 2×2 comparison using learned or numerical spatial modes together with learned or analytical modal coefficients. Implementation details are provided in Appendix H.

Figure 6(a) shows that removing the analytical lifting, trigonometric modal representation, or frequency-band strategy degrades either steady-state or transient accuracy. In particular, **MLP Basis** and **No Freq.** produce much larger transient errors, indicating that the structured modal representation is important for reusing the learned analytical structure in time evolution. **Data Loss** retains low steady-state error but has higher transient error than the full ANNO, showing that accurate steady-state fitting alone does not guarantee reusable spatial structure. The same pattern is observed across all five dynamic conditions in Fig. 6(b). Figure 6(c) compares the four mode–coefficient configurations. The learned-mode configurations achieve comparable trajectory accuracy with substantially lower online cost. Their median trajectory errors are 0.8384% and 0.8561%, compared with 1.1962% and 1.1946% for the numerical-mode configurations, while the median online times are 0.488 and 0.701 ms per sample versus 2.041 and 2.204 ms. Numerical modes require a new eigensolve for each sample, whereas learned modes are obtained through a forward pass. These results show that reusable learned analytical structure can replace repeated numerical mode construction without sacrificing end-to-end accuracy and with much lower online cost.

5 Related Work

Neural operator research initially focused on mappings between function spaces. Building on DeepONet (Lu et al., 2021) and FNO (Li et al., 2021), later studies introduced graph, multiscale, and geometry-aware mechanisms for complex discretizations and irregular domains. The Multipole Graph Neural Operator (Li et al., 2020) uses graphs for irregular discretizations; Multiwavelet-Based Operator Learning (Gupta et al., 2021) captures multiscale features with multiwavelets; and the Geometry-Informed Neural Operator (Li et al., 2023) builds geometry-aware representations for complex domains. For time-dependent PDEs, learning-based methods typically predict full spatiotemporal fields (Li et al., 2021; Diab & Al Kobaisi, 2025; Yu, 2026) or learn recursive propagators (Liu & Cai, 2022; Brandstetter et al., 2023). Meanwhile, PINNs (Raissi et al., 2019), the Deep Ritz Method (E & Yu, 2018), DEM (Nguyen-Thanh et al., 2020), and PINO (Li et al., 2024) incorporate governing equations through PDE residuals, variational principles, or energy functionals.

More recently, researchers have incorporated structured spatial representations into operator learning. POD-DeepONet (Lu et al., 2022) uses POD modes extracted from training data as a fixed basis; Basis Operator Network (Hua & Lu, 2023) learns low-dimensional modes; the Laplacian Eigenfunction-Based Neural Operator (Wang & Hao, 2025) uses Laplacian eigenfunctions for reaction–diffusion dynamics; and the Reduced Basis Neural Operator (Qiu et al., 2026) combines a reduced space with a variational formulation. These methods show the value of structured bases, but mainly target operator approximation rather than reusable analytical structure learning.

ANNO advances this direction but differs from existing approaches. Rather than extracting a fixed empirical basis from solution snapshots or using a preconstructed spatial representation, ANNO learns an eigenstructure constrained by the homogeneous boundary space from steady-state problems and reuses it for time evolution according to the known dynamics of the target PDE.

6 Conclusion and Future Work

We propose the Analytical Neural Operator (ANNO), which explicitly embeds the analytical roles of the lifting function, spatial eigenmodes, and modal coefficients into a neural operator. ANNO learns these structures from steady-state problems under eigenstructure constraints and reuses them for time evolution within a PDE family. After steady-state training, the learned networks are frozen, and the modal coefficients are advanced according to the target PDE dynamics without transient training or parameter updates. Experiments on 1D diffusion, 2D wave, and 2D thermal dynamics show that the learned analytical structure remains effective under nonzero initial conditions, spatiotemporally varying sources, time-varying boundaries, and varying physical parameters.

The current ANNO relies on classical separable solution structures and a well-defined spatial eigenproblem. Strongly heterogeneous coefficients, complex geometries, and strongly nonlinear operators may break this structure or make a compact reusable eigenrepresentation difficult to construct, limiting the direct applicability of the current formulation. Future work will extend ANNO toward reusable approximate spatial structures for nonseparable, heterogeneous, and nonlinear PDE families while retaining the analytical organization of the solution.

REPRODUCIBILITY STATEMENT

Detailed theoretical derivations, model implementations, and experimental settings are provided in the appendices to support reproducibility. Appendices A–B provide the derivations of boundary homogenization, analytical solution structures, and analytical time evolution. Appendix C describes the ANNO architecture, structure-constrained learning, and training strategy. Appendices D–H provide the dataset construction, baseline configurations, training settings, and evaluation metrics for all experiments. The source code and scripts for reproducing the experiments will be released upon publication.

AI USE STATEMENT

In this work, we used generative AI tools to assist with English translation, language polishing, literature search, and code debugging. We did not use generative AI tools to generate experimental results or scientific conclusions, and other required disclosure tasks are not applicable to this work. All AI-assisted text was reviewed and revised by the authors, identified references were manually verified by the authors, and AI-assisted code was inspected and tested by the authors. We take responsibility for the final content of this work, including text, code, references, and other artifacts produced or modified with the aid of generative AI.

REFERENCES

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural PDE solvers, March 2023.

Waleed Diab and Mohammed Al Kobaisi. Temporal neural operator for modeling time-dependent physical phenomena. *Scientific Reports*, 15(1):32791, September 2025. ISSN 2045-2322. doi: 10.1038/s41598-025-16922-5.

Weinan E and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, March 2018. ISSN 2194-671X. doi: 10.1007/s40304-018-0127-z.

Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society, March 2022. ISBN 978-1-4704-6942-9.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24048–24062. Curran Associates, Inc., 2021.

Ning Hua and Wenlian Lu. Basis operator network: A neural network-based model for learning nonlinear operators via neural basis. *Neural Networks*, 164:21–37, July 2023. ISSN 0893-6080. doi: 10.1016/j.neunet.2023.04.017.

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. ISSN 1533-7928.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6755–6766. Curran Associates, Inc., 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, May 2021.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural*

Information Processing Systems, volume 36, pp. 35836–35854. Curran Associates, Inc., 2023. doi: 10.52202/075280-1556.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 1(3):9:1–9:27, May 2024. doi: 10.1145/3648506.

Lizuo Liu and Wei Cai. DeepPropNet – a recursive deep propagator neural network for learning evolution PDE operators, February 2022.

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, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, April 2022. ISSN 0045-7825. doi: 10.1016/j.cma.2022.114778.

Vien Minh Nguyen-Thanh, Xiaoying Zhuang, and Timon Rabczuk. A deep energy method for finite deformation hyperelasticity. *European Journal of Mechanics - A/Solids*, 80:103874, March 2020. ISSN 0997-7538. doi: 10.1016/j.euromechsol.2019.103874.

Yuan Qiu, Wolfgang Dahmen, and Peng Chen. Variationally correct operator learning: Reduced basis neural operator with a posteriori error estimation. *Computer Methods in Applied Mechanics and Engineering*, 461:119228, November 2026. ISSN 0045-7825. doi: 10.1016/j.cma.2026.119228.

M. Raissi, P. Perdikaris, and G. E. 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, February 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045.

Walter A. Strauss. *Partial Differential Equations: An Introduction*. John Wiley & Sons, December 2007. ISBN 978-0-470-05456-7.

Jindong Wang and Wenrui Hao. Laplacian eigenfunction-based neural operator for learning nonlinear reaction-diffusion dynamics. *Journal of Computational Physics*, 543:114400, December 2025. ISSN 0021-9991. doi: 10.1016/j.jcp.2025.114400.

Rixin Yu. An inverse scattering inspired fourier neural operator for time-dependent PDE learning. *Journal of Computational Physics*, 563:115081, October 2026. ISSN 0021-9991. doi: 10.1016/j.jcp.2026.115081.

APPENDIX CONTENTS

A Boundary Homogenization and Analytical Lifting 12
A.1 Unified Representation of Mixed Boundary Conditions 12
A.2 Derivation and Validation of the 1D Lifting Function 13
A.3 Derivation and Validation of the 2D Lifting Function 14

B Analytical Structure and Temporal Dynamics 17
B.1 Shared Eigenstructure for the Three Equation Classes 17
B.2 Analytical Solutions for Poisson, Diffusion, and Wave Equations 20
B.3 Analytical Time Evolution for Diffusion 23
B.4 Analytical Time Evolution for Wave Dynamics 24
B.5 Extension to Two Spatial Dimensions 25

C Implementation Details of ANNO 26
C.1 Learnable Mode Network 26
C.2 Learnable Coefficient Network 27
C.3 Structure-Constrained Learning 27

D Supplementary Details for Experiment 1 29
D.1 Dataset Construction and Experimental Configuration 29
D.2 Model Configuration and Training Settings 30
D.3 Evaluation Metrics and Quantitative Results 31

E Supplementary Details for Experiment 2 32
E.1 Dataset Construction and Experimental Configuration 32
E.2 Baseline Configuration and Training Settings 33
E.3 Evaluation Metrics and Quantitative Results 34
E.4 Additional Analysis of Transient Baselines 34

F Supplementary Details for Experiment 3 35
F.1 Problem Setup 36
F.2 Steady Dataset Construction 36
F.3 Dynamic Dataset 37
F.4 Model Configuration 38
F.5 Evaluation Metrics and Quantitative Results 38

G Supplementary Details for Experiment 4 39
G.1 Problem Setup 39
G.2 Steady Dataset Construction 39
G.3 Dynamic Dataset Construction 40
G.4 Reuse of the Steady-State Model 41
G.5 Baselines and Evaluation 41

H Supplementary Details for Experiment 5 42
H.1 Removing Analytical Lifting 42
H.2 Replacing Trigonometric Modes with an MLP 42
H.3 Removing the Frequency-Band Strategy 43
H.4 Removing the Eigenstructure Constraint 43
H.5 Learned and Numerical Modal Components 43
H.6 Evaluation and Quantitative Results 44

A Boundary Homogenization and Analytical Lifting

A.1 Unified Representation of Mixed Boundary Conditions

Let the computational domain be $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$. For boundaries aligned with the coordinate axes, we express the boundary flux using the partial derivative along the corresponding coordinate direction and absorb the sign induced by the boundary orientation directly into the derivative coefficient β. We write the boundary condition in the unified form

$$\alpha(x)u(x,t) + \beta(x)\frac{\partial u}{\partial x_j}(x,t) = \gamma(x,t), x \in \partial\Omega. \tag{18}$$

Here, x_j denotes the coordinate direction associated with the current boundary. $\alpha(x)$ and $\beta(x)$ determine the type and form of the boundary operator, while $\gamma(x,t)$ denotes the nonhomogeneous

boundary data, which may vary in space and time. With different values of (α, β), this unified form represents Dirichlet, Neumann, and Robin boundary conditions; the correspondence is listed in Table 2.

Table 2: Boundary-condition encoding in the unified three-parameter representation.

Boundary type	Common prescribed form	Three-parameter encoding (α, β, γ)
Dirichlet	$u = g$	$(1, 0, g)$
Neumann	$\frac{\partial u}{\partial n} = g$	$(0, \pm 1, g)$
Robin	$au + b\frac{\partial u}{\partial n} = g$	$(a, \pm b, g)$

Here, $\frac{\partial u}{\partial n}$ denotes the outward normal. Because we consistently write the boundary conditions using coordinate derivatives, the relation between the outward normal and the positive coordinate direction is encoded by the sign of β. For a rectangular domain, the outward normals on the left and bottom boundaries are opposite to the positive x- and y-directions, respectively, so β takes a negative sign. The outward normals on the right and top boundaries agree with the corresponding positive coordinate directions, so β takes a positive sign.

In the following experiments, we use letter abbreviations for the boundary types: D, N, and R denote Dirichlet, Neumann, and Robin boundary conditions, respectively. For a 1D domain, the two letters are ordered from smaller to larger spatial coordinate. For example, DN denotes a Dirichlet condition at the left endpoint and a Neumann condition at the right endpoint, while RR denotes Robin conditions at both endpoints. For a 2D rectangular domain, boundaries are ordered first along x and then along y, and within each coordinate direction from smaller to larger coordinate value, corresponding to the left, right, bottom, and top boundaries.

A.2 Derivation and Validation of the 1D Lifting Function

Consider a 1D domain $\Omega = [x_L, x_R]$ with two endpoint boundary conditions. When the boundary system is nonsingular, a linear function is sufficient, so we assume the lifting function has the linear form $\ell(x, t) = c_0(t) + c_1(t)(x - x_L)$. Differentiation gives

$$\ell_x(x, t) = c_1(t), \qquad \ell_{xx}(x, t) = 0 \tag{19}$$

Substituting it into the left and right boundary conditions gives

$$\alpha_L c_0(t) + \beta_L c_1(t) = \gamma_L(t), \tag{20}$$

$$\alpha_R c_0(t) + \left[\alpha_R (x_R - x_L) + \beta_R\right] c_1(t) = \gamma_R(t). \tag{21}$$

In matrix form,

$$\begin{bmatrix} \alpha_L & \beta_L \\ \alpha_R & \alpha_R (x_R - x_L) + \beta_R \end{bmatrix} \begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} = \begin{bmatrix} \gamma_L(t) \\ \gamma_R(t) \end{bmatrix}. \tag{22}$$

Define the determinant of the boundary matrix as

$$\Delta_x = \alpha_L \left[\alpha_R (x_R - x_L) + \beta_R\right] - \alpha_R \beta_L, \tag{23}$$

When $\Delta_x \neq 0$, the lifting coefficients have the explicit solutions

$$c_0(t) = \frac{[\alpha_R(x_R - x_L) + \beta_R]\gamma_L(t) - \beta_L \gamma_R(t)}{\Delta_x}, \tag{24}$$

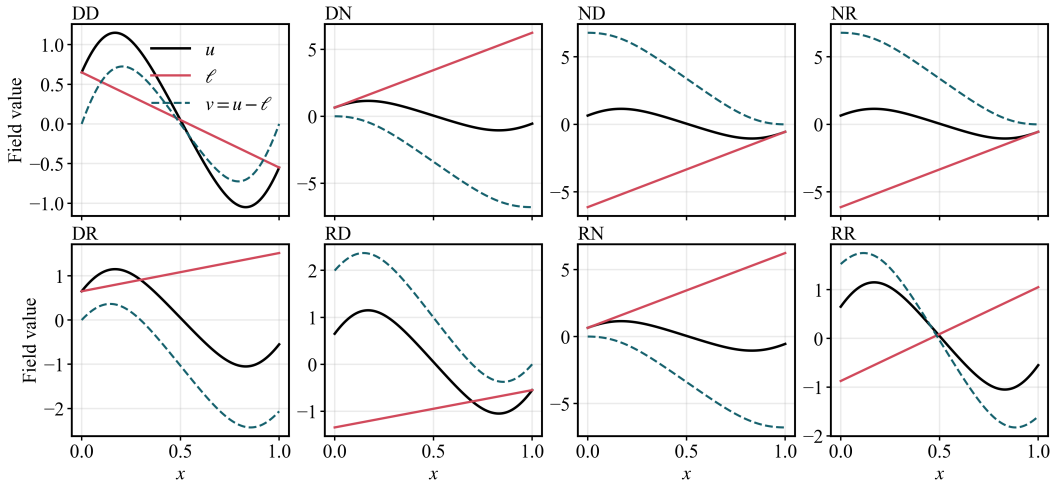

Figure 7: Numerical validation of the 1D lifting function under different boundary-condition combinations.

$$c_1(t) = \frac{\alpha_L \gamma_R(t) - \alpha_R \gamma_L(t)}{\Delta_x}. \tag{25}$$

Substituting $c_0(t)$ and $c_1(t)$ back gives the forward lifting formula without unknown intermediate variables:

$$\ell(x,t) = \frac{\left[\alpha_R\left(x_R - x\right) + \beta_R\right]\gamma_L(t) + \left[\alpha_L\left(x - x_L\right) - \beta_L\right]\gamma_R(t)}{\alpha_L\alpha_R\left(x_R - x_L\right) + \alpha_L\beta_R - \alpha_R\beta_L} \tag{26}$$

To further verify the closed-form lifting formula, we perform numerical tests on the eight nonsingular combinations of Dirichlet, Neumann, and Robin conditions, excluding the pure Neumann-Neumann case, as shown in Fig. 7. The black curve is the original solution u, the red line is the lifting function ℓ obtained from the closed-form formula, and the blue dashed curve is the homogeneous residual v. Under all boundary types, the lifting function accurately extracts the nonhomogeneous boundary contribution so that the remaining residual lies in the corresponding homogeneous boundary space. The residuals at both endpoints are zero up to floating-point roundoff, with a maximum residual of only 4.44×10^{-16}, confirming that the derived closed-form expression exactly satisfies the original boundary conditions for all nonsingular boundary combinations.

A.3 Derivation and Validation of the 2D Lifting Function

Consider a 2D rectangular computational domain $\Omega = [x_L, x_R] \times [y_B, y_T]$, and define

$$L_x = x_R - x_L, L_y = y_T - y_B. \tag{27}$$

The 2D construction can be viewed as a combination of the 1D lifting functions from Appendix A.2 along the two coordinate directions. First consider the x-direction, treating y and t as parameters. The left and right boundaries then form a 1D boundary-value problem. Using the same linear construction as in Appendix A.2, the boundary linear system is

$$\begin{bmatrix} \alpha_L & \beta_L \\ \alpha_R & \alpha_R L_x + \beta_R \end{bmatrix} \begin{bmatrix} c_0^{(x)}(y,t) \\ c_1^{(x)}(y,t) \end{bmatrix} = \begin{bmatrix} \gamma_L(y,t) \\ \gamma_R(y,t) \end{bmatrix} \tag{28}$$

The determinant of the boundary matrix in the x-direction is

$$\Delta_x = \alpha_L \alpha_R L_x + \alpha_L \beta_R - \alpha_R \beta_L. \tag{29}$$

Similarly, for the y-direction, treating x and t as parameters, the bottom and top boundaries give the linear system

$$\begin{bmatrix} \alpha_B & \beta_B \\ \alpha_T & \alpha_T L_y + \beta_T \end{bmatrix} \begin{bmatrix} c_0^{(y)}(x,t) \\ c_1^{(y)}(x,t) \end{bmatrix} = \begin{bmatrix} \gamma_B(x,t) \\ \gamma_T(x,t) \end{bmatrix} \tag{30}$$

The determinant of the boundary matrix in the y-direction is

$$\Delta_y = \alpha_B \alpha_T L_y + \alpha_B \beta_T - \alpha_T \beta_B \tag{31}$$

When $\Delta_x \neq 0$, the lifting function for the left and right boundaries along x is

$$\ell^{(x)}(x,y,t) = \frac{[\alpha_R(x_R - x) + \beta_R]\gamma_L(y,t) + [\alpha_L(x - x_L) - \beta_L]\gamma_R(y,t)}{\Delta_x}. \tag{32}$$

When $\Delta_y \neq 0$, the lifting function for the bottom and top boundaries along y is

$$\ell^{(y)}(x,y,t) = \frac{[\alpha_T(y_T - y) + \beta_T]\gamma_B(x,t) + [\alpha_B(y - y_B) - \beta_B]\gamma_T(x,t)}{\Delta_y}. \tag{33}$$

Here, $\ell^{(x)}$ already satisfies the left and right boundary conditions by itself, and $\ell^{(y)}$ satisfies the bottom and top boundary conditions. If the 2D lifting function is taken as $\ell^{(x)}(x,y,t) + \ell^{(y)}(x,y,t)$, direct superposition double-counts the corner contributions, so the extra terms must be removed.

For the x-direction, construct the left-boundary selection function

$$L(x) = \frac{\alpha_R(x_R - x) + \beta_R}{\Delta_x}. \tag{34}$$

which satisfies

$$\alpha_L L\left(x_L\right) + \beta_L L^{'}\left(x_L\right) = 1, \tag{35}$$

$$\alpha_R L(x_R) + \beta_R L^{'}(x_R) = 0. \tag{36}$$

Similarly, construct the selection functions for the right, bottom, and top boundaries

$$R(x) = \frac{\alpha_L(x - x_L) - \beta_L}{\Delta_x}, \tag{37}$$

$$B(y) = \frac{\alpha_T(y_T - y) + \beta_T}{\Delta_y}, \tag{38}$$

$$T(y) = \frac{\alpha_B(y - y_B) - \beta_B}{\Delta_y}, \tag{39}$$

Taking the lower-left corner (x_L, y_B) as an example, it must satisfy both $L(x)$ and $B(y)$, which gives the lower-left corner selection function

$$H_{LB}(x,y) = L(x)B(y). \tag{40}$$

At the same corner, the left-boundary data $\gamma_L(y,t)$ also satisfies the bottom-boundary operator, so the corresponding corner value is

$$c_{LB}(t) = \alpha_B \gamma_L(y_B, t) + \beta_B \frac{\partial \gamma_L}{\partial y}(y_B, t). \tag{41}$$

The selection functions and corner values for the other three corners are obtained analogously:

$$H_{LT}(x,y) = L(x)T(y), \tag{42}$$

$$H_{RB}(x,y) = R(x)B(y), \tag{43}$$

$$H_{RT}(x,y) = R(x)T(y). \tag{44}$$

$$c_{LT}(t) = \alpha_T \gamma_L(y_T, t) + \beta_T \gamma_{L,y}(y_T, t), \tag{45}$$

$$c_{RB}(t) = \alpha_B \gamma_R(y_B, t) + \beta_B \gamma_{R,y}(y_B, t), \tag{46}$$

$$c_{RT}(t) = \alpha_T \gamma_R(y_T, t) + \beta_T \gamma_{R,y}(y_T, t). \tag{47}$$

The total duplicated contribution from the four corners is then

$$C(x,y,t) = H_{LB}c_{LB} + H_{LT}c_{LT} + H_{RB}c_{RB} + H_{RT}c_{RT}. \tag{48}$$

The final 2D lifting function is

$$\ell(x,y,t) = \ell^{(x)}(x,y,t) + \ell^{(y)}(x,y,t) - C(x,y,t). \tag{49}$$

To validate the 2D lifting construction, Fig. 8 shows the lifting function l, homogeneous residual v, reconstructed field $\ell + v$, and the error $u_{\text{direct}} - (l+v)$ between the direct nonhomogeneous solution and the reconstruction for four representative boundary combinations. The spatial distributions of the lifting and residual change substantially across boundary conditions, but their sum accurately recovers the original solution field in every case. The lifting-reconstruction relative errors of the four cases are 5.90×10^{-5}, 2.54×10^{-4}, 1.27×10^{-4}, and 1.00×10^{-4}, showing that the constructed 2D lifting robustly separates the nonhomogeneous boundary contribution and, together with the independently solved homogeneous residual, accurately reconstructs the full solution field.

Unlike the 1D linear lifting, which satisfies $\ell_{xx} = 0$, the 2D lifting generally has $\Delta\ell \neq 0$. The homogeneous residual equation therefore requires a corrected effective source term $f_{\text{eff}} = f + D\Delta\ell$. We construct this source using the continuous analytical $\Delta\ell$ and then solve the residual with a discrete finite-volume scheme. Consequently, truncation error arises between the continuous lifting and the discrete PDE solve, so the reconstruction error on a finite grid does not reach machine precision but instead converges under grid refinement, as shown in Fig. 9.

The preceding 1D and 2D lifting derivations already allow the nonhomogeneous boundary data γ to vary with time. Thus, for time-varying boundaries, only the corresponding $\gamma(t)$ needs to be updated at each time. For a discrete time point t_n and the next time point t_{n+1}, we obtain $\ell_n = \ell[\gamma(t_n)], \ell_{n+1} = \ell[\gamma(t_{n+1})]$. However, the effective source term still requires the time derivative, which we approximate with a finite-difference formula:

$$\frac{\partial \ell}{\partial t}\Big|_{t_{n+\frac{1}{2}}} \approx \frac{\ell_{n+1} - \ell_n}{\Delta t},\ \Delta t = t_{n+1} - t_n, \tag{50}$$

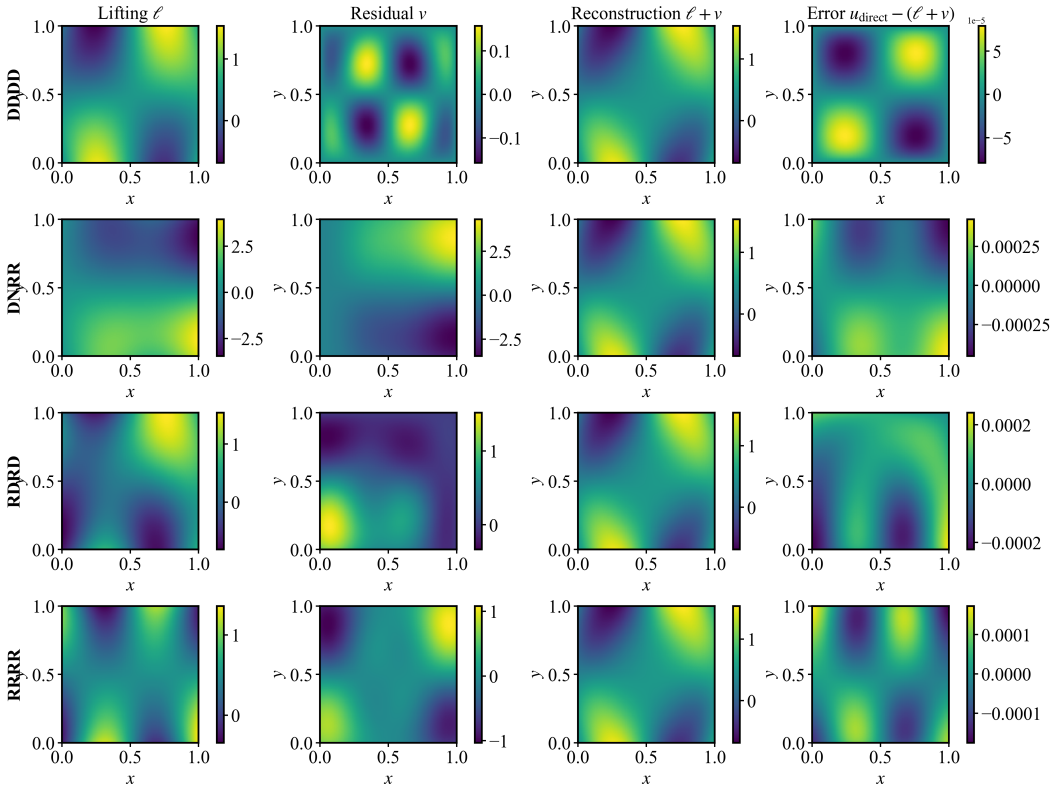

Figure 8: Numerical validation of the 2D lifting function under different boundary-condition combinations.

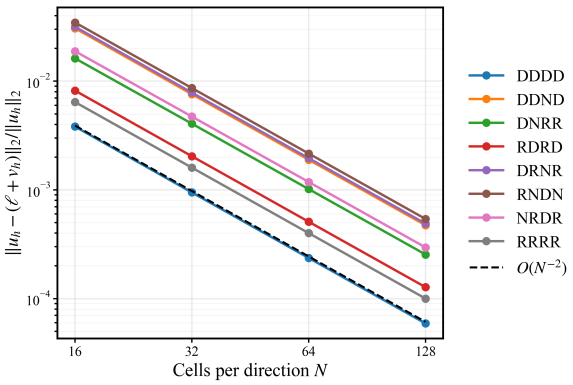

Figure 9: Grid convergence of the 2D lifting-reconstruction error under different boundary conditions.

B Analytical Structure and Temporal Dynamics

B.1 Shared Eigenstructure for the Three Equation Classes

The Poisson, diffusion, and wave problems considered in this work are written as

$$-D\Delta u^{\mathrm{P}}(x) = f^{\mathrm{P}}(x) \tag{51}$$

$$\frac{\partial u^{\mathrm{D}}(x,t)}{\partial t} - D\Delta u^{\mathrm{D}}(x,t) = f^{\mathrm{D}}(x,t) \tag{52}$$

$$\frac{\partial^2 u^{\mathrm{W}}(x,t)}{\partial t^2} + c_d \frac{\partial u^{\mathrm{W}}(x,t)}{\partial t} - D\Delta u^{\mathrm{W}}(x,t) = f^{\mathrm{W}}(x,t) \tag{53}$$

Here, $D > 0$ is a spatially constant coefficient and c_d is the damping coefficient in the wave problem. The superscripts P, D, and W denote the Poisson, diffusion, and wave problems, respectively.

For nonhomogeneous boundary conditions, Appendix A converts the full solution into a residual problem with homogeneous boundary conditions. We therefore work directly with the homogenized residual solution $v^{\mathrm{P}}, \quad v^{\mathrm{D}}, \quad v^{\mathrm{W}}$, and the corresponding effective residual forcing $q^{\mathrm{P}}, \quad q^{\mathrm{D}}, \quad q^{\mathrm{W}}$. The three residual equations are

$$-D\Delta v^{\mathrm{P}} = q^{\mathrm{P}} \tag{54}$$

$$\frac{\partial v^{\mathrm{D}}}{\partial t} - D\Delta v^{\mathrm{D}} = q^{\mathrm{D}}(x,t) \tag{55}$$

$$\frac{\partial^2 v^{\mathrm{W}}}{\partial t^2} + c_d \frac{\partial v^{\mathrm{W}}}{\partial t} - D\Delta v^{\mathrm{W}} = q^{\mathrm{W}}(x,t) \tag{56}$$

After boundary homogenization, if the three problems share the same spatial domain Ω, Laplacian spatial operator, and homogeneous boundary space, they share the eigenproblem

$$-\Delta\phi_k(x) = \lambda_k \phi_k(x), \qquad \mathcal{B}_0\phi_k = 0 \tag{57}$$

where ϕ_k is the k-th spatial eigenmode and λ_k is its eigenvalue. This eigenproblem depends only on the spatial operator, domain, and homogeneous boundary conditions, and is independent of the temporal order of the Poisson, diffusion, or wave equation.

For the self-adjoint cases considered here, the eigenmodes can be chosen orthonormally:

$$\langle \phi_i, \phi_j \rangle = \delta_{ij}. \tag{58}$$

Thus, the residuals of all three equation classes can be expanded in the same spatial eigenbasis:

$$v^{\mathrm{P}}(x) = \sum_{k=1}^{\infty} a_k^{\mathrm{P}} \phi_k(x) \tag{59}$$

$$v^{\mathrm{D}}(x,t) = \sum_{k=1}^{\infty} a_k^{\mathrm{D}}(t) \phi_k(x) \tag{60}$$

$$v^{\mathrm{W}}(x,t) = \sum_{k=1}^{\infty} a_k^{\mathrm{W}}(t) \phi_k(x) \tag{61}$$

The effective residual forcing can likewise be expanded in the same eigenspace. For example, for the Poisson problem,

$$q^{\mathrm{P}}(x) = \sum_{k=1}^{\infty} q_k^{\mathrm{P}} \phi_k(x) \tag{62}$$

where

$$q_k^{\mathrm{P}} = \left\langle q^{\mathrm{P}}, \phi_k \right\rangle \tag{63}$$

We use the Poisson problem to derive the modal-coefficient equation in detail. Substituting Eq. 59 into the Poisson residual equation 54 gives

$$-D\Delta \left(\sum_{k=1}^{\infty} a_k^{\mathrm{P}} \phi_k \right) = q^{\mathrm{P}} \tag{64}$$

Because a_k^{P} is independent of the spatial variable x, it can be taken outside the Laplacian, giving

$$\sum_{k=1}^{\infty} a_k^{\mathrm{P}} \left(-D\Delta \phi_k \right) = q^{\mathrm{P}} \tag{65}$$

Using the eigenrelation in Eq. 57,

$$-D\Delta \phi_k = D\lambda_k \phi_k \tag{66}$$

Eq. 65 can therefore be written as

$$\sum_{k=1}^{\infty} D\lambda_k a_k^{\mathrm{P}} \phi_k = q^{\mathrm{P}} \tag{67}$$

Using the modal expansion of q^{P} in Eq. 62,

$$\sum_{k=1}^{\infty} D\lambda_k a_k^{\mathrm{P}} \phi_k = \sum_{k=1}^{\infty} q_k^{\mathrm{P}} \phi_k \tag{68}$$

To isolate a single modal relation, take the inner product of both sides of Eq. 68 with an arbitrary eigenmode ϕ_j:

$$\left\langle \sum_{k=1}^{\infty} D\lambda_k a_k^{\mathrm{P}} \phi_k, \phi_j \right\rangle = \left\langle \sum_{k=1}^{\infty} q_k^{\mathrm{P}} \phi_k, \phi_j \right\rangle \tag{69}$$

By linearity,

$$\sum_{k=1}^{\infty} D\lambda_k a_k^{\mathrm{P}} \left\langle \phi_k, \phi_j \right\rangle = \sum_{k=1}^{\infty} q_k^{\mathrm{P}} \left\langle \phi_k, \phi_j \right\rangle \tag{70}$$

Using the orthonormality relation $\langle \phi_k, \phi_j \rangle = \delta_{kj}$, all terms vanish except the j-th mode, yielding

$$D\lambda_j a_j^{\mathrm{P}} = q_j^{\mathrm{P}} \tag{71}$$

Because j can denote any mode, relabeling it as k gives the modal relation for the Poisson equation:

$$D\lambda_k a_k^{\mathrm{P}} = q_k^{\mathrm{P}} \tag{72}$$

For the diffusion equation, substituting Eq. 60 and the corresponding effective-source expansion into Eq. 55, and using the eigenproblem and eigenmode orthogonality, gives the diffusion modal relation

$$\frac{da_k^{\mathrm{D}}}{dt} + D\lambda_k a_k^{\mathrm{D}} = q_k^{\mathrm{D}}(t) \tag{73}$$

Similarly, substituting Eq. 61 into Eq. 56 gives the modal relation for the wave equation

$$\frac{d^2 a_k^{\mathrm{W}}}{dt^2} + c_d \frac{da_k^{\mathrm{W}}}{dt} + D\lambda_k a_k^{\mathrm{W}} = q_k^{\mathrm{W}}(t) \tag{74}$$

B.2 Analytical Solutions for Poisson, Diffusion, and Wave Equations

B.2.1 One-Dimensional Eigenmodes

Consider the finite 1D interval

$$\Omega = [x_L, x_R], \qquad L = x_R - x_L. \tag{75}$$

The eigenproblem from the previous section becomes, in 1D,

$$-\frac{d^2\phi_k(x)}{dx^2} = \lambda_k \phi_k(x) \tag{76}$$

subject to the corresponding homogeneous boundary conditions

$$\mathcal{B}_0 \phi_k = 0 \tag{77}$$

For $\lambda_k > 0$, define

$$\omega_k = \sqrt{\lambda_k} \tag{78}$$

Then Eq. 76 becomes

$$\frac{d^2\phi_k}{dx^2} + \omega_k^2 \phi_k = 0 \tag{79}$$

Its characteristic equation is

$$r^2 + \omega_k^2 = 0 \tag{80}$$

with characteristic roots

$$r_{1,2} = \pm i\omega_k \tag{81}$$

Therefore, the general real-valued form of the k-th spatial eigenmode inside the interval is

$$\phi_k(x) = A_k \cos\left[\omega_k (x - x_L)\right] + B_k \sin\left[\omega_k (x - x_L)\right]. \tag{82}$$

where A_k, B_k, and ω_k are further determined by the boundary conditions.

Eq. 82 gives the general interior solution of the 1D constant-coefficient eigenproblem. Substituting it into the homogeneous boundary conditions yields a linear system for A_k and B_k, which determines the discrete frequencies ω_k admissible under the boundary conditions.

For example, consider homogeneous Dirichlet conditions at both endpoints:

$$\phi_k(x_L) = 0, \qquad \phi_k(x_R) = 0 \tag{83}$$

Substituting the left-end condition into Eq. 82 gives $A_k = 0$, and therefore

$$\phi_k(x) = B_k \sin\left[\omega_k (x - x_L)\right]. \tag{84}$$

Table 3: Eigenmodes for different homogeneous boundary conditions on a 1D interval.

Boundary	**Eigenmode** $\phi_k(x)$	**Admissible frequency condition**
DD	$\mathcal{N}_k \sin(\omega_k \xi)$	$\omega_k = k\pi/L, \quad k = 1, 2, \ldots$
DN	$\mathcal{N}_k \sin(\omega_k \xi)$	$\omega_k = \left(k - \frac{1}{2}\right)\pi/L$
ND	$\mathcal{N}_k \cos(\omega_k \xi)$	$\omega_k = \left(k - \frac{1}{2}\right)\pi/L$
NN	$\mathcal{N}_k \cos(\omega_k \xi)$	$\omega_k = k\pi/L, \quad k = 0, 1, \ldots$
DR	$\mathcal{N}_k \sin(\omega_k \xi)$	$\alpha_R \sin(\omega_k L) + \beta_R \omega_k \cos(\omega_k L) = 0$
RD	$\mathcal{N}_k \left[\beta_L \omega_k \cos(\omega_k \xi) - \alpha_L \sin(\omega_k \xi)\right]$	$\alpha_L \sin(\omega_k L) - \beta_L \omega_k \cos(\omega_k L) = 0$
NR	$\mathcal{N}_k \cos(\omega_k \xi)$	$\alpha_R \cos(\omega_k L) - \beta_R \omega_k \sin(\omega_k L) = 0$
RN	$\mathcal{N}_k \left[\beta_L \omega_k \cos(\omega_k \xi) - \alpha_L \sin(\omega_k \xi)\right]$	$\alpha_L \cos(\omega_k L) + \beta_L \omega_k \sin(\omega_k L) = 0$
RR	$\mathcal{N}_k \left[\beta_L \omega_k \cos(\omega_k \xi) - \alpha_L \sin(\omega_k \xi)\right]$	$(\alpha_R \beta_L - \beta_R \alpha_L)\,\omega_k \cos(\omega_k L)$ $- \left(\alpha_R \alpha_L + \beta_R \beta_L \omega_k^2\right) \sin(\omega_k L) = 0$

Using the right-end condition further gives $\sin(\omega_k L) = 0$, so the admissible frequencies are

$$\omega_k = \frac{k\pi}{L}, \qquad k = 1, 2, \ldots \tag{85}$$

After normalization, the eigenmodes under DD boundaries are

$$\phi_k(x) = \sqrt{\frac{2}{L}} \sin\left[\frac{k\pi\,(x - x_L)}{L}\right] \tag{86}$$

Table 3 summarizes the eigenmode forms and frequency conditions for common homogeneous Dirichlet, Neumann, and Robin boundary combinations. Let $\xi = x - x_L$, and let $\mathcal{N}_k$ denote the modal normalization constant.

B.2.2 Analytical Solution of the Poisson Equation

From Eq. 72, the k-th Poisson mode satisfies $D\lambda_k a_k^{\mathrm{P}} = q_k^{\mathrm{P}}$, so it can be solved directly as

$$a_k^{\mathrm{P}} = \frac{q_k^{\mathrm{P}}}{D\lambda_k} \tag{87}$$

where q_k^{P} is the projection of the effective Poisson residual forcing onto the k-th eigenmode.

The analytical solution of the Poisson residual is therefore

$$v^{\mathrm{P}}(x) = \sum_{k=1}^{\infty} \frac{q_k^{\mathrm{P}}}{D\lambda_k} \phi_k(x) \tag{88}$$

Combining this with $u^{\mathrm{P}} = \ell^{\mathrm{P}} + v^{\mathrm{P}}$ gives the full analytical solution of the Poisson equation:

$$u^{\mathrm{P}}(x) = \ell^{\mathrm{P}}(x) + \sum_{k=1}^{\infty} \frac{q_k^{\mathrm{P}}}{D\lambda_k} \phi_k(x) \tag{89}$$

B.2.3 Analytical Solution of the Diffusion Equation

From Eq. 73, the k-th diffusion mode satisfies $\frac{da_k^{\mathrm{D}}}{dt} + D\lambda_k a_k^{\mathrm{D}} = q_k^{\mathrm{D}}(t)$. Multiplying both sides by the integrating factor $\mu_k(t) = e^{D\lambda_k t}$ gives

$$e^{D\lambda_k t}\frac{da_k^{\mathrm{D}}}{dt} + D\lambda_k e^{D\lambda_k t}a_k^{\mathrm{D}} = e^{D\lambda_k t}q_k^{\mathrm{D}}(t) \tag{90}$$

Integrating over the time interval $[0, t]$,

$$e^{D\lambda_k t}a_k^{\mathrm{D}}(t) - a_k^{\mathrm{D}}(0) = \int_0^t e^{D\lambda_k \tau}q_k^{\mathrm{D}}(\tau)\,d\tau \tag{91}$$

the k-th diffusion modal coefficient is

$$a_k^{\mathrm{D}}(t) = e^{-D\lambda_k t}a_k^{\mathrm{D}}(0) + \int_0^t e^{-D\lambda_k (t-\tau)}q_k^{\mathrm{D}}(\tau)\,d\tau \tag{92}$$

When q_k^{D} is time-independent,

$$a_k^{\mathrm{D}}(t) = e^{-D\lambda_k t}a_k^{\mathrm{D}}(0) + q_k^{\mathrm{D}}\int_0^t e^{-D\lambda_k (t-\tau)}\,d\tau \tag{93}$$

$$\int_0^t e^{-D\lambda_k (t-\tau)}\,d\tau = \frac{1 - e^{-D\lambda_k t}}{D\lambda_k} \tag{94}$$

and therefore

$$a_k^{\mathrm{D}}(t) = e^{-D\lambda_k t}a_k^{\mathrm{D}}(0) + \frac{q_k^{\mathrm{D}}}{D\lambda_k}\left(1 - e^{-D\lambda_k t}\right) \tag{95}$$

The full analytical solution of the diffusion equation is thus

$$u^{\mathrm{D}}(x,t) = \ell^{\mathrm{D}}(x,t) + \sum_{k=1}^{\infty} a_k^{\mathrm{D}}(t)\phi_k(x) \tag{96}$$

B.2.4 Analytical Solution of the Wave Equation

From Eq. 74, the k-th wave mode satisfies $\frac{d^2 a_k^{\mathrm{W}}}{dt^2} + c_d\frac{da_k^{\mathrm{W}}}{dt} + D\lambda_k a_k^{\mathrm{W}} = q_k^{\mathrm{W}}(t)$. First consider the corresponding homogeneous equation

$$\frac{d^2 a_k^{\mathrm{W}}}{dt^2} + c_d\frac{da_k^{\mathrm{W}}}{dt} + D\lambda_k a_k^{\mathrm{W}} = 0 \tag{97}$$

Its characteristic equation is

$$r^2 + c_d r + D\lambda_k = 0 \tag{98}$$

For the underdamped case $D\lambda_k > \frac{c_d^2}{4}$ considered primarily in this work, define

$$\zeta = \frac{c_d}{2}, \qquad \omega_k^{\mathrm{W}} = \sqrt{D\lambda_k - \zeta^2} \tag{99}$$

The two characteristic roots are then

$$r_{1,2} = -\zeta \pm i\omega_k^{\mathrm{W}} \tag{100}$$

and the corresponding homogeneous response is

$$a_{k,h}^{\mathrm{W}}(t) = e^{-\zeta t}\left[C_{k,1}\cos\left(\omega_k^{\mathrm{W}} t\right) + C_{k,2}\sin\left(\omega_k^{\mathrm{W}} t\right)\right] \tag{101}$$

Let the initial modal displacement and velocity be $a_k^{\mathrm{W}}(0), \dot{a}_k^{\mathrm{W}}(0)$. The initial conditions give the homogeneous response

$$a_{k,h}^{\mathrm{W}}(t) = e^{-\zeta t}\left[a_k^{\mathrm{W}}(0)\cos\left(\omega_k^{\mathrm{W}} t\right) + \frac{\dot{a}_k^{\mathrm{W}}(0) + \zeta a_k^{\mathrm{W}}(0)}{\omega_k^{\mathrm{W}}}\sin\left(\omega_k^{\mathrm{W}} t\right)\right] \tag{102}$$

For a general time-varying forcing $q_k^{\mathrm{W}}(t)$, Duhamel's principle gives the forced response

$$a_{k,p}^{\mathrm{W}}(t) = \frac{1}{\omega_k^{\mathrm{W}}}\int_0^t e^{-\zeta(t-\tau)}\sin\left[\omega_k^{\mathrm{W}}(t-\tau)\right] q_k^{\mathrm{W}}(\tau)\, d\tau \tag{103}$$

Therefore, the k-th modal coefficient is

$$a_k^{\mathrm{W}}(t) = e^{-\zeta t}\left[a_k^{\mathrm{W}}(0)\cos\left(\omega_k^{\mathrm{W}} t\right) + \frac{\dot{a}_k^{\mathrm{W}}(0) + \zeta a_k^{\mathrm{W}}(0)}{\omega_k^{\mathrm{W}}}\sin\left(\omega_k^{\mathrm{W}} t\right)\right] + \frac{1}{\omega_k^{\mathrm{W}}}\int_0^t e^{-\zeta(t-\tau)}\sin\left[\omega_k^{\mathrm{W}}(t-\tau)\right] q_k^{\mathrm{W}}(\tau)\, d\tau \tag{104}$$

The full analytical solution of the wave equation is then

$$u^{\mathrm{W}}(x,t) = \ell^{\mathrm{W}}(x,t) + \sum_{k=1}^{\infty} a_k^{\mathrm{W}}(t)\phi_k(x) \tag{105}$$

B.3 Analytical Time Evolution for Diffusion

The previous section gives the modal-coefficient relations for the Poisson and diffusion equations. For the k-th spatial mode, the Poisson equation satisfies $a_k^{\mathrm{P}} = \frac{q_k^{\mathrm{P}}}{D\lambda_k}$, while the diffusion equation satisfies $\frac{da_k^{\mathrm{D}}}{dt} + D\lambda_k a_k^{\mathrm{D}} = q_k^{\mathrm{D}}(t)$. The two equations share the same spatial eigenmode ϕ_k and eigenvalue λ_k. Their only difference is that the Poisson modal coefficient is determined directly by the current source term, whereas the diffusion modal coefficient must evolve according to the temporal dynamics.

To accommodate time-varying sources, we do not use a single analytical expression from the initial time 0 to an arbitrary time t. Instead, we derive a one-step recurrence between adjacent time nodes t_n and $t_{n+1} = t_n + \Delta t$. Following the derivation in Appendix B.2.3 gives

$$a_{k,n+1}^{\mathrm{D}} = e^{-D\lambda_k \Delta t} a_{k,n}^{\mathrm{D}} + \int_{t_n}^{t_{n+1}} e^{-D\lambda_k(t_{n+1}-\tau)} q_k^{\mathrm{D}}(\tau)\, d\tau \tag{106}$$

where $a_{k,n}^{\mathrm{D}} = a_k^{\mathrm{D}}(t_n)$.

For a general time-varying source, over a single interval $[t_n, t_{n+1}]$ we approximate it by its midpoint value $q_{k,n+\frac{1}{2}}^{\mathrm{D}}$. If the source is given only at discrete nodes, the midpoint value is taken as

$$q_{k,n+\frac{1}{2}}^{\mathrm{D}} \approx \frac{q_{k,n}^{\mathrm{D}} + q_{k,n+1}^{\mathrm{D}}}{2} \tag{107}$$

Substituting this approximation into Eq. 106 and proceeding as in Appendix B.2.3 gives

$$a_{k,n+1}^{\mathrm{D}} = e^{-D\lambda_k \Delta t} a_{k,n}^{\mathrm{D}} + \frac{1 - e^{-D\lambda_k \Delta t}}{D\lambda_k} q_{k,n+\frac{1}{2}}^{\mathrm{D}} \tag{108}$$

Consider the instantaneous Poisson problem associated with the current time interval, and set its effective residual source equal to that of the diffusion equation over the same step, i.e., $q^{\mathrm{P}}_{k,n+\frac{1}{2}} = q^{\mathrm{D}}_{k,n+\frac{1}{2}}$. Using the Poisson modal relation $a^{\mathrm{P}}_k = \frac{q^{\mathrm{P}}_k}{D\lambda_k}$, we obtain

$$a^{\mathrm{P}}_{k,n+\frac{1}{2}} = \frac{q^{\mathrm{P}}_{k,n+\frac{1}{2}}}{D\lambda_k} = \frac{q^{\mathrm{D}}_{k,n+\frac{1}{2}}}{D\lambda_k} \tag{109}$$

Substituting Eq. 109 into Eq. 108 gives the one-step Poisson-based analytical update for diffusion:

$$a^{\mathrm{D}}_{k,n+1} = e^{-D\lambda_k \Delta t} a^{\mathrm{D}}_{k,n} + \left(1 - e^{-D\lambda_k \Delta t}\right) a^{\mathrm{P}}_{k,n+\frac{1}{2}} \tag{110}$$

For a time-varying source with a fixed homogeneous boundary operator, the analytical update requires only updating $q^{\mathrm{D}}_{k,n+\frac{1}{2}}$ at each time step to obtain the corresponding $a^{\mathrm{P}}_{k,n+\frac{1}{2}}$, while the spatial eigenmodes ϕ_k and eigenvalues λ_k remain unchanged.

B.4 Analytical Time Evolution for Wave Dynamics

For the wave problem, the previous section gives the k-th modal equation $\frac{d^2 a^{\mathrm{W}}_k}{dt^2} + c_d \frac{da^{\mathrm{W}}_k}{dt} + D\lambda_k a^{\mathrm{W}}_k = q^{\mathrm{W}}_k(t)$. As with diffusion, to accommodate time-varying external forcing we approximate the source over each time interval $[t_n, t_{n+1}]$ by its midpoint value.

$$q^{\mathrm{W}}_{k,n+\frac{1}{2}} \approx \frac{q^{\mathrm{W}}_{k,n} + q^{\mathrm{W}}_{k,n+1}}{2} \tag{111}$$

The wave modal equation over the current time interval is therefore approximated as

$$\frac{d^2 a^{\mathrm{W}}_k}{dt^2} + c_d \frac{da^{\mathrm{W}}_k}{dt} + D\lambda_k a^{\mathrm{W}}_k = q^{\mathrm{W}}_{k,n+\frac{1}{2}} \tag{112}$$

Consider the instantaneous Poisson problem associated with the current wave source and set $q^{\mathrm{P}}_{k,n+\frac{1}{2}} = q^{\mathrm{W}}_{k,n+\frac{1}{2}}$. The Poisson modal relation gives

$$q^{\mathrm{W}}_{k,n+\frac{1}{2}} = D\lambda_k a^{\mathrm{P}}_{k,n+\frac{1}{2}} \tag{113}$$

Eq. 112 can then be written as

$$\frac{d^2 a^{\mathrm{W}}_k}{dt^2} + c_d \frac{da^{\mathrm{W}}_k}{dt} + D\lambda_k a^{\mathrm{W}}_k = D\lambda_k a^{\mathrm{P}}_{k,n+\frac{1}{2}} \tag{114}$$

Unlike diffusion, the wave equation is second order in time, so its state contains not only the modal coefficient but also the modal velocity. Define the modal velocity $v^{\mathrm{W}}_k = \frac{da^{\mathrm{W}}_k}{dt}$, and let $a^{\mathrm{W}}_{k,n} = a^{\mathrm{W}}_k(t_n)$ and $v^{\mathrm{W}}_{k,n} = v^{\mathrm{W}}_k(t_n)$. For the underdamped case, use the notation $\zeta = \frac{c_d}{2}$, $\omega^{\mathrm{W}}_k = \sqrt{D\lambda_k - \zeta^2}$ from the previous section. Because $a^{\mathrm{P}}_{k,n+\frac{1}{2}}$ is treated as constant over the current time interval, define the deviation from the current instantaneous Poisson equilibrium state as

$$\widetilde{a}^{\mathrm{W}}_k(t) = a^{\mathrm{W}}_k(t) - a^{\mathrm{P}}_{k,n+\frac{1}{2}} \tag{115}$$

Eq. 112 then becomes

$$\frac{d^2 \widetilde{a}^{\mathrm{W}}_k}{dt^2} + c_d \frac{d\widetilde{a}^{\mathrm{W}}_k}{dt} + D\lambda_k \widetilde{a}^{\mathrm{W}}_k = 0 \tag{116}$$

Thus, wave propagation over one time step is equivalent to an analytical damped-oscillation update around the current instantaneous Poisson modal state. Let $C_k = cos\left(\omega_k^{\mathrm{W}}\Delta t\right)$, $S_k = sin\left(\omega_k^{\mathrm{W}}\Delta t\right)$. Using the homogeneous wave solution from the previous section, the one-step modal-coefficient update is

$$a_{k,n+1}^{\mathrm{W}} = a_{k,n+\frac{1}{2}}^{\mathrm{P}} + e^{-\zeta\Delta t}\left[\left(C_k + \frac{\zeta}{\omega_k^{\mathrm{W}}}S_k\right)\left(a_{k,n}^{\mathrm{W}} - a_{k,n+\frac{1}{2}}^{\mathrm{P}}\right) + \frac{S_k}{\omega_k^{\mathrm{W}}}v_{k,n}^{\mathrm{W}}\right] \tag{117}$$

The corresponding modal-velocity update is

$$v_{k,n+1}^{\mathrm{W}} = e^{-\zeta\Delta t}[-\frac{D\lambda_k}{\omega_k^{\mathrm{W}}}S_k\left(a_{k,n}^{\mathrm{W}} - a_{k,n+\frac{1}{2}}^{\mathrm{P}}\right) + \left(C_k - \frac{\zeta}{\omega_k^{\mathrm{W}}}S_k\right)v_{k,n}^{\mathrm{W}}] \tag{118}$$

Eqs. 117 and 118 are the analytical time-update formulas for the wave equation. As in diffusion, when the external excitation varies with time, only the modal coefficient $a_{k,n+1}^{\mathrm{W}}$ is updated step by step, while the spatial eigenmode ϕ_k and eigenvalue λ_k remain fixed.

B.5 Extension to Two Spatial Dimensions

B.5.1 Tensor-Product Eigenmodes in 2D

Consider a 2D rectangular domain $\Omega = [x_L, x_R] \times [y_B, y_T]$. The corresponding eigenproblem is

$$-\Delta\Phi_{ij}(x,y) = \lambda_{ij}\Phi_{ij}(x,y), \qquad \mathcal{B}_0\Phi_{ij} = 0 \tag{119}$$

For a rectangular domain, write the 2D mode as the product of 1D modes along the two coordinate directions:

$$\Phi_{ij}(x,y) = \phi_i^x(x)\phi_j^y(y) \tag{120}$$

where the 1D eigenmodes in the two directions satisfy

$$-\frac{d^2\phi_i^x}{dx^2} = \lambda_i^x\phi_i^x \tag{121}$$

$$-\frac{d^2\phi_j^y}{dy^2} = \lambda_j^y\phi_j^y \tag{122}$$

Therefore, the 1D eigenmode forms derived in Appendix B.2 can be applied separately in the x and y directions. Substituting Eq. 120 into the 2D Laplacian equation gives

$$-\Delta\Phi_{ij} = -\frac{d^2\phi_i^x}{dx^2}\phi_j^y - \phi_i^x\frac{d^2\phi_j^y}{dy^2} = \left(\lambda_i^x + \lambda_j^y\right)\Phi_{ij} \tag{123}$$

Hence, the eigenvalue of the 2D tensor-product mode is

$$\lambda_{ij} = \lambda_i^x + \lambda_j^y \tag{124}$$

B.5.2 Analytical Structure and Temporal Dynamics in 2D

In the 2D eigenspace, the residual solution can be expanded as

$$v(x,y,t) = \sum_i\sum_j a_{ij}(t)\Phi_{ij}(x,y) \tag{125}$$

The projection of the effective residual source onto the (i, j)-th mode is correspondingly

$$q_{ij}(t) = \langle q(\cdot, \cdot, t), \Phi_{ij} \rangle = \int_{y_B}^{y_T} \int_{x_L}^{x_R} q(x, y, t) \phi_i^x(x) \phi_j^y(y) \, dx \, dy \tag{126}$$

Using the 2D eigenrelation $-\Delta \Phi_{ij} = \lambda_{ij} \Phi_{ij}$, the modal relations for the three equation classes in Appendix B.1 extend directly to 2D:

$$D \lambda_{ij} a_{ij}^{\mathrm{P}} = q_{ij}^{\mathrm{P}} \tag{127}$$

$$\frac{da_{ij}^{\mathrm{D}}}{dt} + D \lambda_{ij} a_{ij}^{\mathrm{D}} = q_{ij}^{\mathrm{D}}(t) \tag{128}$$

$$\frac{d^2 a_{ij}^{\mathrm{W}}}{dt^2} + c_d \frac{da_{ij}^{\mathrm{W}}}{dt} + D \lambda_{ij} a_{ij}^{\mathrm{W}} = q_{ij}^{\mathrm{W}}(t) \tag{129}$$

The analytical solutions are obtained by substituting the corresponding modal coefficients into Eq. 125. The diffusion and wave time updates then directly use the analytical recurrences derived in Appendices B.3 and B.4 and are not repeated here.

C Implementation Details of ANNO

This appendix provides implementation details for the ANNO networks.

C.1 Learnable Mode Network

C.1.1 Boundary Condition Encoding

After the Analytical Lifting Module, the nonhomogeneous boundary values γ have been separated from the original problem, so the Learnable Mode Network uses only the (α, β) information in the homogeneous boundary operator. For a 1D problem, the two homogeneous endpoint boundaries are described by $(\alpha_L, \beta_L), (\alpha_R, \beta_R)$. We first normalize the (α, β) parameters on each boundary, then flatten the two boundary parameter sets and pass them to a Boundary Encoder to obtain a unified boundary feature. The Boundary Encoder is a three-layer fully connected network with SiLU activations.

$$\mathbf{h}_B = E_B \left(\alpha_L, \beta_L, \alpha_R, \beta_R \right). \tag{130}$$

C.1.2 Trigonometric Mode Reconstruction

The boundary feature $\mathbf{h}_B$ is then passed to the frequency head and amplitude head:

$$\widetilde{\omega} = H_\omega \left(\mathbf{h}_B \right) \tag{131}$$

$$\{A_k, B_k\}_{k=1}^{K} = H_A \left(\mathbf{h}_B \right) \tag{132}$$

Here, K is the number of learned modes, $\widetilde{\omega}$ contains their frequency parameters subject to the frequency-band strategy in Appendix C.1.3, and A_k, B_k contains their amplitudes. Both output heads are fully connected networks.

As derived analytically in Appendix B.2, the eigenfunctions of the 1D constant-coefficient eigenproblems considered here belong to a common trigonometric family. The Learnable Mode Network therefore reconstructs frequencies and amplitudes with trigonometric modes to obtain learnable eigenmodes with structural priors. Define the normalized coordinate $\xi = \frac{x - x_L}{x_R - x_L}$. The k-th learned mode is

$$\phi_k(x) = A_k \cos(\omega_k \xi) + B_k \sin(\omega_k \xi) \tag{133}$$

C.1.3 FREQUENCY-BAND STRATEGY

If all modal frequencies are learned freely over the full frequency interval, different learned modes can converge to similar frequencies, causing eigenfunction collapse. ANNO therefore introduces a frequency-band strategy that assigns ordered frequency bands to successive modes in advance. Let the frequency range of the 0-th mode be $\omega_0 \in [0, 1.25\pi]$. For the $k \geq 1$-th mode, we use

$$\omega_k \in [(k + 0.25)\pi, (k + 1.25)\pi] \tag{134}$$

The network first predicts an unconstrained frequency parameter $\widetilde{\omega}_k$, maps it to the corresponding band with a sigmoid, and then substitutes it into Eq. 133 for mode reconstruction.

$$\omega_k = \omega_k^{low} + \left(\omega_k^{high} - \omega_k^{low}\right) \sigma(\widetilde{\omega}_k) \tag{135}$$

After each mode is obtained, it is further normalized to remove the ambiguity in overall amplitude scale, yielding the final learned modes $\phi_k(x)$.

C.2 LEARNABLE COEFFICIENT NETWORK

The Learnable Coefficient Network predicts the steady-state modal coefficients associated with the current learned modes. For an effective residual forcing q, its projection onto the k-th learned mode is

$$q_k = \langle q, \phi_k \rangle, \tag{136}$$

which represents the corresponding modal component of the forcing. Stacking all projections gives $\mathbf{q} = [q_1, \ldots, q_K]^\mathsf{T}$. For 2D problems, the corresponding projection is $q_{ij} = \langle q, \Phi_{ij} \rangle$.

C.2.1 INPUT ENCODING AND NETWORK ARCHITECTURE

The coefficient network uses two inputs: the normalized source projections and the diffusion coefficient D broadcast across the modal channels. In the implementation, D is represented in log scale before concatenation with the normalized projections. Denoting the resulting input by $\mathbf{z}_c$, the modal coefficients are predicted as

$$\mathbf{a} = F_c(\mathbf{z}_c), \qquad \mathbf{a} = [a_1, \ldots, a_K]^\mathsf{T}. \tag{137}$$

The Learnable Coefficient Network is a four-layer fully connected network with SiLU activations in the hidden layers. The same construction is used for all steady-state coefficient inference in ANNO.

C.3 STRUCTURE-CONSTRAINED LEARNING

Training ANNO only with steady-state solution-field error can provide strong function-approximation accuracy, but it does not ensure that the spatial modes produced by the Learnable Mode Network satisfy the eigenrelation in Section 2.2. ANNO therefore introduces an eigenstructure constraint so that the learned modes form a reusable analytical structure. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_{\text{eig}} \mathcal{L}_{\text{eig}} \tag{138}$$

C.3.1 DATA LOSS

ANNO's data loss is the relative L_2 error computed on the residual field. Let the predicted and reference residuals be v_{pred} and v_{ref}, respectively. Then $\mathcal{L}_{\text{data}}$ is

$$\mathcal{L}_{\text{data}} = \frac{\| v_{pred} - v_{ref} \|_2}{\| v_{ref} \|_2 + \varepsilon} \tag{139}$$

C.3.2 EIGENSTRUCTURE CONSTRAINT

For the k-th learned mode, the theoretical eigenrelation is $L\phi_k = \lambda_k \phi_k$. Since the eigenvalue is not prescribed, we estimate it from the current learned mode using the Rayleigh quotient

$$\hat{\lambda}_k = \frac{\langle \phi_k, L_h \phi_k \rangle}{\langle \phi_k, \phi_k \rangle}, \tag{140}$$

where L_h is the discrete spatial operator. We then define the normalized eigenstructure error as

$$e_k = \frac{\| L_h \phi_k - \hat{\lambda}_k \phi_k \|_2^2}{\| L_h \phi_k \|_2^2 + \| \hat{\lambda}_k \phi_k \|_2^2 + \varepsilon} \tag{141}$$

If all modes are assigned equal weight, the uniform eigenstructure loss is

$$\mathcal{L}_{\text{uniform}} = \frac{1}{K} \sum_{k=1}^{K} e_k \tag{142}$$

C.3.3 COEFFICIENT-BASED MODE IMPORTANCE WEIGHTING

Different learned modes generally contribute unequally to the current solution reconstruction. To focus the eigenstructure constraint on the modes that actually dominate the current solution field, ANNO constructs importance weights from the modal coefficients. For a 1D problem, the importance of the k-th mode is

$$s_k = |a_k|^2 \tag{143}$$

After normalizing s_k to obtain the modal weights w_k, the weighted eigenstructure loss is defined as

$$\mathcal{L}_{\text{weighted}} = \sum_{k=1}^{K} w_k e_k. \tag{144}$$

The final eigenstructure loss combines the uniform and weighted terms. In this work, $\eta = 0.8$:

$$\mathcal{L}_{\text{eig}} = (1 - \eta)\mathcal{L}_{\text{uniform}} + \eta \mathcal{L}_{\text{weighted}}. \tag{145}$$

C.3.4 STAGED TRAINING STRATEGY

Early in training, the learned modes, modal coefficients, and reconstructed residual have not yet formed stable structures, and imposing a strong eigenstructure constraint at this stage makes optimization more difficult. ANNO therefore uses a staged training strategy. During the first 15% of training, we optimize only $\mathcal{L}_{\text{data}}$, i.e., $\lambda_{\text{eig}} = 0$ and $\eta = 0$. We then enter an eigenbasis warm-up stage: as training progress increases from 15% to 30%, λ_{eig} is increased linearly from 0 to 0.1 while using the uniform eigenstructure constraint. From 30% to 60%, $\lambda_{\text{eig}} = 0.1$ is fixed and the importance-mixing coefficient η is gradually increased from 0 to 0.8. Finally, $\lambda_{\text{eig}} = 0.1, \eta = 0.8$ is fixed during the remaining training for joint fine-tuning.

The specific network sizes, optimizer, learning-rate schedule, batch size, training epochs, and early-stopping configuration are given in the appendix for each experiment.

D Supplementary Details for Experiment 1

This appendix provides additional details on data generation, model configuration, training settings, evaluation metrics, and statistical results for the 1D Poisson experiment in Section 5.1.

D.1 Dataset Construction and Experimental Configuration

Experiment 1 considers the 1D steady-state Poisson problem

$$-D\frac{\partial^2 u^{\mathrm{P}}(x)}{\partial x^2} = f^{\mathrm{P}}(x), \qquad x \in [0,1], \tag{146}$$

with boundary conditions

$$\alpha u^{\mathrm{P}}(x) + \beta\frac{\partial u^{\mathrm{P}}(x)}{\partial x} = \gamma, \qquad x \in \partial\Omega. \tag{147}$$

The problem parameters, sampling ranges, and dataset sizes are summarized in Table 4.

Table 4: Experimental configuration of Experiment 1.

Item	**Setting**
Spatial domain	$\Omega = [0,1]$
Spatial resolution	$N_x = 128$
Diffusion coefficient	$D \in [0.2, 5.0]$, log-uniform
Boundary combinations	DD, DN, ND, DR, RD, NR, RN, RR
Dirichlet	$\alpha = 1,\ \beta = 0$
Neumann	$\alpha = 0,\ \beta_L = -1,\ \beta_R = 1$
Robin	$\alpha \in [0.2, 5.0],\ \beta_L = -1,\ \beta_R = 1$
Nonhomogeneous boundary data	$\gamma \in [-1, 1]$, uniform
Robin sampling	3 logarithmic strata
Source magnitude	$\lvert A_f\rvert \in [0.5, 2.0]$, random sign
Training / validation / test	$12,288/2,048/2,048$

To cover different spatial scales, locality, and sign variations, we construct eight source families $f^{\mathrm{P}}(x)$. In the following, A_f denotes the signed source amplitude, while the remaining symbols are local random parameters for each source family. The specific forms are listed in Table 5.

Table 5: Source families used in Experiment 1.

Source family	**Definition and random parameters**
Uniform	$f^{\mathrm{P}}(x) = A_f$
Linear gradient	$f^{\mathrm{P}}(x) = A_f[1 + c(2x-1)], \quad c \in [-1, 1]$
Single Gaussian	$f^{\mathrm{P}}(x) = A_f \exp[-(x-\mu)^2/(2\sigma^2)], \quad \mu \in [0.15, 0.85],\ \sigma \in [0.04, 0.18]$
Double Gaussian	$f^{\mathrm{P}}(x) = A_f[g^1(x) + cg^2(x)], \quad c \in [-1, 1]$; centers $\in [0.1, 0.9]$, widths $\in [0.04, 0.16]$
Periodic	$f^{\mathrm{P}}(x) = A_f \sin(m\pi x + \theta), \quad m \in \{1, 2, 3, 4\},\ \theta \in [0, 2\pi]$
Exponential decay	$f^{\mathrm{P}}(x) = A_f \exp(-\xi/\tau), \quad \tau \in [0.08, 0.35]$; decay direction selected randomly
Piecewise constant	interface $x_c \in [0.2, 0.8]$; amplitude ratio $c \in [-1, 1]$
Source-sink dipole	difference of two Gaussian terms; centers $\in [0.15, 0.85]$, widths $\in [0.04, 0.14]$

The eight source families are assigned cyclically across samples. The training set uses three logarithmic strata for the Robin parameter, resulting in different numbers of samples across boundary combinations, as detailed in Table 6.

Table 6: Training sample allocation.

Boundary	**Samples**
DD	512
DN	512
ND	512
DR	1,536
RD	1,536
NR	1,536
RN	1,536
RR	4,608
Total	12,288

D.2 Model Configuration and Training Settings

Experiment 1 compares ANNO, FNO, and DeepONet. All three models use the same training, validation, and test datasets and receive the same physical information f^{P}, D, and full boundary parameters $(\alpha_L, \beta_L, \gamma_L, \alpha_R, \beta_R, \gamma_R)$.

FNO learns the mapping from the input to the solution field through frequencies in Fourier space. Its input is

$$[f^{\mathrm{P}}(x), x, D, \alpha_L, \beta_L, \gamma_L, \alpha_R, \beta_R, \gamma_R];$$

multiple Fourier operator layers extract global spatial features and directly predict the full Poisson solution $\hat{u}^{\mathrm{P}}(x)$. We use 14 Fourier modes and 4 Fourier layers.

DeepONet consists of a Branch Net and a Trunk Net. The Branch Net encodes the source term, diffusion coefficient, and boundary conditions, with input

$$[f^{\mathrm{P}}(x^1), \ldots, f^{\mathrm{P}}(x^{128}), D, \alpha_L, \beta_L, \gamma_L, \alpha_R, \beta_R, \gamma_R].$$

The Trunk Net encodes the spatial location x. We set the latent dimension to 80, and both the Branch Net and Trunk Net use three fully connected layers.

The three models have comparable parameter counts, as shown in Table 7. Training hyperparameters are listed in Table 8. FNO and DeepONet optimize only the full-solution relative L_2 loss, while ANNO uses the same basic optimizer settings together with the structure-constrained curriculum in Appendix C.3.

Table 7: Model configurations.

Model	**Main configuration**	**Parameters**
ANNO	$K = 12$; hidden dim. 96	57,358
FNO	Fourier modes 14; width 22; blocks 4	59,701
DeepONet	latent 80; Branch 96-96-96; Trunk 80-80-80	59,041

Table 8: Training settings.

Setting	**Value**
Optimizer	AdamW
Maximum epochs	2000
Batch size	64

Setting	Value
Initial learning rate	3×10^{-4}
Weight decay	10^{-5}
Scheduler	Cosine annealing
Gradient clipping	1.0
Early-stopping patience	50
Random seed	2026

D.3 Evaluation Metrics and Quantitative Results

Steady-state prediction error is measured on the full solution u^{P} using the relative L_2 error

$$e_{\mathrm{rel}} = \frac{\left\|\hat{u}^{\mathrm{P}} - u^{\mathrm{P}}\right\|_2}{\left\|u^{\mathrm{P}}\right\|_2}. \tag{148}$$

In Fig. 2(b), the bar height for each boundary combination is the median over 256 test samples, and the error bars show the 25th-75th percentile interval. Overall results are reported in Table 9.

Table 9: Overall steady-state solution prediction statistics.

Model	Mean	25th percentile	Median	75th percentile
ANNO	0.1643%	0.0343%	0.0749%	0.1639%
FNO	2.6326%	0.9753%	1.5594%	2.8098%
DeepONet	4.0295%	1.3593%	2.4238%	4.4192%

The median relative L_2 errors for the eight boundary combinations are summarized in Table 10.

Table 10: Median relative L_2 error for different boundary combinations.

Model	DD	DN	ND	DR	RD	NR	RN	RR
ANNO	0.0476%	0.0629%	0.0658%	0.0713%	0.0604%	0.0893%	0.0975%	0.1210%
FNO	1.9634%	1.4585%	1.4641%	1.6794%	1.3384%	1.7476%	1.6276%	1.4311%
DeepONet	2.6477%	2.2141%	2.1081%	2.2211%	2.3005%	2.6313%	2.5729%	2.6571%

In Fig. 2(c), the learned mode is denoted by $\hat{\phi}_k$ and the corresponding numerical reference eigenmode by ϕ_k. Both are normalized using L_2, and their modal overlap is defined as

$$\rho_k = \left| \left\langle \hat{\phi}_k, \phi_k \right\rangle_h \right|. \tag{149}$$

Values closer to 1 indicate greater similarity. Statistics for the first four modes over the full test set are reported in Table 11.

Table 11: Learned modal overlap over the complete test set.

Mode	25th percentile	Median	75th percentile
$k = 1$	0.9999989	0.9999997	0.9999999
$k = 2$	0.9993443	0.9999719	0.9999999
$k = 3$	0.9943639	0.9998169	0.9999909
$k = 4$	0.9893528	0.9996496	0.9999292

We also measure the relative error

$$\frac{\left|\hat{a}_k^{\mathrm{P}} - a_k^{\mathrm{P}}\right|}{\left|a_k^{\mathrm{P}}\right| + \varepsilon}$$

between the learned modal coefficients $\hat{a}_k^{\mathrm{P}}$ and the numerically computed coefficients a_k^{P}, with results summarized in Table 12. The coefficient error increases for higher modes, which is expected for two reasons. First, the denominator in a relative error becomes smaller as the absolute coefficient magnitude decreases, which inflates the relative error; higher modes contribute less to the solution and therefore generally have smaller coefficients. Second, the eigenstructure constraint is weighted according to the modal coefficients. Because low-frequency modes account for most of the reconstructed solution in the majority of problems, ANNO is intentionally encouraged to learn these low-frequency modes first.

Table 12: Modal coefficient prediction accuracy.

Mode	Median relative error
$k = 1$	0.0901%
$k = 2$	1.3778%
$k = 3$	5.6599%
$k = 4$	12.3863%

The normalized modal energy in Fig. 2(c) is computed as

$$E_k^{\mathrm{P}} = \frac{|a_k^{\mathrm{P}}|^2}{\sum_j |a_j^{\mathrm{P}}|^2}, \qquad \hat{E}_k^{\mathrm{P}} = \frac{|\hat{a}_k^{\mathrm{P}}|^2}{\sum_j |\hat{a}_j^{\mathrm{P}}|^2}. \tag{150}$$

E Supplementary Details for Experiment 2

This appendix provides additional details on data generation, baseline configurations, training settings, evaluation metrics, and quantitative results for the 1D Poisson-to-diffusion experiment in Section 4.2.

E.1 Dataset Construction and Experimental Configuration

Experiment 2 considers the 1D transient diffusion problem

$$\frac{\partial u^{\mathrm{D}}}{\partial t} - D\frac{\partial^2 u^{\mathrm{D}}}{\partial x^2} = f^{\mathrm{D}}(x,t), \qquad x \in [0,1], \tag{151}$$

with boundary conditions

$$\alpha u^{\mathrm{D}} + \beta u_x^{\mathrm{D}} = \gamma(t). \tag{152}$$

The spatial domain, diffusion-coefficient range, eight boundary combinations, and base source families are the same as in Experiment 1. Experiment 2 further introduces nonzero initial conditions, time-varying sources, and time-varying boundaries, as summarized in Table 13.

To normalize the time scale across different diffusion coefficients and boundary conditions, each trajectory defines a characteristic time from the lowest eigenvalue under its homogeneous boundary conditions:

$$\tau = (D\lambda_1)^{-1}. \tag{153}$$

We set $T = 3\tau$ and $\Delta t/\tau = 3/256$. Thus, each trajectory spans three characteristic times with 256 time intervals and 257 stored frames.

Combining the six dynamic cases with the eight boundary combinations gives 48 settings. For each setting, we generate 32 training, 8 validation, and 64 test trajectories, giving 1,536, 384, and 3,072 trajectories, respectively.

Table 13: Dynamic cases used in Experiment 2.

Dynamic case	Setting
Static-0	The source and boundary remain fixed, and the initial residual is zero.
Static-NZ	The source and boundary remain fixed, but the initial residual is nonzero.
Modulated	The spatial source profile remains fixed while its amplitude varies with time.
Moving	The source profile moves spatially over time.
Time-BC	The source remains fixed, while only $\gamma(t)$ varies with time.
Combined	A nonzero initial condition, time-varying source, and time-varying boundary are present simultaneously.

Table 14: Experimental configuration of Experiment 2.

Item	Setting
Spatial domain	$x \in [0, 1]$
Spatial resolution	$N_x = 128$
Diffusion coefficient	$D \in [0.2, 5.0]$, log-uniform
Boundary combinations	DD, DN, ND, DR, RD, NR, RN, RR
Nonhomogeneous boundary data	$\gamma \in [-1, 1]$, uniform
Robin coefficient	$\alpha \in [0.2, 5.0]$
Characteristic time	$\tau = (D\lambda_1)^{-1}$
Final time	$T = 3\tau$
Temporal intervals / stored frames	256 / 257
Temporal cycles	$[0.5, 2.0]$
Temporal modulation amplitude	$[0.25, 0.75]$
Training / validation / test	1,536 / 384 / 3,072 trajectories

E.2 Baseline Configuration and Training Settings

Experiment 2 uses two transient baselines representing two common learning strategies for time-dependent PDEs: a fully supervised space-time Fourier Neural Operator (ST-FNO) and a POD-based recursive propagation network, POD-DeepPropNet.

ST-FNO follows the Fourier Neural Operator framework (Li et al., 2021), which parameterizes global operator kernels in Fourier space. In our implementation, both space and time are treated as Fourier dimensions, so the model directly learns a spatiotemporal mapping from the complete external conditions to the full transient solution. At each spatiotemporal location, the input is

$$\mathbf{z}_{ij} = \left[f^{\mathrm{D}}(x_i, t_j), u^{\mathrm{D}}(x_i, 0), D, \alpha_L, \beta_L, \gamma_L(t_j), \alpha_R, \beta_R, \gamma_R(t_j), x_i, \frac{t_j}{\tau} \right]. \tag{154}$$

The 11-dimensional input is mapped to a width-4 feature space, followed by four 2D Fourier layers operating jointly over space and time. The network retains 14 spatial modes and 16 temporal modes and predicts the complete trajectory in one pass. It is therefore a fully supervised spatiotemporal neural-operator baseline and does not use autoregressive rollout.

POD-DeepPropNet is based on DeepPropNet (Liu & Cai, 2022), which learns a short-time evolution propagator and recursively applies it to model long-time PDE dynamics. The original DeepPropNet uses a POD-based reduced representation together with a learned propagator. In our implementation, the transient field is represented using 12 POD modes, and a neural propagation network recursively advances the corresponding modal coefficients. The propagation network contains six hidden layers of width 101 and is trained with an eight-step rollout. This baseline therefore represents a data-driven reduced-order temporal-propagation approach.

The model configurations and parameter counts are summarized in Table 15. All three models have comparable parameter counts.

ST-FNO is trained with AdamW for at most 2,000 epochs using batch size 4, an initial learning rate of 10^{-4}, weight decay 10^{-5}, and early stopping. POD-DeepPropNet is also trained for at most 2,000 epochs with an initial learning rate of 10^{-4}, using batch size 256 and an eight-step rollout during training. Both transient baselines use the 1,536 training trajectories and 384 validation trajectories.

Table 15: Model configurations and parameter counts in Experiment 2.

Model	Main configuration	Parameters
ANNO	12 learned modes; frozen Exp. 1 model	57,358
ST-FNO	modes $(14, 16)$ in (x, t); width 4; 4 Fourier layers; padding $(8, 8)$	58,273
POD-DeepPropNet	12 POD modes; 6 hidden layers; width 101	58,968

Table 16: Input information shown for the representative cases in Fig. 3(a).

Dynamic case	Input information shown
Static-0	Not shown
Static-NZ	Initial field and the nonzero initial residual after homogenization
Modulated	Time-varying source amplitude
Moving	Source $f^{\mathrm{D}}(x,t)$ moving over time
Time-BC	Left and right boundary values $\gamma_L(t)$ and $\gamma_R(t)$
Combined	Initial field and residual, time-varying source, and time-varying boundary

In contrast, ANNO directly reuses the model trained in Experiment 1. All network parameters remain frozen, and no transient trajectories are used for training. At each time interval, the frozen steady-state ANNO provides the current steady modal response, while the transient modal coefficients are advanced according to Eq. 10. For time-varying boundaries, the lifting function is recomputed from the current boundary data.

E.3 Evaluation Metrics and Quantitative Results

Figure 3(a) shows representative test trajectories for different dynamic cases. For each case, a sample with trajectory relative L_2 error close to the case median is selected. The input variations shown for each case are summarized in Table 16. Pointwise absolute error is defined as

$$\mathrm{Error}(x,t) = \left|\hat{u}^{\mathrm{D}}(x,t) - u^{\mathrm{D}}(x,t)\right|. \tag{155}$$

For the n-th trajectory, the full-trajectory relative L_2 error is defined as

$$e_{\mathrm{traj}}^{(n)} = \frac{\left\|\hat{u}_n^{\mathrm{D}} - u_n^{\mathrm{D}}\right\|_2}{\left\|u_n^{\mathrm{D}}\right\|_2}. \tag{156}$$

The mean errors under each dynamic case are listed in Table 17. Each dynamic case contains 512 test trajectories, covering all eight boundary-condition combinations.

ANNO maintains low trajectory errors across all six dynamic cases, including nonzero initial conditions, moving or modulated sources, time-varying boundaries, and their combined variation. Its overall mean relative L_2 error is 0.2531%, compared with 22.9472% for ST-FNO and 45.5992% for POD-DeepPropNet.

Figure 3(c) further reports the prediction error over time. For the n-th trajectory, the instantaneous relative L_2 error at the j-th saved time is

$$e_n(t_j) = \frac{\left\|\hat{u}_n^{\mathrm{D}}(\cdot,t_j) - u_n^{\mathrm{D}}(\cdot,t_j)\right\|_2}{\left\|u_n^{\mathrm{D}}(\cdot,t_j)\right\|_2}. \tag{157}$$

At each saved time, the error distribution is computed over all test trajectories. The median is used as the center curve, and the shaded region denotes the 25th–75th percentile interval.

E.4 Additional Analysis of Transient Baselines

We note that the transient baselines achieve lower accuracy on the present tasks than reported in their original studies. To better understand this behavior, we first examine whether it can be explained by

Table 17: Mean full-trajectory relative L_2 error in Experiment 2.

Dynamic case	ANNO	ST-FNO	POD-DeepPropNet
Static-0	0.2030%	18.1586%	48.7573%
Static-NZ	0.2046%	16.1155%	32.7774%
Modulated	0.2320%	20.6232%	45.7023%
Moving	0.2187%	24.1393%	39.5996%
Time-BC	0.3183%	28.6999%	59.2658%
Combined	0.3423%	29.9468%	47.4929%
Overall	0.2531%	22.9472%	45.5992%

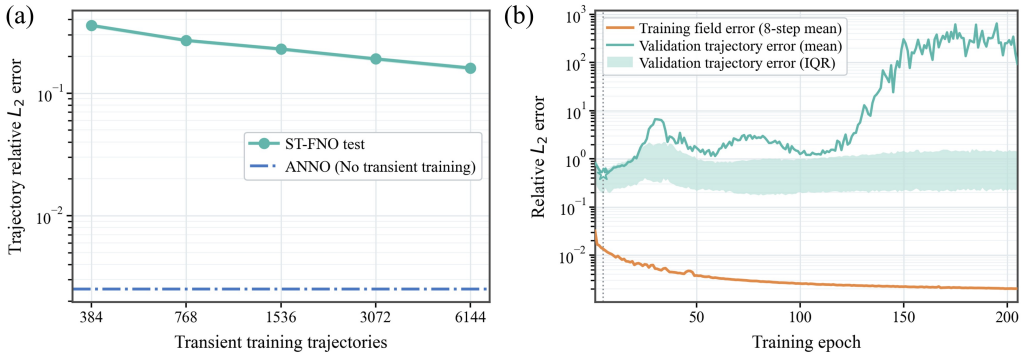

Figure 10: **Additional analysis of transient baselines in Experiment 2.** (a) ST-FNO test error with 384–6144 transient training trajectories, with ANNO shown as a reference without transient training. (b) POD-DeepPropNet local training error and full-trajectory validation error over training epochs.

limited training data or unsuccessful optimization. We vary the number of ST-FNO training trajectories from 384 to 6144 while keeping the network architecture, training protocol, validation set, and test set fixed. As shown in Fig. 10(a), the ST-FNO test error decreases consistently as more transient trajectories are provided, but remains substantially higher than that of ANNO even with 6144 training trajectories. We also inspect the training dynamics of POD-DeepPropNet. Figure 10(b) shows that its local eight-step training error continues to decrease, whereas the full-trajectory validation error does not improve accordingly and eventually increases substantially.

These observations led us to suspect that the difficulty of the transient family itself may be the main reason for the poor baseline performance. We therefore construct a simplified diffusion family containing only DD boundaries, a uniform source, and a static zero initial condition, while keeping the same spatial and temporal resolutions. ST-FNO and POD-DeepPropNet are trained on 1,536 transient trajectories, and all methods are evaluated on the same 3,072 test trajectories. As shown in Fig. 11, both transient baselines recover low trajectory errors on this simplified family, with performance comparable to ANNO. This provides additional evidence that the large baseline errors in the full Experiment 2 are associated with the diversity and difficulty of the original transient family, rather than simply insufficient data or failed optimization.

Since a systematic study of why different transient neural operators degrade under increasingly complex PDE families is beyond the scope of this work, we leave a more detailed investigation to future work.

F Supplementary Details for Experiment 3

This appendix provides additional details on the physical problem, dataset construction, model configuration, and evaluation for the 2D elastically supported membrane problem in Section 4.3.

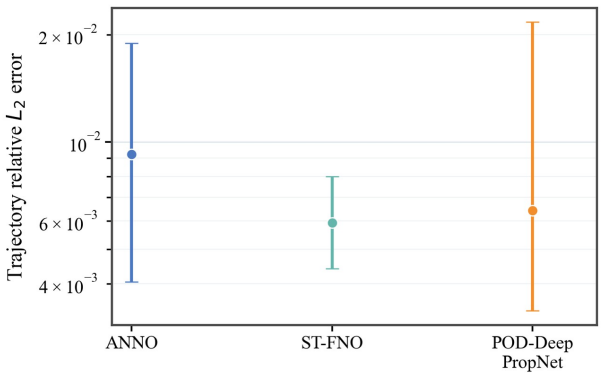

Figure 11: **Baseline performance on a simplified diffusion family.** Full-trajectory relative L_2 errors of ANNO, ST-FNO, and POD-DeepPropNet on the DD/uniform/static-zero family. Markers denote the median and error bars show the interquartile range.

F.1 Problem Setup

Experiment 3 considers a 2D tensioned membrane with elastic supports. The steady-state problem is

$$-T\Delta w(x,y) = p(x,y), \tag{158}$$

where T is the membrane tension and $p(x,y)$ is the transverse load. All four edges use elastic Robin supports,

$$T\frac{\partial w}{\partial n} + k_s(w - w_s) = 0, \tag{159}$$

where k_s is the support stiffness and w_s is the prescribed support displacement.

The corresponding time-dependent problem is the 2D damped wave equation

$$\rho\frac{\partial^2 w}{\partial t^2} + c\frac{\partial w}{\partial t} - T\Delta w = p(x,y,t), \tag{160}$$

where ρ and c denote the density and damping coefficient, respectively. The 2D spatial extension of ANNO follows Appendix B.5, while the analytical modal update for wave dynamics follows Appendix B.4. All network parameters learned from the steady-state problems are frozen during time evolution.

F.2 Steady Dataset Construction

The steady-state dataset contains total-force-normalized uniform-pressure, single-contact, and dual-contact loads in a $1:4:5$ ratio. Local contacts are represented by smooth tanh disks and normalized by discrete area integration such that

$$\sum_{i,j} p_{ij}\Delta A = F, \tag{161}$$

where F is the prescribed total force. For dual-contact loads, the total force is randomly divided between two contact regions while maintaining sufficient separation between their centers. The main dataset settings are summarized in Table 18.

Table 18: Steady-state dataset configuration for Experiment 3.

Item	Setting
Domain	$L_x = 0.18,\ L_y = 0.12$
Grid	72×48
Membrane tension	$T \in [800, 1600]$, log-uniform
Support parameter	$\eta_i \in [0.5, 20]$, four edges independently
Support displacement	$w_s \in [0, 2 \times 10^{-4}]$
Total force	$F \in [3, 12]$
Contact radius	$r \in [0.012, 0.022]$
Contact edge width	0.005
Load-family ratio	uniform : single : dual $= 1 : 4 : 5$
Train / validation / test	5,000 / 500 / 500

Table 19: Dynamic dataset configuration for Experiment 3.

Item	Setting
Spatial grid	72×48
Membrane tension	$T \in [800, 1600]$, log-uniform
Support parameter	$\eta_i \in [0.5, 20]$, four edges independently
Support displacement	$w_s \in [0, 2 \times 10^{-4}]$
Density	$\rho = 2.0$
Damping coefficient	$c = 1.5$
Spatial load families	single contact / dual contact, balanced
Characteristic time	$\tau = 2\pi/\omega_1$
Final time	$T_{\text{final}} = 3\tau$
Temporal resolution	128 intervals / 129 stored fields
Localized Pulse	Gaussian temporal pulse
Harmonic Excitation	$\omega_f/\omega_1 \in [0.7, 1.3]$
Frequency Sweep	$0.7\omega_1 \to 1.3\omega_1$
Train / validation / test	96 / 24 / 192 trajectories
Per-case train / validation / test	32 / 8 / 64 trajectories

F.3 Dynamic Dataset

The dynamic data use the same domain, spatial grid, membrane-tension range, and elastic-support parameter ranges as the steady-state problems, while the spatial loads are restricted to single- and dual-contact cases. The two load families are balanced within each dynamic case.

Each trajectory defines its characteristic time from its fundamental angular frequency,

$$\tau = \frac{2\pi}{\omega_1}, \tag{162}$$

and uses

$$T_{\text{final}} = 3\tau, \qquad \Delta t = \frac{3\tau}{128}. \tag{163}$$

Thus, each trajectory spans three fundamental periods with 128 time intervals and 129 stored fields.

We consider three types of time-dependent excitation. **Localized Pulse** uses a Gaussian temporal envelope centered at $0.25T_{\text{final}}$. **Harmonic Excitation** uses an excitation frequency satisfying $\omega_f/\omega_1 \in [0.7, 1.3]$. **Frequency Sweep** continuously increases the excitation frequency from $0.7\omega_1$ to $1.3\omega_1$. The detailed settings are summarized in Table 19.

The initial displacement equals the constant lifting field w_s, so the initial residual is zero, and the initial velocity is zero. The spatial load remains fixed within each trajectory, while its temporal amplitude follows the selected excitation type.

Table 20: Model configurations and parameter counts in Experiment 3.

Model	Main configuration	Parameters
ANNO	12 modes per direction; 144 spatial modes	1,640,310
ST-FNO	modes $(8, 10, 10)$ in (y, x, t); width 8; 4 Fourier layers	1,640,193
POD-DeepPropNet	144 POD modes; 8-step training rollout	1,652,331

Table 21: Mean full-trajectory relative L_2 error for the three dynamic excitation cases in Experiment 3.

Dynamic case	ANNO	ST-FNO	POD-DeepPropNet
Localized Pulse	1.4898%	29.1344%	326.3571%
Harmonic Excitation	2.3496%	33.6624%	187.8865%
Frequency Sweep	1.7544%	27.2749%	182.4213%
Overall	1.8646%	30.0239%	232.2216%

F.4 Model Configuration

ANNO uses 12 learned modes along each spatial direction, giving $12^2 = 144$ two-dimensional modes. The transient baselines are the same ST-FNO and POD-DeepPropNet introduced in Experiment 2, with the necessary extension to the 2D membrane problem. Their main configurations and parameter counts are summarized in Table 20.

ANNO is trained on 5,000 steady-state solution fields and uses no transient trajectories for network training. In contrast, ST-FNO and POD-DeepPropNet are trained on 96 complete transient trajectories, each containing 129 stored solution fields, corresponding to 12,384 transient solution fields in total. Other baseline settings follow Appendix E and are not repeated here.

F.5 Evaluation Metrics and Quantitative Results

The pointwise absolute error, full-trajectory relative L_2 error, and instantaneous relative L_2 error follow the definitions in Appendix E. Table 21 summarizes the mean full-trajectory displacement errors on the 192 test trajectories.

Figure 4 also compares the learned spatial modes with numerical eigenmodes and examines the resulting wave dynamics. For the mechanical-energy comparison, the total energy is defined as

$$E(t) = E_{\mathrm{kin}}(t) + E_{\mathrm{strain}}(t), \tag{164}$$

where the kinetic energy is

$$E_{\mathrm{kin}}(t) = \frac{1}{2}\rho L_{\mathrm{ref}}^2 \sum_{i,j} \dot{w}_{ij}^2(t)\Delta x \Delta y, \tag{165}$$

and the membrane strain energy is

$$E_{\mathrm{strain}}(t) = \frac{T}{2}\sum_{i,j}\left[\left(\frac{\partial w}{\partial x}\right)_{ij}^2 + \left(\frac{\partial w}{\partial y}\right)_{ij}^2\right]\Delta x \Delta y. \tag{166}$$

For visualization, both the numerical solution and ANNO prediction use the maximum ground-truth mechanical energy of the corresponding trajectory as the common normalization reference,

$$\bar{E}(t) = \frac{E(t)}{\max_t E_{\mathrm{GT}}(t)}. \tag{167}$$

This common normalization allows the mechanical-energy evolution of the numerical solution and ANNO prediction to be compared directly.

G SUPPLEMENTARY DETAILS FOR EXPERIMENT 4

This appendix provides additional details on the physical model, dataset construction, and evaluation for the 2D battery cooling-plate problem in Section 4.4. The 2D implementation and diffusion dynamics follow Appendix B.5 and Appendix B.2.3, respectively, and are not rederived here.

G.1 PROBLEM SETUP

Experiment 4 considers an equivalent 2D bottom cooling plate for a battery module. A 6×4 array of cylindrical cells is placed above the plate, and each cell transfers heat to the plate through a local circular region. Depending on the cooling layout, the plate edges use either water-cooled Robin boundaries or adiabatic Neumann boundaries.

The steady-state temperature satisfies

$$-k\Delta T(x, y) = Q(x, y), \tag{168}$$

where k is the effective thermal conductivity and $Q(x, y)$ is the equivalent volumetric heat source. A water-cooled edge satisfies

$$k\frac{\partial T}{\partial n} + h(T - T_c) = 0, \tag{169}$$

where h is the convective heat-transfer coefficient and T_c is the coolant temperature. Adiabatic edges satisfy

$$\frac{\partial T}{\partial n} = 0. \tag{170}$$

The corresponding transient problem is

$$C_v\frac{\partial T}{\partial t} - k\Delta T = Q(x, y, t), \tag{171}$$

where $C_v = 2.0 \times 10^6$ is the volumetric heat capacity, and the corresponding thermal diffusivity is $D = k/C_v$. During transient operation, both the heat source and the cooling conditions may vary with time.

G.2 STEADY DATASET CONSTRUCTION

The cooling plate has dimensions 0.18×0.12 m, thickness 0.07 m, and a 72×48 spatial grid. Twenty-four cells are arranged in a 6×4 layout. The total heat-generation power of each cell is mapped to its circular footprint such that

$$\sum_{i,j} Q_{ij}^{(m)} \Delta x \Delta y \, d = P_m, \tag{172}$$

where P_m is the prescribed heat-generation power of the mth cell and d is the plate thickness.

Four source scenarios are considered: **Normal**, in which all cell powers are sampled from 1-5 W; **Single High**, in which one randomly selected cell is increased to 8-15 W; **Cluster High**, in which a local group of adjacent cells is assigned 7-13 W; and **Partially Inactive**, in which 3-8 randomly selected cells are set to zero power.

Three cooling topologies are used: two adjacent water-cooled edges, three water-cooled edges, and four water-cooled edges. For the first two topologies, rotational orientations are balanced so that cooling is not tied to a fixed coordinate direction. The main continuous parameters are summarized in Table 22.

Table 22: Steady-state dataset configuration for Experiment 4.

Item	Setting
Plate size	0.18×0.12 m
Plate thickness	0.07 m
Grid	72×48
Cell arrangement	6×4
Cell pitch / radius	0.027 m / 0.0105 m
Thermal conductivity	$k \in [5, 20]$, log-uniform
Coolant temperature	$T_c \in [288.15, 303.15]$ K
Base heat-transfer coefficient	$h_{\text{base}} \in [300, 1500]$
Edge multiplier	$m_h \in [0.60, 1.67]$
Source scenarios	Normal / Single High / Cluster High / Partially Inactive
Cooling topologies	Two-edge / Three-edge / Four-edge cooling
Train / validation / test	4,096 / 512 / 512 solution fields

Table 23: Dynamic operating conditions in Experiment 4.

Dynamic case	Setting
Cold	The initial plate temperature is uniformly set to $T_{\text{ref}} = 298.15$ K and then evolves under the prescribed heat source and cooling conditions.
Pulse	The base heat source is temporarily increased during the trajectory.
Periodic	A periodic modulation is applied to the base heat source.
Cooling	The coolant temperature and heat-transfer coefficient vary continuously with time.
Combined	A temporary power pulse and a cooling-condition transition occur within the same trajectory.

G.3 Dynamic Dataset Construction

The transient dataset uses the same plate geometry, source scenarios, cooling topologies, and material-parameter ranges. Five dynamic cases are introduced:

For the **Pulse** case, the heat source is

$$Q(x, y, t) = Q_0(x, y) \left[1 + r_p I_{0.30<t/T_{\text{final}}<0.55}\right], \tag{173}$$

where $r_p \in [1.5, 2.5]$. The active heat source is therefore 2.5-3.5 times its base level. For the **Periodic** case,

$$Q(x, y, t) = Q_0(x, y) \left[1 + A_p \sin\left(4\pi t / T_{\text{final}}\right)\right], \tag{174}$$

where $A_p \in [0.20, 0.50]$, giving two periods over each trajectory.

For the **Cooling** case, a normalized ramp $r(t)$ remains zero over the first 30% of the trajectory, increases linearly from 0 to 1 over 30%–70%, and remains one thereafter. The coolant temperature and heat-transfer coefficient evolve as

$$T_c(t) = T_{c,0} + \Delta T_c r(t), \tag{175}$$

$$h(t) = h_0 \left[1 + (m_h - 1) r(t)\right], \tag{176}$$

where $\Delta T_c \in [-7, 7]$ K and $m_h \in [0.60, 1.40]$. The **Combined** case applies the power pulse and cooling transition simultaneously.

Each trajectory uses a characteristic time determined from the slowest positive decay rate of its current thermal diffusion operator. Let μ_1 denote the smallest positive decay rate of the corresponding semi-discrete operator. We define

$$\tau = \frac{1}{\mu_1}, \tag{177}$$

and set

Table 24: Steady-state model configuration for Experiment 4.

Model	Main configuration	Parameters
ANNO	12 modes per direction; 144 spatial modes; hidden dimension 96	1,640,310
FNO	2D FNO with parameter-matched configuration	1,672,707
DeepONet	2D DeepONet with parameter-matched configuration	1,638,369

Table 25: Transient model configuration for Experiment 4.

Model	Main configuration	Parameters
ANNO	Frozen steady-state ANNO; analytical diffusion recurrence	1,640,310
ST-FNO	modes $(8, 10, 10)$ in (y, x, t); width 8; 4 Fourier layers	1,640,185
POD-DeepPropNet	144 POD modes; 8-step training rollout	1,657,226

$$T_{\text{final}} = 3\tau, \qquad \Delta t = \frac{3\tau}{128}. \tag{178}$$

For cases with time-varying cooling conditions, the characteristic times of the initial and final cooling states are both evaluated, and the slower time scale is used. Each trajectory therefore contains 128 time intervals and 129 stored solution fields.

The dataset is stratified over dynamic case, cooling topology, cooling orientation, and source scenario. Their Cartesian product gives

$$5 \times 3 \times 4 \times 4 = 240 \tag{179}$$

strata. Each stratum is repeated $4/1/2$ times for training, validation, and testing, respectively, giving 960/240/480 complete trajectories.

For the non-cold-start cases, the initial temperature is the steady-state solution under the initial heat source and cooling conditions. In the Cold case, the plate starts uniformly at T_{ref}.

G.4 Reuse of the Steady-State Model

The same frozen ANNO2D learned from the steady-state dataset is reused for all transient cases. At each time interval, the current effective source and boundary conditions are passed through the frozen model to obtain the current spatial modes, eigenvalues, and instantaneous steady-state modal coefficients. The transient modal state is then advanced using the analytical diffusion recurrence described in Appendix B.2.3.

For time-varying cooling conditions, $h(t)$ changes the homogeneous Robin boundary operator. The frozen model is therefore reevaluated at the current boundary parameters to obtain the corresponding boundary-conditioned modes and eigenvalues at each interval, within the steady-state parameter range used for training.

G.5 Baselines and Evaluation

The steady-state and transient comparison models are summarized separately in Tables 24 and 25. The steady-state baselines use the same data splits and input variables as ANNO, while the transient baselines are trained directly on complete temperature trajectories.

All three steady-state models are trained on the same 4,096 training fields and evaluated on the same 512 test fields. Their parameter budgets are closely matched.

For transient prediction, ST-FNO and POD-DeepPropNet are trained on 960 complete trajectories with 129 stored solution fields per trajectory, corresponding to 123,840 transient solution fields. ANNO instead reuses the model trained on 4,096 steady-state solution fields and uses no transient trajectories for network training. ST-FNO and POD-DeepPropNet use closely matched parameter budgets of 1,640,185 and 1,657,226 parameters, respectively.

Table 26: Mean full-trajectory relative L_2 error in Experiment 4.

Dynamic case	ANNO	ST-FNO	POD-DeepPropNet
Cold	1.3075%	15.1773%	35.4972%
Pulse	1.0511%	5.8112%	15.6504%
Periodic	0.8082%	5.7260%	9.3663%
Cooling	0.8772%	7.7017%	21.2057%
Combined	1.2090%	8.0156%	22.7906%
Overall	1.0506%	8.4864%	20.9020%

The full-trajectory relative L_2 error is computed as in the preceding experiments. Table 26 reports the mean errors over the 480 formal test trajectories.

For time-resolved evaluation, the instantaneous relative L_2 error is computed over the spatial field at each saved time. The center curve reports the median over the 480 test trajectories, and the shaded region denotes the 25th–75th percentile interval.

H Supplementary Details for Experiment 5

This appendix provides additional details for the component ablations and the learned-versus-numerical modal study in Experiment 5. All experiments use the 2D thermal setting and datasets of Experiment 4. Unless otherwise stated, only the component under study is modified, while the remaining architecture and training procedure are kept unchanged.

H.1 Removing Analytical Lifting

Full ANNO separates the nonhomogeneous boundary contribution through the Analytical Lifting Module,

$$u = \ell + v, \tag{180}$$

and reconstructs only the homogeneous residual v. In **No Lift.**, this decomposition is removed and the modal reconstruction directly predicts the complete solution,

$$\hat{u}(x, y) = \sum_{i=1}^{K} \sum_{j=1}^{K} \hat{a}_{ij} \hat{\phi}_i^x(x) \hat{\phi}_j^y(y). \tag{181}$$

Because the nonhomogeneous boundary contribution no longer enters explicitly through ℓ, the Learnable Coefficient Network additionally receives the normalized boundary parameters on all four edges. The ground-truth lifting is not used as a model input and is retained only for common residual-error evaluation.

H.2 Replacing Trigonometric Modes with an MLP

ANNO represents each one-dimensional spatial mode using the boundary-conditioned trigonometric form

$$\hat{\phi}_k(x) = \hat{A}_k \cos(\hat{\omega}_k x) + \hat{B}_k \sin(\hat{\omega}_k x). \tag{182}$$

In **MLP Basis**, only this representation is replaced by a coordinate MLP,

$$\hat{\phi}_k(x) = \mathrm{MLP}_k(x, \mathbf{b}_{\mathrm{bc}}), \tag{183}$$

where $\mathbf{b}_{\mathrm{bc}}$ denotes the encoded homogeneous boundary information. The lifting, modal projection, coefficient prediction, and reconstruction procedures remain unchanged. The resulting MLP modes use the same discrete normalization, spatial operator, and Rayleigh-quotient eigenvalue evaluation as the full model.

Table 27: Mode–coefficient configurations in Experiment 5.

Configuration	Spatial modes	Modal coefficients
Learned / Learned	Learned	Learned
Numerical / Learned	Numerical	Learned
Learned / Analytical	Learned	Analytical
Numerical / Analytical	Numerical	Analytical

H.3 Removing the Frequency-Band Strategy

Full ANNO assigns different ordered frequency intervals to successive learned modes. In **No Freq.**, the trigonometric representation and boundary conditioning are retained, but all modes share the same frequency range,

$$\hat{\omega}_k \in [0, (K + 1.25)\pi], \qquad k = 1, \ldots, K. \tag{184}$$

This ablation therefore removes only the mode-specific frequency partition.

H.4 Removing the Eigenstructure Constraint

The full ANNO objective is

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda_{\text{eig}} \mathcal{L}_{\text{eig}}, \tag{185}$$

where the final eigenstructure weight is $\lambda_{\text{eig}} = 0.1$. The eigenstructure loss combines uniformly weighted and coefficient-weighted terms,

$$\mathcal{L}_{\text{eig}} = (1 - \eta)\mathcal{L}_{\text{eig}}^{\text{uniform}} + \eta \mathcal{L}_{\text{eig}}^{\text{weighted}}, \tag{186}$$

with $\eta = 0.8$ in the final training stage.

In **Data Loss**, the eigenstructure constraint is removed by setting $\lambda_{\text{eig}} = 0$,

$$\mathcal{L} = \mathcal{L}_{\text{data}}. \tag{187}$$

The full model follows the staged training schedule described in Appendix C.3, whereas Data Loss uses only the data objective throughout training.

H.5 Learned and Numerical Modal Components

To further separate the roles of the spatial modes and modal coefficients, we construct a 2×2 comparison in which the source of each component is varied independently. Spatial modes are either obtained from the Learnable Mode Network or computed numerically from the discretized spatial operator. Modal coefficients are either predicted by the Learnable Coefficient Network or obtained by directly solving the projected steady-state modal system. The four configurations are summarized in Table 27.

For the numerical-mode configurations, the first K eigenpairs are computed from the same cell-centered finite-volume spatial operator used by ANNO. The numerical modes use the same discrete normalization, modal projection, tensor-product construction, and reconstruction conventions as the learned modes. This isolates the effect of replacing the Learnable Mode Network with a per-sample numerical eigensolve.

For the analytical-coefficient configurations, the modal coefficients are obtained from

$$\mathbf{K}_\phi \mathbf{a} = \mathbf{q}_\phi, \tag{188}$$

where $\mathbf{q}_\phi$ denotes the projected effective source and $\mathbf{K}_\phi$ is the spatial operator represented in the selected modal space. The corresponding coefficient vector is obtained by directly solving Eq. 188. In the learned-coefficient configurations, the same Learnable Coefficient Network used by ANNO predicts the modal coefficients.

The **Learned / Learned** configuration is the full ANNO model. In **Numerical / Learned**, the learned modes are replaced by numerical eigenmodes and a new Learnable Coefficient Network is trained using the same steady-state training data. In **Learned / Analytical**, the spatial modes

Table 28: Median steady-state error, full-trajectory error, and online computation time for the four mode–coefficient configurations.

Configuration	Median steady error	Median trajectory error	Median online time
Learned / Learned	0.3737%	0.8384%	0.488 ms
Numerical / Learned	0.6529%	1.1962%	2.041 ms
Learned / Analytical	0.3209%	0.8561%	0.701 ms
Numerical / Analytical	0.6507%	1.1946%	2.204 ms

are learned while the coefficients are obtained analytically; the mode-related networks are trained from scratch under the same eigenstructure-constrained objective. The **Numerical / Analytical** configuration uses numerical eigenmodes together with the analytical modal solve and contains no learned mode or coefficient predictor.

All four configurations are first evaluated on the same 512 steady-state test samples. They are then reused for the 480 transient trajectories from Experiment 4 using the same analytical diffusion recurrence. Thus, the transient comparison changes only the source of the spatial modes and instantaneous steady-state coefficients, while the time-evolution rule and reconstruction remain unchanged.

H.6 Evaluation and Quantitative Results

For the component ablations, we report the steady-state relative L_2 error and full-trajectory relative L_2 error using the same definitions as in the preceding experiments. The learned-versus-numerical study further compares the four mode–coefficient configurations in Table 28 in terms of steady-state accuracy, transient accuracy, and online computation time.

For modal diagnostics, each candidate mode set is aligned with the numerical reference using the discrete modal overlap. The corresponding permutation and sign correction are then applied before comparing eigenvalues and modal coefficients. Online time is measured per steady-state sample and includes modal construction, coefficient evaluation, and reconstruction. Learned modes are obtained through a forward pass of the Learnable Mode Network, whereas numerical modes require a per-sample eigensolve.

In terms of accuracy, the learned-mode configurations remain at a comparable error level to the corresponding numerical-mode configurations. Although some differences exist across the four combinations, the results show that the Learnable Mode Network can achieve competitive prediction accuracy without requiring a per-sample numerical eigensolve. Under the current discretization and finite-mode truncation, the final prediction error is jointly affected by spatial discretization, modal truncation, projection, and reconstruction, so numerical modes do not necessarily yield the lowest end-to-end error.

In terms of computational efficiency, the learned-mode configurations have substantially lower online cost. Numerical-mode approaches require a new eigensolve for each input sample, whereas the Learnable Mode Network obtains the spatial modes through a single forward pass. ANNO therefore replaces repeated per-sample numerical mode construction with reusable network inference and significantly reduces the online computational cost.