LIBRARY-BASED CORRECTOR NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

PDEs with boundary, corner, or interior layers exhibit localized multiscale behavior that is hard to capture with standard neural architectures. Classical asymptotic analysis decomposes such solutions into a smooth component and a problem-specific analytical corrector, but designing correctors requires case-by-case analysis and does not transfer across problems. We propose the Library-based Corrector Network (LibCorrNet), which replaces the problem-specific corrector with a shared, family-level structural prior. LibCorrNet combines a Fourier neural operator for the smooth component with a corrector branch that evaluates a fixed candidate-function library on a learned stretched coordinate, combined via adaptive, input-conditioned weights. The same library and architecture are reused across different PDE classes without redesigning a corrector module for each problem. Across several representative classes of singularly perturbed and multiscale PDEs, LibCorrNet consistently outperforms strong neural operator baselines, with the largest gains observed under limited training data. Ablation studies show complementary contributions from coordinate learning, candidate transformation, and adaptive weighting, and the method remains effective even when the exact analytical corrector is excluded from the library, supporting its use as a reusable structural prior rather than a problem-specific design.

1 INTRODUCTION

Singularly perturbed partial differential equations (SPPDEs) arise as fundamental models in a broad range of scientific and engineering applications, including fluid dynamics, reaction–diffusion systems in biology, and chemical kinetics Schlichting & Gersten (2016); Batchelor (2000); Roos et al. (2008); Morton (1996); Trangenstein & Bell (1989); Gie et al. (2018); Ducrot & Langlais (2012); Gaucel & Langlais (2007); Mimura (2003). These equations are particularly important for describing multiscale phenomena in which small physical scales induce sharp transitions and localized structures. In this work, we consider representative SPPDEs of the form

$$\begin{aligned} -\varepsilon\Delta u_\varepsilon + b(\mathbf{x})\cdot\nabla u_\varepsilon + c(\mathbf{x})u_\varepsilon + f(\mathbf{x}, u_\varepsilon) = 0 \quad &\text{in } \Omega, \\ u_\varepsilon = 0 \quad &\text{on } \partial\Omega, \end{aligned} \tag{1}$$

where $\varepsilon > 0$ is a small perturbation parameter multiplying the highest-order differential term, and $b(\mathbf{x})$ denotes a convection or transport field. As ε decreases, the problem becomes increasingly stiff, and the solution may exhibit sharp spatial variations near the boundary or in the interior of the domain even when the PDE coefficients and forcing terms are smooth. In this work, we consider a collection of controlled PDE benchmarks exhibiting boundary layers, corner layers, interior layers, nonlinear reaction–diffusion structures, and Helmholtz-type multiscale solutions. Accurately resolving such localized structures has motivated the development of specialized numerical methods. One important class consists of layer-adapted and adaptive mesh techniques, such as Shishkin meshes Linß & Stynes (1999); Linß (1999); Wang et al. (2024), which allocate additional spatial resolution to regions in which layers are expected to form. These approaches can provide accurate approximations for strongly singular problems, but increasingly fine spatial resolution may be required as the characteristic layer width decreases with ε. Another important strategy is provided by enriched numerical methods based on asymptotic analysis Shih & Kellogg (1987); Jung & Temam (2009); Jung et al. (2017); Shih et al. (2011); Hong et al. (2014); Jung et al. (2016); Chekroun et al. (2020). Rather than approximating the entire multiscale solution using a single numerical representation, these methods exploit a decomposition of the form

$$u_\varepsilon \approx u_{\text{smooth}} + \Phi, \tag{2}$$

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.

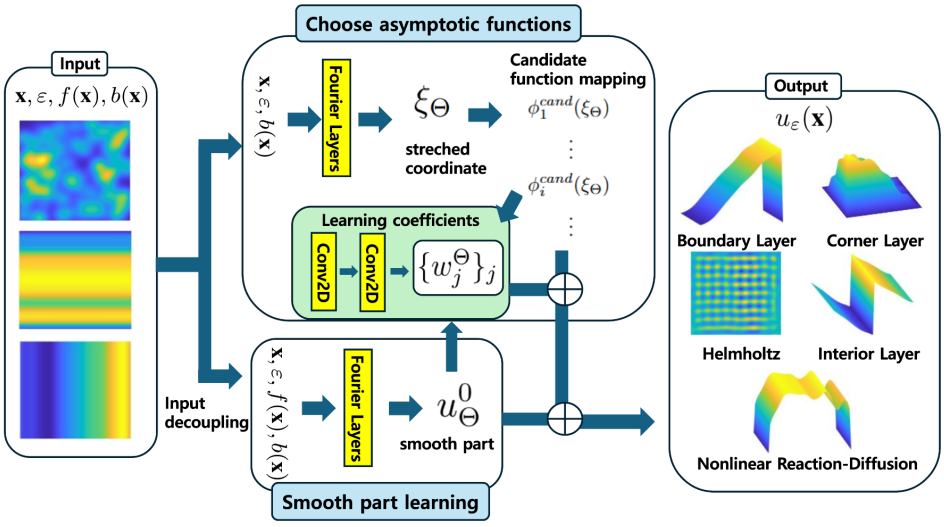


Figure 1: Overview of the proposed LibCorrNet framework and representative multiscale solution structures considered in this work, including boundary layers, corner layers, interior layers, nonlinear reaction–diffusion problems, and Helmholtz-type problems. Rather than prescribing a problem-specific analytical corrector, LibCorrNet uses a shared candidate-function library together with a learned coordinate representation and adaptive candidate composition.

where u_{smooth} represents a globally smooth component and Φ captures localized singular behavior. Such decompositions provide an effective representation of sharp layers. However, the functional form of Φ and its associated stretched coordinates generally depend on the PDE coefficients, boundary conditions, and singular perturbation structure. Consequently, applying this strategy to a new problem may require additional asymptotic analysis to determine an appropriate corrector representation and spatial transformation. Neural-network-based approaches, including physics-informed neural networks and neural operators Raissi et al. (2019); Li et al. (2021); Raonic et al. (2023); Hao et al. (2023); Lu et al. (2021), provide an alternative framework for approximating PDE solution operators. However, sharply localized structures remain challenging for standard neural representations. Recent asymptotic-informed neural approaches Li et al. (2024); Gie et al. (2024); Jung et al. (2025) address this difficulty by combining neural representations of global solution behavior with analytically motivated singular structures. While effective, these approaches generally incorporate prior information about suitable corrector profiles or spatial transformations for the problem under consideration. This observation motivates the central question of this work: *Can problem-specific analytical correctors and their spatial transformations be replaced by a reusable structural prior that can be shared across different singularly perturbed problems?* Our objective is not to eliminate asymptotic prior knowledge. Instead, we replace a problem-specific analytical prior with a shared family-level structural prior. Rather than explicitly prescribing the exact analytical corrector associated with each PDE, we construct a candidate library from recurring function families appearing in the singular and multiscale structures considered in this work and learn their problem-dependent realization from data. Conceptually, this changes the modeling paradigm from

$$\text{problem-specific analytical corrector} \quad \longrightarrow \quad \text{shared candidate-based structural prior.} \tag{3}$$

This perspective is conceptually motivated by library-based system identification, most notably Sparse Identification of Nonlinear Dynamics (SINDy) Brunton et al. (2016). SINDy constructs a candidate-function library and determines system-dependent active terms from data rather than prescribing a single functional form in advance. We adopt this library-based viewpoint for the representation of localized solution components. Our use of this viewpoint, however, is conceptual rather than algorithmically equivalent to SINDy. LibCorrNet does not seek to recover a sparse symbolic governing equation, nor does it perform discrete candidate selection. Instead, the candidate library serves as a structured inductive bias, and the contributions of its elements are determined through continuous neural weights. The objective is therefore not symbolic corrector discovery, but the construction of a flexible candidate-based representation of localized solution structures. An

additional challenge arises because an asymptotic corrector is typically expressed not simply as $\phi_i(\mathbf{x})$, but as

$$\phi_i\big(\xi_i(\mathbf{x},\varepsilon)\big), \tag{4}$$

where ξ_i is a stretched coordinate associated with the location and spatial scale of a localized structure. Thus, constructing a corrector representation requires not only a suitable functional profile but also the coordinate in which that profile is evaluated. Motivated by this structure, we propose the *Library-based Corrector Network (LibCorrNet)*, which represents the solution as

$$\hat{u}_\varepsilon(\mathbf{x}) = u^0_\Theta(\mathbf{x}) + \sum_{j=1}^{N_c} w_j^\Theta(\mathbf{x})\phi_j^{\mathrm{cand}}\big(\xi_\Theta(\mathbf{x})\big). \tag{5}$$

Here, u^0_Θ represents the learned global component, $\{\phi_j^{\mathrm{cand}}\}_{j=1}^{N_c}$ is a shared candidate-function library, ξ_Θ is a learned coordinate representation, and w_j^Θ denotes the learned contribution of each candidate. A Fourier neural operator models the global solution component, while a second operator learns the coordinate representation used to evaluate the candidate functions. Their contributions are subsequently determined by a neural weighting module conditioned on the global and candidate features. The same candidate library and basic architecture are reused across the PDE problems considered in this work. Our focus is therefore not zero-shot generalization to unseen PDEs, but *architectural reusability*: the ability to apply the same candidate-based formulation across different PDE classes without redesigning a problem-specific corrector module. We further examine the dependence of the proposed representation on candidate-library specification through controlled experiments in which the analytical profile associated with a target problem is removed from the library. In addition, we study the behavior of LibCorrNet across perturbation scales and limited-data regimes. Our main contributions are summarized as follows:

- We introduce a shared candidate-based structural prior for singularly perturbed solution representations, replacing the need to explicitly prescribe a different analytical corrector within the architecture for each considered PDE problem.
- We propose LibCorrNet, which combines a Fourier-based global representation with a candidate-based corrector representation and jointly learns the coordinate representation and continuous candidate contributions from solution data.
- We systematically analyze the sensitivity of the proposed representation to candidate-library specification, including controlled experiments in which the corresponding analytical profile is excluded from the library.
- We evaluate the proposed framework across controlled PDE benchmarks with boundary, corner, interior, nonlinear, and oscillatory solution structures, and investigate its behavior across perturbation scales and limited-data regimes.

2 Related Works

Learning-based methods provide efficient surrogate models for PDEs, including FNO Li et al. (2021), CNO Raonic et al. (2023), GNOT Hao et al. (2023), and DeepONet Lu et al. (2021). Localized and multiscale approaches Liu-Schiaffini et al. (2024); Tripura & Chakraborty (2023), as well as adaptive mesh and tokenization frameworks Xu et al. (2025); Zhao et al. (2026), further improve the representation of spatially heterogeneous structures by allocating higher capacity to localized dynamic regions. However, singularly perturbed PDEs remain particularly challenging because small perturbation parameters can generate sharp boundary, interior, and corner layers that standard global approximations and uniform discretization schemes struggle to capture accurately Roohi & Mahdavi (2025); Nekoozadeh et al. (2023).

Classical asymptotic analysis addresses such multiscale behavior by decomposing solutions into global and localized corrector components. Recent asymptotic-informed learning methods Li et al. (2024); Gie et al. (2024); Jung et al. (2025) incorporate this structure into neural approximations. In particular, ComFNO Li et al. (2024) employs problem-specific analytical correctors and spatial transformations. In contrast, LibCorrNet uses a shared library of candidate functions and learns their contributions and coordinate representation from data, allowing the same candidate-based architecture to be applied across the PDE problems considered in this work without prescribing a separate analytical corrector for each problem.

3 METHODOLOGY

3.1 ASYMPTOTIC REPRESENTATION

Solutions of singularly perturbed PDEs can often be represented asymptotically as a combination of a smooth component and localized corrector terms. We consider a truncated representation of the form

$$u_{\mathrm{as}}(\mathbf{x}) = u_0(\mathbf{x}) + \sum_{i=1}^{N_w} w_i(\mathbf{x})\phi_i\big(\xi_i(\mathbf{x},\varepsilon)\big), \tag{6}$$

where u_0 denotes the smooth component, ϕ_i is a corrector profile, w_i denotes its corresponding coefficient, and ξ_i is a stretched coordinate. For suitable singular perturbation problems and asymptotic expansions, the truncated representation may satisfy an approximation estimate of the form

$$\|u_\varepsilon - u_{\mathrm{as}}\| \le C\varepsilon^{m+1}, \tag{7}$$

where the norm, constant C, and expansion order m depend on the problem and the corresponding asymptotic analysis. The functional forms of ϕ_i and ξ_i vary across singular perturbation problems. Representative corrector structures considered in this work are summarized in Table 5 in the Appendix. Across the considered problems, recurring function types include exponential, error-function, trigonometric, and hyperbolic profiles. This observation motivates the use of a shared candidate-function library as a family-level structural prior. We do not assume that this candidate library constitutes a universal set of correctors for arbitrary SPPDEs. Instead, it defines a structured representation space motivated by the singular and multiscale solution structures considered in this work. The coordinate representation is also an important component of the asymptotic structure. Even when the same candidate function is used, different coordinate transformations can produce localized structures at different spatial locations and scales. We therefore treat the coordinate representation, in addition to the candidate contributions, as a learned component of LibCorrNet.

3.2 MODEL ARCHITECTURE

Motivated by the asymptotic representation in Section 3.1, LibCorrNet approximates the solution as

$$\hat{u}_\varepsilon(\mathbf{x}) = u^0_\Theta(\mathbf{x}) + \sum_{j=1}^{N_c} w_j^\Theta(\mathbf{x})\phi_j^{\mathrm{cand}}\big(\xi_\Theta(\mathbf{x})\big), \tag{8}$$

where

$$\Phi_c = \big\{\phi_1^{\mathrm{cand}}, \ldots, \phi_{N_c}^{\mathrm{cand}}\big\} \tag{9}$$

denotes the shared candidate library. The model consists of a global-component branch, a coordinate-learning branch, candidate evaluation, and an adaptive weighting module.

3.2.1 INPUT PROCESSING

For the two-dimensional problems considered in this work, the complete input is represented as $a = (\mathbf{x}, \varepsilon, b_1, b_2, f)$. The global-component branch receives $a_{\mathrm{global}} = (\mathbf{x}, \varepsilon, b_1, b_2, f)$, whereas the coordinate-learning branch receives $a_{\mathrm{cor}} = (\mathbf{x}, \varepsilon, b_1, b_2)$. Thus, the forcing term f is included in the global-component branch but is not used as an input to the coordinate-learning branch.

3.2.2 GLOBAL COMPONENT LEARNING

The global solution component is predicted by the first FNO branch:

$$u^0_\Theta = \mathcal{N}_{\mathrm{global}}(a_{\mathrm{global}}). \tag{10}$$

The resulting representation is used directly in the final solution reconstruction and is also provided to the candidate-weighting module.

3.2.3 COORDINATE REPRESENTATION LEARNING

Localized structures in singularly perturbed PDEs are often described using problem-specific stretched coordinates. Instead of prescribing such coordinates analytically, LibCorrNet learns an adaptive coordinate representation directly from the PDE input:

$$\xi_\Theta = \mathcal{N}_{\mathrm{coord}}(a_{\mathrm{cor}}). \tag{11}$$

We use a fixed candidate library Φ_c shared across all PDE problems. Rather than representing exact analytical correctors, the library serves as a shared structural prior. Each candidate is evaluated in the learned coordinate space,

$$z_j(\mathbf{x}) = \phi_j^{\mathrm{cand}}\big(\xi_\Theta(\mathbf{x})\big), \qquad j = 1, \dots, N_c. \tag{12}$$

Thus, while the candidate functions provide reusable structural forms, the learned coordinates adapt their spatial representations to each PDE input. The resulting features are then combined through adaptive weighting to construct the final correction.

3.2.4 ADAPTIVE WEIGHTING AND COMPOSITION

For each candidate feature z_j, a deep CNN computes its contribution using both the predicted global component and the corresponding candidate feature:

$$w_j^\Theta = \mathrm{deepCNN}_\Theta\left(\left[u_\Theta^0, z_j\right]\right). \tag{13}$$

The candidate-based corrector component is then constructed as

$$\Phi_\Theta(\mathbf{x}) = \sum_{j=1}^{N_c} w_j^\Theta(\mathbf{x}) z_j(\mathbf{x}) = \sum_{j=1}^{N_c} w_j^\Theta(\mathbf{x}) \phi_j^{\mathrm{cand}}\big(\xi_\Theta(\mathbf{x})\big). \tag{14}$$

The candidate functions are therefore combined through learned continuous weights rather than through a discrete selection mechanism. The model does not assume that a single candidate must correspond exactly to the analytical corrector. Instead, the candidate features, learned coordinate representation, and continuous weights jointly define the candidate-based correction used in the final solution approximation.

3.2.5 SOLUTION RECONSTRUCTION

Finally, the predicted solution is obtained by combining the global and candidate-based components:

$$\hat{u}_\varepsilon(\mathbf{x}) = u_\Theta^0(\mathbf{x}) + \Phi_\Theta(\mathbf{x}) = u_\Theta^0(\mathbf{x}) + \sum_{j=1}^{N_c} w_j^\Theta(\mathbf{x}) \phi_j^{\mathrm{cand}}\big(\xi_\Theta(\mathbf{x})\big). \tag{15}$$

The complete model is trained end-to-end using the target PDE solution u_ε. No separate supervision is imposed on the intermediate global component u_Θ^0, the learned coordinate representation ξ_Θ, or the candidate weights w_j^Θ. These quantities are jointly optimized through the final solution prediction. Accordingly, LibCorrNet should not be interpreted as eliminating asymptotic prior knowledge. Instead, it replaces the explicit specification of a problem-specific analytical corrector and stretched coordinate with a shared candidate-based structural prior and a learned problem-dependent representation. This formulation preserves asymptotic structure as an inductive bias while reducing the need to redesign the corrector module for each PDE problem considered in this work.

4 EXPERIMENTS

We evaluate LibCorrNet on five representative classes of singularly perturbed and multiscale PDEs: Boundary Layer, Corner Layer, Interior Layer, Nonlinear Reaction–Diffusion (RD), and Helmholtz problems. Our experiments examine three aspects of the proposed representation: approximation under limited training data, robustness to the choice of candidate functions, and adaptation across perturbation regimes. Detailed PDE formulations and data-generation procedures are provided in Appendix A.

4.1 Experimental Setup

Unless stated otherwise, the perturbation parameter is set to $\varepsilon = 10^{-5}$ for the singular perturbation benchmarks. For Helmholtz, we use $\varepsilon = \omega^{-2}$ with $\omega^2 = 225$. All problems are defined on $\Omega = [0, 1]^2$ and discretized on a uniform 52×52 Cartesian grid. We denote by N_T the number of training samples used to train each model. We consider training budgets

$$N_T \in \{128, 256, 512, 1024\}, \tag{16}$$

covering both data-scarce and relatively data-rich regimes. The same candidate library is used across all PDE families:

$$\Phi_c = \{\mathrm{erf}, \cos, \sin, \mathrm{sech}, \exp, \tanh\}. \tag{17}$$

The library is motivated by recurring structures in the considered multiscale PDEs but is kept fixed across all main experiments. We therefore treat Φ_c as a shared structural prior rather than a problem-specific analytical corrector. For numerical stability, the exponential candidate is evaluated using a non-positive-domain constraint.

High-fidelity reference solutions are generated using an enriched finite element method with degree-P_5 polynomial approximation and mesh size $h = 1/104$, with $h = 1/208$ for $\varepsilon \geq 10^{-2}$ Jung & Temam (2009); Jung et al. (2017); Shih et al. (2011); Hong et al. (2014); Jung et al. (2016); Chekroun et al. (2020). We compare LibCorrNet with FNO Li et al. (2021), CNO Raonic et al. (2023), GNOT Hao et al. (2023), and ComFNO Li et al. (2024), using identical training, validation, and test partitions within each benchmark.

Generalization performance is measured by the relative L^2 error

$$\mathcal{E}_{\mathrm{rel}} = \frac{\|u_{\mathrm{pred}} - u_{\mathrm{true}}\|_{L^2(\Omega)}}{\|u_{\mathrm{true}}\|_{L^2(\Omega)}}. \tag{18}$$

Each PDE configuration is trained and evaluated over 10 independent runs and 1,024 held-out test instances. For benchmark families containing multiple configurations, we first average the test error over the 10 runs for each configuration and then average across configurations to obtain the reported family-level result.

4.2 Benchmark Performance

Table 1 compares the five methods across all benchmark families and training budgets.

LibCorrNet achieves the lowest error in 19 of 20 settings while using only 82K parameters. The improvements are particularly clear in data-scarce regimes, where generic operators must infer localized or oscillatory structures from few examples. The consistent performance across distinct PDE families suggests that the shared candidate representation provides a reusable structural bias rather than relying on a single problem-specific corrector. The Interior Layer result at $N_T = 1024$, where GNOT performs slightly better, also indicates that the advantage is most pronounced when structural guidance is valuable under limited data.

4.3 Ablation Studies and Structural Sensitivity

We examine the contributions of the learned coordinate representation, candidate transformations ϕ, and adaptive candidate weights w_i. Standard FNO retains only the global branch; w/o ϕ preserves the learned coordinates but removes the nonlinear candidate transformations; and w/o w_i removes adaptive candidate weighting.

Table 2 shows that the learned coordinate representation alone improves substantially over standard FNO in the available component-ablation settings. This indicates that transforming the input into an adaptive coordinate representation is useful for localized structures that are difficult to describe directly in the physical coordinate system.

The full LibCorrNet further improves over the coordinate-based variants. Thus, coordinate learning alone does not explain the performance gain: the candidate transformations provide structured correction profiles, while adaptive weighting determines how these profiles are combined according

Table 1: Average relative L^2 test errors across five PDE benchmarks (↓). For each PDE configuration, results are averaged over 10 independent runs and 1,024 held-out test instances. Best results are highlighted in **bold**.

PDE Family	Model	Input Signature	Params	$N_T = 128$	$N_T = 256$	$N_T = 512$	$N_T = 1024$
Boundary Layer	FNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	122K	8.70×10^{-3}	6.90×10^{-3}	3.30×10^{-3}	3.50×10^{-3}
	CNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	2.00M	1.54×10^{-2}	1.18×10^{-2}	6.40×10^{-3}	3.00×10^{-3}
	GNOT	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	1.34M	7.19×10^{-2}	7.07×10^{-2}	1.51×10^{-2}	1.13×10^{-2}
	ComFNO	$(\mathbf{x}, f(\mathbf{x}), f(\xi))$	231K	7.10×10^{-3}	5.00×10^{-3}	3.00×10^{-3}	2.70×10^{-3}
	LibCorrNet	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	**82K**	$\mathbf{5.06 \times 10^{-3}}$	$\mathbf{3.09 \times 10^{-3}}$	$\mathbf{2.58 \times 10^{-3}}$	$\mathbf{2.10 \times 10^{-3}}$
Corner Layer	FNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	122K	9.24×10^{-2}	5.90×10^{-2}	3.05×10^{-2}	1.85×10^{-2}
	CNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	2.00M	3.38×10^{-2}	1.84×10^{-2}	1.07×10^{-2}	8.00×10^{-3}
	GNOT	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	1.34M	3.60×10^{-1}	1.42×10^{-1}	7.66×10^{-2}	3.66×10^{-2}
	ComFNO	$(\mathbf{x}, f(\mathbf{x}), f(\xi))$	231K	5.70×10^{-2}	3.60×10^{-2}	1.90×10^{-2}	1.30×10^{-2}
	LibCorrNet	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	**82K**	$\mathbf{1.99 \times 10^{-2}}$	$\mathbf{1.17 \times 10^{-2}}$	$\mathbf{8.75 \times 10^{-3}}$	$\mathbf{6.51 \times 10^{-3}}$
Interior Layer	FNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	122K	7.26×10^{-2}	2.75×10^{-2}	1.30×10^{-2}	1.09×10^{-2}
	CNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	2.00M	1.33×10^{-1}	3.75×10^{-2}	1.59×10^{-2}	9.80×10^{-3}
	GNOT	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	1.34M	1.99×10^{-2}	8.40×10^{-3}	4.20×10^{-3}	$\mathbf{2.20 \times 10^{-3}}$
	ComFNO	$(\mathbf{x}, f(\mathbf{x}), f(\xi))$	231K	4.00×10^{-2}	1.60×10^{-2}	8.50×10^{-3}	6.50×10^{-3}
	LibCorrNet	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	**82K**	$\mathbf{8.81 \times 10^{-3}}$	$\mathbf{3.48 \times 10^{-3}}$	$\mathbf{3.73 \times 10^{-3}}$	2.59×10^{-3}
Nonlinear RD	FNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	122K	2.58×10^{-2}	1.55×10^{-2}	9.80×10^{-3}	4.80×10^{-3}
	CNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	2.00M	9.90×10^{-3}	5.80×10^{-3}	6.20×10^{-3}	1.60×10^{-3}
	GNOT	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	1.34M	1.39×10^{-1}	1.17×10^{-1}	1.72×10^{-1}	1.22×10^{-1}
	ComFNO	$(\mathbf{x}, f(\mathbf{x}), f(\xi))$	231K	1.50×10^{-2}	8.90×10^{-3}	6.00×10^{-3}	3.10×10^{-3}
	LibCorrNet	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	**82K**	$\mathbf{2.95 \times 10^{-3}}$	$\mathbf{2.02 \times 10^{-3}}$	$\mathbf{1.82 \times 10^{-3}}$	$\mathbf{1.15 \times 10^{-3}}$
Helmholtz ($\omega^2 = 225$)	FNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	122K	1.54×10^{-1}	9.17×10^{-2}	6.93×10^{-2}	3.85×10^{-2}
	CNO	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	2.00M	9.80×10^{-2}	6.24×10^{-2}	4.30×10^{-2}	3.52×10^{-2}
	GNOT	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	1.34M	8.78×10^{-1}	8.83×10^{-1}	1.21×10^{-1}	3.42×10^{-2}
	ComFNO	$(\mathbf{x}, f(\mathbf{x}), f(\xi))$	231K	1.10×10^{-1}	7.00×10^{-2}	5.00×10^{-2}	3.00×10^{-2}
	LibCorrNet	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$	**82K**	$\mathbf{9.79 \times 10^{-2}}$	$\mathbf{4.23 \times 10^{-2}}$	$\mathbf{4.11 \times 10^{-2}}$	$\mathbf{2.36 \times 10^{-2}}$

Table 2: Component ablation analysis. Relative L^2 error ($\times 10^{-2}$, ↓), averaged over 10 independent runs.

	Boundary Layer			Interior Layer			Corner Layer			Nonlinear RD		
Configuration	$N_T = 256$	$N_T = 512$	$N_T = 1024$	$N_T = 256$	$N_T = 512$	$N_T = 1024$	$N_T = 256$	$N_T = 512$	$N_T = 1024$	$N_T = 256$	$N_T = 512$	$N_T = 1024$
Standard FNO	0.69	0.33	0.35	2.75	1.30	1.09	5.90	3.05	1.85	1.55	0.98	0.48
w/o ϕ	0.95	0.89	0.77	0.85	0.62	0.55	4.31	3.39	2.20	0.55	0.42	0.33
w/o w_i	0.41	0.36	0.31	1.21	1.04	0.95	3.32	3.46	2.31	0.29	0.25	0.21
LibCorrNet (Full)	**0.309**	**0.258**	**0.210**	**0.348**	**0.373**	**0.259**	**1.17**	**0.875**	**0.651**	**0.202**	**0.182**	**0.115**

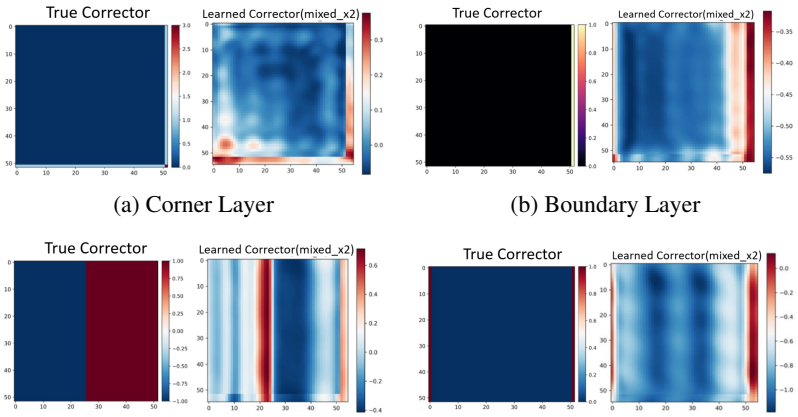
Table 3: Ablation study of the candidate library size N_c at $N_T = 1024$. Values denote the average relative L^2 test error (↓). The complete candidate library and results across all training-set sizes are reported in Appendix C.4.

N_c	**Boundary**	**Corner**	**Interior**	**Nonlinear RD**
0 (FNO)	3.50×10^{-3}	1.85×10^{-2}	1.09×10^{-2}	4.80×10^{-3}
2	2.19×10^{-3}	8.40×10^{-3}	6.70×10^{-3}	1.55×10^{-3}
4	2.20×10^{-3}	$\mathbf{8.17 \times 10^{-3}}$	6.56×10^{-3}	$\mathbf{1.49 \times 10^{-3}}$
6	$\mathbf{2.16 \times 10^{-3}}$	8.37×10^{-3}	$\mathbf{6.34 \times 10^{-3}}$	1.51×10^{-3}
Exact ϕ	2.70×10^{-3}	1.30×10^{-2}	6.50×10^{-3}	3.10×10^{-3}

to the local solution context. The results support the complementary roles of coordinate adaptation and candidate-based correction in the proposed architecture.

We further examine the sensitivity of LibCorrNet to the candidate library size N_c. We consider $N_c \in \{2, 4, 6\}$ across the boundary layer, corner layer, interior layer, and nonlinear reaction–diffusion benchmarks. These configurations are compared with the standard FNO without candidate correctors ($N_c = 0$) and a prescribed analytical corrector (Exact ϕ). The complete candidate pool, random selection procedure, and detailed results across different training-set sizes are provided in Appendix C.4.

Table 3 shows that introducing candidate correctors improves upon the standard FNO across all four PDE benchmarks. However, increasing N_c does not lead to monotonic improvement. Larger candidate sets are beneficial for some problems, whereas smaller sets perform comparably or slightly better for others. This suggests that the effectiveness of the library depends not only on the number



(a) Corner Layer (b) Boundary Layer

(c) Interior Layer (d) Nonlinear Reaction–Diffusion

Figure 3: **Qualitative comparison of analytical correctors and learned internal representations.** Left panels show representative analytical corrector structures; right panels show intermediate features produced by the candidate branch.

of candidates but also on the suitability of the selected functional profiles for the underlying localized structure.

The candidate-library configurations also remain competitive with, and in most cases improve upon, the prescribed Exact ϕ baseline. Therefore, the observed gains do not appear to require a single analytically matched corrector. Instead, the results support viewing the candidate library as a flexible structural prior whose realization is adapted through the learned coordinate representation and candidate weights. Further details on the candidate functions and their selection are provided in Appendix C.4.

4.4 Emergence of Localized Internal Representations

LibCorrNet is trained end-to-end using only the final PDE solution; no supervision is imposed on the learned coordinates, candidate weights, or intermediate corrector features. We therefore inspect the candidate branch to examine whether its learned features reflect the localized structure targeted by the architecture.

As shown in Figure 3, the intermediate features exhibit localized spatial patterns qualitatively aligned with the corresponding analytical corrector structures. Importantly, these patterns emerge without direct supervision on the intermediate representation. This behavior is consistent with the intended role of the candidate branch in representing localized components separately from the smoother global component. We do not interpret this correspondence as exact recovery of the analytical corrector. Since the decomposition is learned only through the final prediction objective, the intermediate representation is not uniquely identifiable. Rather, the visualization provides qualitative evidence that the architectural prior encourages features aligned with localized solution structure.

4.5 Cross-Regime Adaptation

We finally examine whether the learned representation remains useful when the perturbation scale changes. Following Herde et al. (2024), models are pretrained at $\varepsilon_{\text{source}} = 10^{-3}$ and fine-tuned using only $N_T = 128$ target samples at

$$\varepsilon_{\text{target}} \in \{10^{-1}, 10^{-2}, 10^{-4}, 10^{-5}\}. \tag{19}$$

These targets include both weaker and stronger perturbation regimes relative to the source. Throughout this section, TL denotes transfer learning, i.e., pretraining at the source perturbation scale followed by fine-tuning at the target perturbation scale. We report Accuracy Gain relative to FNO trained from

Table 4: Accuracy Gain ($AG_{N_T} \uparrow$) at $N_T = 128$ across target perturbation scales. Both models are pretrained at $\varepsilon = 10^{-3}$ and fine-tuned using 128 target samples. TL denotes transfer learning.

Target ε	Model	Boundary	Corner	Interior	Nonlinear RD
10^{-1}	TL FNO	1.13	2.52	1.62	2.21
	TL LibCorrNet	**1.32**	**3.21**	**2.82**	**3.32**
10^{-2}	TL FNO	1.53	3.16	1.89	1.93
	TL LibCorrNet	**2.36**	**4.33**	**3.77**	**4.22**
10^{-4}	TL FNO	1.34	4.54	2.17	1.74
	TL LibCorrNet	**2.69**	**6.49**	**5.94**	**4.00**
10^{-5}	TL FNO	1.21	4.11	2.53	2.46
	TL LibCorrNet	**2.45**	**6.90**	**9.56**	**4.29**

scratch:

$$AG_{N_T} = \frac{\mathcal{E}_{N_T}(\mathrm{FNO}_{\mathrm{scratch}})}{\mathcal{E}_{N_T}(\mathrm{Model})}. \tag{20}$$

Values above one indicate improvement over the common scratch baseline.

Table 4 shows that fine-tuned LibCorrNet consistently achieves larger accuracy gains than fine-tuned FNO relative to the same FNO-from-scratch reference. The difference is particularly pronounced for the Corner and Interior Layer problems at small ε, where increasingly localized structures must be represented. This trend is consistent with the design of LibCorrNet: the learned coordinates and candidate-based corrector provide flexibility for adapting the representation as the characteristic spatial scale changes. These results suggest that the learned representation remains useful outside the perturbation scale used during pretraining.

Because AG_{N_T} uses FNO trained from scratch as the common reference, it combines architectural and pretraining effects and therefore does not isolate the pure transfer gain of LibCorrNet. We thus interpret this experiment as evidence of cross-regime adaptability under limited target data rather than as a direct measurement of transfer gain.

5 Conclusion

We introduced LibCorrNet, a structured neural operator for singularly perturbed and multiscale PDEs that combines a global solution branch with a candidate-based corrector representation. Instead of prescribing problem-specific analytical correctors and stretched coordinates, LibCorrNet uses a shared candidate library, learned coordinate representations, and adaptive candidate weighting.

Across five PDE families, LibCorrNet achieves the lowest error in 19 of 20 evaluated settings with only 82K trainable parameters, with particularly strong performance under limited training data. The component ablations show that coordinate learning, candidate transformations, and adaptive weighting provide complementary contributions. Candidate-library experiments further indicate that exact analytical matching is not required, while the heterogeneous shared library provides the strongest overall performance. The learned intermediate features also exhibit localized structures without explicit intermediate supervision, and the cross-regime experiments suggest that the representation remains useful as the perturbation scale changes. Together, these results support LibCorrNet as a middle ground between generic neural operators and problem-specific asymptotic architectures: it retains useful structural information through a reusable candidate prior while learning how that structure should be expressed for each input.

Future work will consider adaptive candidate-library construction, resolution and geometry generalization, time-dependent multiscale PDEs, and problems whose singular structures extend beyond the candidate families considered here.

AI USE STATEMENT

Generative AI tools were used to assist with code development and debugging, experimental and methodological feedback, interpretation and presentation of experimental results, literature search and summarization, and manuscript editing and polishing. They were also used to improve the clarity, grammar, and organization of the text and to assist with the preparation of figures and tables. All AI-assisted code, technical content, experimental interpretations, and references were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work focuses on computational methods for singularly perturbed and multiscale partial differential equations and does not involve human subjects, personal data, or sensitive information. We are not aware of any direct ethical concerns arising from the methodology, datasets, or experiments presented in this work.

REPRODUCIBILITY STATEMENT

We provide the model architecture, training procedure, experimental settings, and evaluation metrics in the main paper and appendix. Detailed PDE formulations, data-generation procedures, numerical solver settings, and additional experimental configurations are provided in the appendix. The same candidate library and model hyperparameters are used across all PDE benchmarks unless explicitly stated otherwise. The source code will be made publicly available after the review process.

REFERENCES

I. Andronov, D. Bouche, and F. Molinet. *Asymptotic and Hybrid Methods in Electromagnetics*, volume 48 of *IEE Electromagnetic Waves Series*. Institution of Electrical Engineers (IEE), London, 2005. doi: 10.1049/PBEW051E.

George Keith Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, 2000.

D. Bouche, Y. Hong, and C.-Y. Jung. Asymptotic analysis of the scattering problem for the helmholtz equations with high wave numbers. *Discrete and Continuous Dynamical Systems - Series A*, 37(3): 1159–1181, 2017.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Mickaël D. Chekroun, Youngjoon Hong, and Roger Temam. Enriched numerical scheme for singularly perturbed barotropic quasi-geostrophic equations. *Journal of Computational Physics*, 416:109493, 2020.

A. Ducrot and M. Langlais. A singular reaction–diffusion system modelling prey-predator interactions: invasion and coextinction waves. *Journal of Differential Equations*, 253(2):502–532, 2012.

S. Gaucel and M. Langlais. Some remarks on a singular reaction–diffusion system arising in predator–prey modeling. *Discrete and Continuous Dynamical Systems - Series B*, 8(1):61–72, 2007.

Gung-Min Gie, Makram Hamouda, Chang-Yeol Jung, and Roger M. Temam. *Singular Perturbations and Boundary Layers*. Springer International Publishing, 2018.

Gwang-il Gie, Youngjoon Hong, Chang-Yeol Jung, and Temuulen Munkhjin. Semi-analytic pinn methods for boundary layer problems in a rectangular domain. *Journal of Computational and Applied Mathematics*, 450:115989, 2024.

Zhongkai Hao, Chengyang Ying, Zhengyi Wang, Hang Su, Yinpeng Dong, Songming Liu, Ze Cheng, Jun Zhu, and Jian Song. Gnot: A general neural operator transformer for operator learning. *arXiv preprint arXiv:2302.14376*, 2023.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. In Amir Globerson, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624. Curran Associates, Inc., 2024.

Youngjoon Hong, Chang-Yeol Jung, and Roger Temam. On the numerical approximations of stiff convection–diffusion equations in a circle. *Numerische Mathematik*, 127(2), 2014. ISSN 0029-599X. doi: 10.1007/s00211-013-0585-x. URL https://doi.org/10.1007/s00211-013-0585-x.

C.-Y. Jung and R. Temam. Interaction of boundary layers and corner singularities. *Discrete and Continuous Dynamical Systems - Series A*, 23:315–339, 2009.

C.-Y. Jung, E. Park, and R. Temam. Boundary layer analysis on nonlinear reaction–diffusion equations. *Advances in Nonlinear Analysis*, 2016. http://dx.doi.org/10.1515/anona-2015-0148.

Chang-Yeol Jung and Roger Temam. Convection-diffusion equations in a circle: the compatible case. *Journal de Mathématiques Pures et Appliquées*, 96(1):88–107, 2011.

Chang-Yeol Jung and Roger Temam. Singular perturbations and boundary layer theory for convection-diffusion equations in a circle: the generic noncompatible case. *SIAM Journal on Mathematical Analysis*, 44(6):4274–4296, 2012.

Chang-Yeol Jung, Junghwa Kim, and Eaint Phoo Ngon. Singular layer pinn methods for steep reaction–diffusion equations in a smooth convex domain. *Engineering Analysis with Boundary Elements*, 175:106178, 2025. ISSN 0955-7997. doi: https://doi.org/10.1016/j.enganabound.2025.106178. URL https://www.sciencedirect.com/science/article/pii/S0955799725000669.

C.Y. Jung, E. Park, and R. Temam. Boundary layer analysis of nonlinear reaction–diffusion equations in a polygonal domain. *Nonlinear Analysis: Theory, Methods & Applications*, 148:161–202, 2017.

David Kinderlehrer and Guido Stampacchia. *An Introduction to Variational Inequalities and their Applications*, volume 31 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1980 original.

Ye Li, Ting Du, Yiwen Pang, and Zhongyi Huang. Component fourier neural operator for singularly perturbed differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13691–13699, 2024.

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

T. Linß. An Upwind Difference Scheme on a Novel Shishkin-Type Mesh for a Linear Convection-Diffusion Problem. *Journal of Computational and Applied Mathematics*, 110(1):93–104, 1999. doi: 10.1016/S0377-0427(99)00198-3. URL http://dx.doi.org/10.1016/S0377-0427(99)00198-3.

T. Linß and M. Stynes. A Hybrid Difference Scheme on a Shishkin Mesh for Linear Convection-Diffusion Problems. *Applied Numerical Mathematics*, 31(3):255–270, 1999. doi: 10.1016/S0168-9274(98)00136-6. URL http://dx.doi.org/10.1016/S0168-9274(98)00136-6.

Miguel Liu-Schiaffini, Julius Berner, Boris Bonev, Thorsten Kurth, Kamyar Azizzadenesheli, and Anima Anandkumar. Neural operators with localized integral and differential kernels. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=vl9GB3fbht.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

M. Mimura. Reaction–diffusion systems arising in biological and chemical systems: application of singular limit procedures. In *Mathematical Aspects of Evolving Interfaces (Funchal, 2000)*, volume 1812 of *Lecture Notes in Mathematics*, pp. 89–121. Springer, Berlin, 2003.

K.W. Morton. *Numerical Solution of Convection-Diffusion Problems*, volume 12 of *Applied Mathematics and Mathematical Computation*. Chapman & Hall, London, 1996.

Ali Nekoozadeh, Mohammad Reza Ahmadzadeh, and Zahra Mardani. Multiscale attention via wavelet neural operators for vision transformers. *arXiv preprint arXiv:2303.12398*, 2023.

Maziar Raissi, Paris Perdikaris, and George 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, 2019.

B. Raonic, R. Molinaro, T. Rohner, S. Mishra, and E. de Bezenac. Convolutional neural operators. *arXiv preprint arXiv:2302.01178*, 2023. URL https://arxiv.org/abs/2302.01178.

Ehsan Roohi and Alireza Mahdavi. Shock-aware physics-guided fusion-deeponet operator for rarefied micro-nozzle flows. *arXiv preprint arXiv:2510.17887*, 2025.

Hans-Görg Roos, Martin Stynes, and Lutz Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems*, volume 24. Springer Science & Business Media, Berlin, 2008.

Hermann Schlichting and Klaus Gersten. *Boundary-Layer Theory*. Springer, Berlin, 2016.

Shagi-Di Shih and R Bruce Kellogg. Asymptotic analysis of a singular perturbation problem. *SIAM Journal on Mathematical Analysis*, 18(5):1467–1511, 1987. doi: 10.1137/0518107.

Yin-Tzer Shih, R. Bruce Kellogg, and Young-Hoon Chang. Characteristic tailored finite point method for convection-dominated convection-diffusion-reaction problems. *Journal of Scientific Computing*, 47(2):198–215, May 2011. ISSN 1573-7691. doi: 10.1007/s10915-010-9433-7. URL https://doi.org/10.1007/s10915-010-9433-7.

Martin Stynes. Steady-state convection-diffusion problems. *Acta Numerica*, 14:445–508, 2005. doi: 10.1017/S0962492904000261.

J.A. Trangenstein and J.B. Bell. Mathematical Structure of the Black-Oil Model for Petroleum Reservoir Simulation. *SIAM Journal on Applied Mathematics*, 49(3):749–783, 1989. doi: 10.1137/0149044. URL http://dx.doi.org/10.1137/0149044.

T. Tripura and S. Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023. ISSN 0045-7825.

Yue Wang, Yonghai Li, and Xiangyun Meng. An Upwind Finite Volume Element Method on a Shishkin Mesh for Singularly Perturbed Convection–Diffusion Problems. *Journal of Computational and Applied Mathematics*, 438:115493, 2024. ISSN 0377-0427. doi: 10.1016/j.cam.2023.115493. URL https://doi.org/10.1016/j.cam.2023.115493.

Zhen Xu, Jiacheng Liu, Kai Chen, Yuan Chen, Zhen Hu, and Bingbing Ni. Amr-transformer: Enabling efficient long-range interaction for complex neural fluid simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5804–5813, 2025.

Yanshun Zhao, Xiaoyu Peng, Jiamin Jiang, Congcong Zhu, and Jingrun Chen. MeshTok: Efficient multi-scale tokenization for scalable PDE transformers. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

A PDE SETUP

We first consider a convection-diffusion equation with homogeneous Dirichlet boundary conditions in the domain $\Omega = [0,1]^2$, given by

$$\begin{cases} -\varepsilon \Delta u_\varepsilon + b(\mathbf{x}) \cdot \nabla u_\varepsilon & = f(\mathbf{x}) \quad \text{in } \Omega \\ u_\varepsilon & = 0 \quad \text{on } \partial\Omega \end{cases} \tag{21}$$

where $b(\mathbf{x}) = (b_1(\mathbf{x}), b_2(\mathbf{x}))$ is a given convection vector field, and f is a source function. One of the key characteristics of singularly perturbed convection-diffusion equations is their sensitivity to the convection field $b(\mathbf{x}) = (b_1(\mathbf{x}), b_2(\mathbf{x}))$ when the perturbation parameter ε is small. In such cases, even slight variations in $b(\mathbf{x})$ can cause significant changes in the structure of the solution, particularly in the location and shape of the layer parts. This behavior is evident in the formation of boundary and corner layers, as illustrated in Fig. 1. Depending on the sign and structure of the components of $b(\mathbf{x})$, the problem can be categorized into three distinct types: boundary layer problems, corner layer problems, and interior layer problems.

Table 5: Summary of the functional forms of the corrector function ϕ for various problem types (boundary, interior, corner, and nonlinear reaction-diffusion layers) determined by the signs of the convection coefficients b_1 and b_2.

Problem	$\mathbf{b}_1(\mathbf{x})$	$\mathbf{b}_2(\mathbf{x})$	$\phi(\xi)$
Boundary Layer	$+$	0	$e^{-b_1(1,y)(1-x)/\varepsilon}$
	$-$	0	$e^{-b_1(0,y)(x)/\varepsilon}$
	0	$+$	$e^{-b_2(x,1)(1-y)/\varepsilon}$
	0	$-$	$e^{-b_2(x,0)(y)/\varepsilon}$
Interior Layer	$(x-x_0)$	0	$\operatorname{erf}\left(x\sqrt{\frac{b_1'(x_0)}{2\varepsilon}}\right)$
Corner Layer	$+$	$+$	$e^{-b_1(1,y)(1-x)/\varepsilon}, e^{-b_2(x,1)(1-y)/\varepsilon}, e^{-(b_1(1,y)(1-x)+b_2(x,1)(1-y))/\varepsilon}$
	$+$	$-$	$e^{-b_1(0,y)x/\varepsilon}, e^{-b_2(x,1)(1-y)/\varepsilon}, e^{-(b_1(0,y)x+b_2(x,1)(1-y))/\varepsilon}$
	$-$	$+$	$e^{-b_1(1,y)(1-x)/\varepsilon}, e^{-b_2(x,0)y/\varepsilon}, e^{-(b_1(1,y)(1-x)+b_2(x,0)y)/\varepsilon}$
	$-$	$-$	$e^{-b_1(0,y)x/\varepsilon}, e^{-b_2(x,0)y/\varepsilon}, e^{-(b_1(0,y)x+b_2(x,0)y)/\varepsilon}$
Nonlinear Reaction-Diffusion	0	0	$e^{-x/\sqrt{\varepsilon}}, e^{-(1-x)/\sqrt{\varepsilon}}$

A.0.1 BOUNDARY LAYER PROBLEM

We consider a boundary layer problem characterized by mixed boundary conditions, consisting of Dirichlet and periodic conditions on different parts of the boundary. Specifically, let the domain Ω be defined such that $\partial\Omega_{dirichlet}$ represents the boundary segments where the Dirichlet condition is imposed, while $\partial\Omega_{periodic}$ corresponds to the periodic boundary. For instance, one may set $\partial\Omega_{dirichlet}$ at $x = 0$ and $x = 1$, with $\partial\Omega_{periodic}$ at $y = 0$ and $y = 1$, or vice versa. The governing equation for the solution u_ε is given by:

$$\begin{cases} -\varepsilon \Delta u_\varepsilon + b(\mathbf{x}) \cdot \nabla u_\varepsilon & = f(\mathbf{x}) \quad \text{in } \Omega \\ u_\varepsilon & = 0 \quad \text{on } \partial\Omega_{dirichlet} \\ u_\varepsilon \text{ is periodic on } \partial\Omega_{periodic}. \end{cases} \tag{22}$$

In this case, one component of the drift vector $b(\mathbf{x}) = (b_1(\mathbf{x}), b_2(\mathbf{x}))^T$ is identically zero, while the other is non-zero. Depending on the sign of the non-zero component $b_i(\mathbf{x})$, we classify the boundary layer problems into the following four types, whose representative solution profiles are illustrated in Figure 1:

- **(Type I)** $b_1(\mathbf{x}) \equiv 1, b_2(\mathbf{x}) \equiv 0$: An **outflow boundary layer** forms at $x = 1$. In this configuration, $\partial\Omega_{dirichlet}$ is set at $x = 0, 1$, and $\partial\Omega_{periodic}$ is at $y = 0, 1$.
- **(Type II)** $b_1(\mathbf{x}) \equiv -1, b_2(\mathbf{x}) \equiv 0$: The boundary layer forms at the left boundary $x = 0$, with the same boundary partition as Type I.
- **(Type III)** $b_1(\mathbf{x}) \equiv 0, b_2(\mathbf{x}) \equiv 1$: An outflow boundary layer forms at $y = 1$. Here, $\partial\Omega_{dirichlet}$ is defined at $y = 0, 1$, and $\partial\Omega_{periodic}$ at $x = 0, 1$.
- **(Type IV)** $b_1(\mathbf{x}) \equiv 0, b_2(\mathbf{x}) \equiv -1$: The boundary layer forms at the bottom boundary $y = 0$, with the same boundary partition as Type III.

Next, we consider the boundary layer problem described in Example 6 of Stynes (2005), which exhibits two characteristic boundary layers. The parameters are set to $\varepsilon = 0.01$ and $\Omega = [0, 1]^2$.

- **(Type V)** $b_1(\mathbf{x}) \equiv 1, b_2(\mathbf{x}) \equiv 0$: Unlike Type I, a **Dirichlet boundary condition** is applied to the entire boundary $\partial\Omega$.

A.0.2 CORNER LAYER PROBLEM

In the case of the Corner Layer problem, we consider the following equation

$$\begin{cases} -\varepsilon\Delta u_\varepsilon + b(\mathbf{x}) \cdot \nabla u_\varepsilon & = f(\mathbf{x}) \quad \text{in } \Omega \\ u_\varepsilon & = 0 \quad \text{on } \partial\Omega \end{cases}$$

with purely homogeneous Dirichlet boundary conditions imposed on the entire boundary $\partial\Omega$. Specifically, we set $u_\varepsilon = 0$ on $\partial\Omega$, where the interaction of boundary layers at the corners of the domain creates complex singular behaviors. Both components $b_1(\mathbf{x})$ and $b_2(\mathbf{x})$ are nonzero and maintain a constant sign throughout the domain. Depending on the sign of the non-zero component $b_i(\mathbf{x})$, we classify the corner layer problems into the following four types:

- **(Type I)** $b_1(\mathbf{x}) \equiv 1, b_2(\mathbf{x}) \equiv -1$: The corner layer forms at $(1, 0)$ (intersection of $x = 1$ and $y = 0$).
- **(Type II)** $b_1(\mathbf{x}) \equiv 1, b_2(\mathbf{x}) \equiv 1$: The corner layer forms at $(1, 1)$ (intersection of $x = 1$ and $y = 1$).
- **(Type III)** $b_1(\mathbf{x}) \equiv -1, b_2(\mathbf{x}) \equiv 1$: The corner layer forms at $(0, 1)$ (intersection of $x = 0$ and $y = 1$).
- **(Type IV)** $b_1(\mathbf{x}) \equiv -1, b_2(\mathbf{x}) \equiv -1$: The corner layer forms at $(0, 0)$ (intersection of $x = 0$ and $y = 0$).

A.0.3 INTERIOR LAYER PROBLEM

Similar to the boundary layer problem, the Interior Layer problem involves a combination of periodic and Dirichlet boundary conditions along two directions. In this paper, we specifically consider the following formulation where the Dirichlet condition is imposed on the x-boundaries and the periodic condition is applied to the y-boundaries:

$$\begin{cases} -\varepsilon\Delta u_\varepsilon + b(\mathbf{x}) \cdot \nabla u_\varepsilon & = f(\mathbf{x}) \quad \text{in } \Omega \\ u_\varepsilon & = 0 \quad \text{on } x = 0 = 1 \\ u_\varepsilon \text{ is periodic on } y = 0 = 1. \end{cases} \tag{23}$$

At least one component of $b(\mathbf{x})$, either $b_1(\mathbf{x})$ or $b_2(\mathbf{x})$, changes sign within the domain. In this paper, we test three different cases

- (Type I) $b_1(\mathbf{x}) = x - 0.2, b_2(\mathbf{x}) \equiv 0$: interior layer at $x = 0.2$
- (Type II) $b_1(\mathbf{x}) = x - 0.5, b_2(\mathbf{x}) \equiv 0$: interior layer at $x = 0.5$
- (Type III) $b_1(\mathbf{x}) = x - 0.8, b_2(\mathbf{x}) \equiv 0$: interior layer at $x = 0.8$

A.0.4 NONLINEAR REACTION-DIFFUSION PROBLEM

We also consider a nonlinear reaction-diffusion problem Ducrot & Langlais (2012); Gaucel & Langlais (2007); Mimura (2003); Kinderlehrer & Stampacchia (2000) with homogeneous Dirichlet boundary conditions in the domain $\Omega = [0, 1]^2$, given by

$$\begin{cases} -\varepsilon\Delta u_\varepsilon - u_\varepsilon + u_\varepsilon^3 & = f(\mathbf{x}) \quad \text{in } \Omega \\ u_\varepsilon & = 0 \quad \text{on } x = 1, x = 0 \end{cases} \tag{24}$$

where f is a source function. In this paper, we choose $g(u) = u^3$. Although the problem equation 24 is nonlinear, the associated boundary layer behaves linearly. By applying the method of matched asymptotic expansions, it can be shown that a boundary layer of thickness ε forms near the outflow boundary at $x = 0$, consistent with the behavior observed in standard boundary layer problems.

A.0.5 HELMHOLTZ PROBLEM

Asymptotic analysis for the Helmholtz scattering problem with the Sommerfeld radiation condition has been extensively studied in the literature; see, for example, Andronov et al. (2005); Bouche et al. (2017). However, in this paper, we focus on a simpler model problem to examine whether our proposed method remains effective even when the solution exhibits complex structures and when asymptotic information is not explicitly known. Our goal is to test the robustness of the method in more general scenarios where standard asymptotic techniques may not be readily applicable.

Specifically, we consider the following Helmholtz equation posed on the domain $\Omega = [0,1]^2$:

$$\begin{cases} -\Delta u - \omega^2 u = f & \text{in } \Omega, \\ \nabla u \cdot \mathbf{n} + i\omega u = 0 & \text{on } \partial\Omega, \end{cases} \tag{25}$$

where $\omega > 0$ is the wave number, f represents a harmonic source, and $i = \sqrt{-1}$ is the imaginary unit. In our framework, we use only the real part of the complex-valued solution u for training and prediction. This simplification is commonly adopted in learning-based methods for Helmholtz-type problems to reduce computational complexity.

For numerical experiments, we investigate the problem in the frequency regime with $\omega^2 = 225$.

A.0.6 BOUNDARY LAYER PROBLEM ON A CIRCLE

Finally, we consider the singularly perturbed problems of the form

$$\begin{cases} -\varepsilon \Delta u_\varepsilon + b(\mathbf{x}) \cdot \nabla u_\varepsilon & = f(\mathbf{x}) \quad \text{in } \Omega \\ u_\varepsilon & = 0 \quad \text{on } \partial\Omega. \end{cases}$$

where Ω is the unit disc centered at (0,0), and the function f is as smooth as needed. In the study of singularly perturbed convection-diffusion problems, it has been established that a boundary layer develops along the lower semicircular boundary Jung & Temam (2011; 2012). The regularity of the leading-order solution u_0 at the characteristic points $(\pm 1, 0)$ is strictly governed by the compatibility of the source term $f(x, y)$. Specifically, if f does not vanish at these points, it becomes incompatible with the reduced equation, leading to the emergence of singularities in the derivatives of u_0. Such singular behaviors at the characteristic points necessitate the satisfaction of specific compatibility conditions to ensure the smoothness of the asymptotic expansion. Therefore, in this paper, we ensure that the source term f satisfies the compatibility condition by requiring it to vanish at the characteristic points $(\pm 1, 0)$.

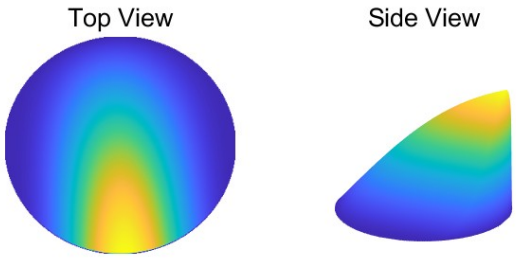


Figure 4: Solution profile for the boundary layer problem on circle

The corrector function of this problem is given by

$$\theta^0(\eta, \xi) = -u^0(\cos\eta, \sin\eta) \exp\left(\frac{\sin\eta}{\varepsilon}\xi\right) \chi_{[\pi, 2\pi]}(\eta),$$

where $\chi_A(\eta)$ is the characteristic function of A, and r is the distance to the origin and η is the polar angle.

A.1 Fourier Neural Operator

Let $\Omega \subset \mathbb{R}^d$ be a bounded open domain, and let $\mathcal{A} = \mathcal{A}(\Omega; \mathbb{R}^{d_a})$ and $\mathcal{U} = \mathcal{U}(\Omega; \mathbb{R}^{d_u})$ denote separable Banach spaces of input and output functions, respectively. The Fourier Neural Operator defines a parameterized mapping

$$\mathcal{N}_\Theta : \mathcal{A} \to \mathcal{U}. \tag{26}$$

Given an input $a \in \mathcal{A}$, the FNO is expressed as

$$\mathcal{N}_\Theta(a) = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}(a), \tag{27}$$

where $\mathcal{R}$ denotes a lifting operator, $\mathcal{Q}$ denotes a projection operator, and $\mathcal{L}_l$ is the l-th Fourier layer Li et al. (2021). Each layer takes the form

$$\mathcal{L}_l(v)(\mathbf{x}) = \sigma\left(W_l v(\mathbf{x}) + b_l(\mathbf{x}) + (\mathcal{K}_{\theta_l} v)(\mathbf{x})\right), \tag{28}$$

where W_l is a pointwise linear transformation, b_l is a bias term, and σ is a nonlinear activation function. The Fourier kernel operator is defined by

$$(\mathcal{K}_{\theta_l} v)(\mathbf{x}) = \mathcal{F}^{-1}\left(P_{\theta_l}(\mathbf{k}) \cdot \mathcal{F}(v)(\mathbf{k})\right)(\mathbf{x}), \tag{29}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively, and P_{θ_l} contains learnable transformations of the retained Fourier modes. In LibCorrNet, Fourier representations are used in separate branches to model the global solution component and the coordinate representation used by the candidate-based corrector module.

A.2 Deep Convolutional Neural Networks

The deep CNNs considered in this paper have two essential ingredients: a rectified linear unit (ReLU) defined as a univariate nonlinear function σ given by

$$\sigma(u) = (u)_+ = \max\{u, 0\}, \quad u \in \mathbb{R} \tag{30}$$

and a sequence of convolutional filter masks $\mathbf{w} = \{w^{(j)}\}_j$ inducing sparse convolutional structures. Here a filter mask $w = (w_k)_{k=-\infty}^{\infty}$ means a sequence of filter coefficients. We use a fixed integer filter length $\hat{s} \geq 2$ to control the sparsity, and assume that $w_k^{(j)} \neq 0$ only for $0 \leq k \leq \hat{s}$. The convolution of such a filter mask w with another sequence $v = (v_0, \ldots, v_{D-1})$ is a sequence $w * v$ given by $(w * v)_i = \sum_{k=0}^{D-1} w_{i-k} v_k$. This leads to a $(D + \hat{s}) \times D$ Toeplitz type convolutional matrix T which has constant diagonals. Starting with the action of $T^{(1)}$ on the input data vector $x \in \mathbb{R}^d$, we can define a deep CNN of depth J as a sequence of J vectors $h^{(j)}(x)$ of functions on $\mathbb{R}^d$ given iteratively by $h^{(0)}(x) = x$ and

$$h^{(j)}(x) = \sigma\left(T^{(j)} h^{(j-1)}(x) - b^{(j)}\right), \quad j = 1, 2, \ldots, J, \tag{31}$$

where $T^{(j)} = \left(w_{i-k}^{(j)}\right)$ is a $d_j \times d_{j-1}$ convolutional matrix, σ acts on vectors componentwise, and $\mathbf{b}$ is a sequence of bias vectors $b^{(j)}$.

B Experimental Settings

Each source function $f(\mathbf{x})$ is constructed as a linear combination of $n_g = 200$ radial basis functions $\{\phi_i(\mathbf{x})\}_{i=1}^{n_g}$, where each $\phi_i(\mathbf{x})$ is a two-dimensional isotropic Gaussian function centered at position $\mathbf{x}_i \in [0, 1]^2$. Specifically, we define:

$$f(x) = \sum_{i=1}^{n_g} p_i \, \phi(\mathbf{x} - \mathbf{x}_i). \tag{32}$$

Here, $\phi(\mathbf{x})$ is a Gaussian function with standard deviation $\sigma = 0.05$, which is small relative to the domain size and chosen to induce high spatial variability. The Gaussian centers $\mathbf{x}_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$ are sampled independently and uniformly from the unit square $[0, 1]^2$. When the boundary condition is

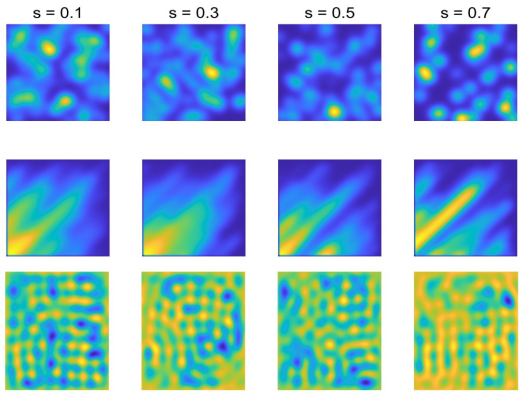


Figure 5: Visualization of the solution profiles corresponding to different sparsity levels s. The top row illustrates the source functions $f(\mathbf{x})$ constructed from Gaussian radial basis functions, while the bottom row shows the corresponding PDE solutions for a corner layer problem with $\varepsilon = 1e-05$ and Helmholtz equation with $\omega^2 = 3600$. Increasing sparsity leads to more localized source functions, which significantly affect the solution structures.

imposed on $x = 0$ and $x = 1$, we use a modified basis function that varies only in the x-direction. Similarly, if the boundary condition is applied on $y = 0$ and $y = 1$, the basis function is taken to vary only in the y-direction. To ensure consistency across samples and control the overall magnitude of the source input, the function $f(\mathbf{x})$ is normalized so that its maximum value over the domain is equal to 5. The amplitude parameters $\{p_i\}$ are initially sampled from a uniform distribution $U(0, 1)$, and sparsity is introduced by applying a random binary mask. That is,

$$p_i = \begin{cases} \text{sampled from } U(0,1), & \text{with probability } 1-s \\ 0, & \text{with probability } s. \end{cases} \tag{33}$$

The sparsity level s controls the fraction of zero entries in $p = \{p_i\}_{i=1}^{n_g}$, and is varied across different configurations: specifically, $s = 0.1, 0.3, 0.5, 0.7$, corresponding to increasing levels of sparsity.
While each source function $f(\mathbf{x})$ is constructed as a linear combination of Gaussian radial basis functions to induce spatial variability, we impose additional constraints for problems involving boundary layers (Types I–IV), interior layers, nonlinear reaction-diffusion equations, and circular domains. According to the analytical frameworks established in prior studies Jung & Temam (2011; 2012); Hong et al. (2014); Gie et al. (2024; 2018), if f fails to satisfy the necessary compatibility conditions, the derivatives of the leading-order solution u_0 exhibit singular behaviors or blow up at characteristic points. Such singularities prevent the formation of a smooth asymptotic expansion and can lead to significant numerical instabilities during the training process. To ensure compatibility and numerical stability, we modify the source functions as follows:

- **Boundary Layer (Types I, II), Nonlinear Reaction-Diffusion, and Time-Dependent Problems**: Instead of using two-dimensional isotropic Gaussians, we employ basis functions that vary only in the x-direction:

$$\phi_i(x, y) = \exp\left(-\frac{(x - x_i)^2}{2\sigma^2}\right).$$

- **Boundary Layer (Types III, IV)**: Similarly, for these cases, we utilize basis functions that vary exclusively in the y-direction:

$$\phi_i(x, y) = \exp\left(-\frac{(y - y_i)^2}{2\sigma^2}\right).$$

- **Interior Layer Problem**: To ensure the required transition across the internal layer, we multiply the original source function $f(\mathbf{x})$ by a shifting factor:

$$\tilde{f}(\mathbf{x}) = (x - x_0) f(\mathbf{x}).$$

- **Boundary Layer on a Circle**: To satisfy the compatibility conditions at the characteristic points $(\pm 1, 0)$, we weight the source function by a factor that vanishes at these boundaries:

$$\tilde{f}(\mathbf{x}) = (1 - x^2)^2 f(\mathbf{x}).$$

B.1 Model Parameters

In all experiments, we use the relative L^2 error as the loss function for both FNO and LibCorrNet. Unless stated otherwise, FNO and LibCorrNet are trained under identical experimental conditions across all types of PDE problems with mode $=$ width $= 10$. The optimizer is AdamW with an initial learning rate of 1×10^{-2} and weight decay of 1×10^{-7}. To enhance training efficiency, a ReduceLROnPlateau scheduler reduces the learning rate by a factor of 0.5 if the validation loss does not improve for 10 consecutive epochs. Early stopping is applied with a patience of 150 epochs to prevent overfitting. The numerical experiments were conducted on a single NVIDIA RTX 5070 Ti GPU with 12GB of VRAM. The computational setup included an Intel Xeon® E5-2620 v3 CPU.

FNO and LibCorrNet are trained for 1300 epochs with a batch size of 24, and the GELU activation function is consistently applied throughout all layers. The number of Fourier layers varies depending on the model architecture: the FNO model incorporates 6 Fourier layers, the LibCorrNet model uses 4 Fourier layers (3 for the smooth part and 1 for spatial variable ξ), and the latent encoder consists of two convolutional layers.

Table 6: Summary of input variables spatial coordinate (ξ) and corrector functions (ϕ) for FNO, ComFNO and LibCorrNet.

Model	Manual Selection	Input
FNO	$\times$	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$
ComFNO	✓	$(\mathbf{x}, f(\mathbf{x}), f(\xi))$
LibCorrNet	$\times$	$(\mathbf{x}, f(\mathbf{x}), b(\mathbf{x}), \varepsilon)$

For ComFNO, the chosen optimizer for all minimization problems is Adam, accompanied by the consistent utilization of the GELU activation function. Further details concerning the remaining parameters pivotal for our result generation can be found in Table 7. For both CNO and GNOT models,

blockNum	**LR**	**epoch**	**batch size**
2	0.001	1000	20

Table 7: Experimental parameters for ComFNO investigations. The term "blockNum" denotes the quantity of layer blocks implemented within the architecture. "LR" designates the learning rate employed, while "epoch" signifies the count of training iterations performed.

we adopted the optimal hyperparameters provided in their respective official GitHub implementations to ensure a fair comparison and reproducibility. "'latex

B.2 Computational Cost

Table 13 reports the wall-clock training time and per-sample inference latency of the standard FNO and LibCorrNet at two training budgets, $N_T = 256$ and $N_T = 1024$, measured on the same hardware described in Appendix B.1.

Despite incorporating an additional coordinate-learning branch and a candidate-weighting module, LibCorrNet requires less wall-clock training time than the standard FNO at both training budgets. Specifically, the training time decreases from 185.49 s to 127.11 s for $N_T = 256$ and from 1665.83 s to 674.91 s for $N_T = 1024$. Under the same early-stopping criterion, LibCorrNet reaches its validation-loss plateau in fewer epochs. One possible explanation is that the candidate-based structural prior reduces the optimization burden on the neural backbone: localized layer structures are partially represented by the prescribed candidate library rather than being learned entirely through the Fourier layers. Consequently, the reduction in the number of training epochs can offset the additional per-epoch cost introduced by the coordinate-learning branch.

Table 8: Comparison of training and inference times for FNO and LibCorrNet at $N_T = 256$ and $N_T = 1024$.

N_T	Model	Training Time (s)	Inference Time / Sample (ms)
256	FNO	185.49	4.79
	LibCorrNet (Ours)	**127.11**	6.15
1024	FNO	1665.83	4.71
	LibCorrNet (Ours)	**674.91**	6.12

At inference time, LibCorrNet introduces a modest overhead of approximately 1.3–1.4 ms per sample compared with FNO. This increase is consistent with the additional forward computations required by the coordinate-learning and candidate-weighting modules. The inference overhead remains nearly unchanged between $N_T = 256$ and $N_T = 1024$, indicating that it primarily reflects the fixed architectural cost of the corrector branch rather than the size of the training set. Overall, LibCorrNet achieves the accuracy improvements reported in Table 1 while maintaining competitive computational efficiency, with reduced total training time and only a modest increase in per-sample inference latency.

C Supplementary Experimental Results

C.1 Statistical Analysis over Independent Runs

To assess the robustness of LibCorrNet to random initialization and training-data sampling, we report statistics over the 10 independent runs used to obtain the results in Table 1. For each PDE benchmark and training-set size, the model is trained independently using different random seeds and evaluated on 1,024 held-out test instances. Table 9 reports the mean and standard deviation of the relative L^2 test error across the 10 runs.

Table 9: Mean and standard deviation of the relative L^2 test errors of LibCorrNet over 10 independent runs. Each run is evaluated on 1,024 held-out test instances.

PDE	Statistic	$N_T = 128$	$N_T = 256$	$N_T = 512$	$N_T = 1024$
Boundary Layer	Mean	5.06×10^{-3}	3.09×10^{-3}	2.58×10^{-3}	2.10×10^{-3}
	Std. Dev.	5.0×10^{-4}	3.0×10^{-4}	2.0×10^{-4}	2.0×10^{-4}
Corner Layer	Mean	1.99×10^{-2}	1.17×10^{-2}	8.75×10^{-3}	6.51×10^{-3}
	Std. Dev.	7.0×10^{-3}	5.0×10^{-3}	4.0×10^{-3}	3.0×10^{-3}
Interior Layer	Mean	8.81×10^{-3}	3.48×10^{-3}	3.73×10^{-3}	2.59×10^{-3}
	Std. Dev.	3.0×10^{-3}	1.0×10^{-3}	5.0×10^{-4}	5.0×10^{-4}
Nonlinear RD	Mean	2.95×10^{-3}	2.02×10^{-3}	1.82×10^{-3}	1.15×10^{-3}
	Std. Dev.	3.0×10^{-4}	3.0×10^{-4}	3.0×10^{-4}	3.0×10^{-4}

The mean errors generally decrease as the training-set size increases. For the boundary layer problem, the mean relative L^2 error decreases from 5.06×10^{-3} at $N_T = 128$ to 2.10×10^{-3} at $N_T = 1024$. Similar reductions are observed for the corner layer and nonlinear reaction–diffusion problems, where the errors decrease from 1.99×10^{-2} to 6.51×10^{-3} and from 2.95×10^{-3} to 1.15×10^{-3}, respectively. The interior layer problem also shows an overall reduction, with a small non-monotonic variation between $N_T = 256$ and $N_T = 512$.

The standard deviations provide additional information on the run-to-run variability. For the boundary layer problem, the standard deviation decreases from 5.0×10^{-4} to 2.0×10^{-4} as the training-set size increases. The corner and interior layer problems exhibit larger variability, particularly in the lower-data regime, while the nonlinear reaction–diffusion problem shows relatively small variations across the considered training-set sizes. Overall, these statistics complement the averaged results in Table 1 by quantifying the variability across independent training runs.

C.2 Qualitative Error Analysis

Figures 10-14 provide qualitative comparisons for the five PDE benchmarks considered in the main experiments. For each problem, we visualize the reference solution and the absolute prediction error for a representative test instance using a model trained with $N_T = 128$. We additionally report the absolute error aggregated over all 1,024 held-out test instances.

These visualizations complement the quantitative results in Table 1 by showing where the remaining approximation errors are spatially concentrated. In particular, they allow us to examine the behavior of LibCorrNet near localized structures such as boundary, corner, and interior layers, as well as in the nonlinear reaction-diffusion settings. Together with the quantitative results, these examples provide additional evidence that the learned corrector representation can approximate localized solution structures across the considered PDE configurations.

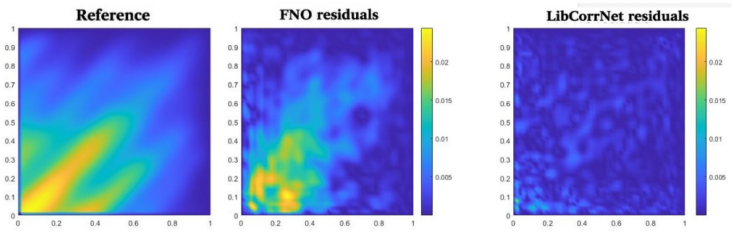


Figure 6: Numerical results for the corner layer problem with $b(\mathbf{x}) = (-1, 1)$ and $\varepsilon = 10^{-5}$. The panels display, from left to right: the reference solution for a representative test case, the absolute residuals of the standard FNO, and the absolute residuals of the proposed LibCorrNet. The residual plots represent the average error distributions evaluated over 1,024 randomly selected test samples. The models were trained on a $128 \times 52 \times 52$ dataset, and the absolute residuals are defined as $|u_{ref} - u_{pred}|$.

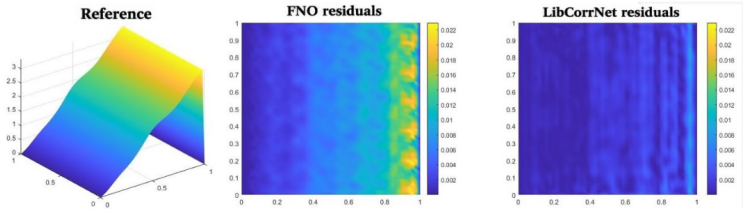


Figure 7: Numerical results for the boundary layer problem with $b(\mathbf{x}) = (1, 0)$ and $\varepsilon = 10^{-5}$. The panels display, from left to right: the reference solution for a representative test case, the absolute residuals of the standard FNO, and the absolute residuals of the proposed LibCorrNet. The residual plots represent the average error distributions evaluated over 1,024 randomly selected test samples. The models were trained on a $128 \times 52 \times 52$ dataset, and the absolute residuals are defined as $|u_{ref} - u_{pred}|$.

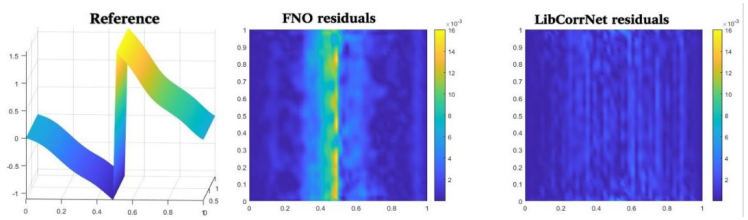


Figure 8: Numerical results for the interior layer problem with $b(\mathbf{x}) = (x - 0.5, 0)$ and $\varepsilon = 10^{-5}$. The panels display, from left to right: the reference solution for a representative test case, the absolute residuals of the standard FNO, and the absolute residuals of the proposed LibCorrNet. The residual plots represent the average error distributions evaluated over 1,024 randomly selected test samples. The models were trained on a $128 \times 52 \times 52$ dataset, and the absolute residuals are defined as $|u_{ref} - u_{pred}|$.

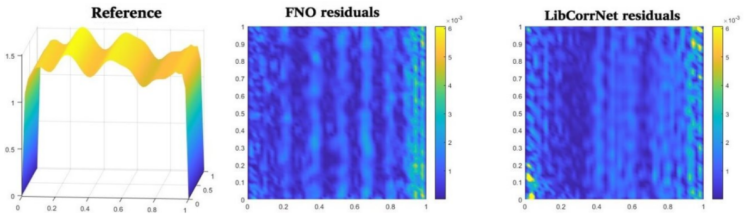


Figure 9: Numerical results for the nonlinear reaction-diffusion layer problem with $b(\mathbf{x}) = (0, 0)$ and $\varepsilon = 10^{-5}$. The panels display, from left to right: the reference solution for a representative test case, the absolute residuals of the standard FNO, and the absolute residuals of the proposed LibCorrNet. The residual plots represent the average error distributions evaluated over 1,024 randomly selected test samples. The models were trained on a $128 \times 52 \times 52$ dataset, and the absolute residuals are defined as $|u_{ref} - u_{pred}|$.

C.3 Extension to SPPDEs on Circular Domain

In this section, we evaluate the performance of our proposed model on a circular domain as introduced in Section C.3. To adapt these functions to the circular geometry, the values outside the unit disk were padded with zeros, effectively restricting the source input to the domain of interest. This result underscores the robustness of LibCorrNet in handling non-rectangular domains and its capability to resolve localized boundary layer structures more effectively than conventional operator learning architectures.

As shown in Table 10, the Library-based Corrector Network (LibCorrNet) demonstrates superior accuracy compared to the standard FNO. Specifically, for a singular perturbation parameter of $\epsilon = 1 \times 10^{-3}$, LibCorrNet achieves a relative L^2 error of 3.19×10^{-2}, whereas the vanilla FNO yields a higher error of 5.23×10^{-2}.

Table 10: Relative (L^2) error on a circular domain with $\epsilon = 1 \times 10^{-3}$

Model	FNO	LibCorrNet
rel L^2 error	$5.23e^{-2}$	$3.19e^{-2}$

C.4 Ablation Study on Candidate Library Size and Selection

We investigate the effect of the candidate library size N_c and the training-set size N_T on operator-learning performance across four singularly perturbed PDE benchmarks: the boundary layer, corner layer, interior layer, and nonlinear reaction–diffusion problems.

We construct a candidate pool $\mathcal{D}_{\text{pool}}$ containing 20 representative basis functions defined on $[0,1] \times [0,1]$. The pool consists of six polynomial functions and 14 non-polynomial functions representing exponential decay, error-function transitions, hyperbolic transitions, and localized profiles:

- **Polynomial functions (6):**

$$\{x,\ y,\ x(1-x),\ y(1-y),\ xy(1-x)(1-y),\ (x-0.5)^2+(y-0.5)^2\}. \tag{34}$$

- **Non-polynomial functions (14):**

$$\Big\{e^{-3x},\ e^{-3(1-x)},\ e^{-3y},\ e^{-3(1-y)},\ e^{-4((x-0.5)^2+(y-0.5)^2)},\\ \operatorname{erf}(3x),\ \operatorname{erf}(3(1-x)),\ \operatorname{erf}(3(x-0.5)),\ \operatorname{erf}(2x)\operatorname{erf}(2y),\\ \tanh(3(x-0.5)),\ \tanh(3(y-0.5)),\ \tanh(3(x-y)),\\ \operatorname{sech}(4(x-0.5)),\ \operatorname{sech}(4(y-0.5))\Big\}. \tag{35}$$

For each library size $N_c \in \{2,4,6\}$, we randomly sample N_c functions from $\mathcal{D}_{\text{pool}}$ using independent random seeds and report the averaged relative L^2 test errors. We compare these configurations with two reference settings: $N_c = 0$, corresponding to the standard FNO without a candidate corrector library, and Exact ϕ, which uses a prescribed analytical corrector profile.

Table 11: Average relative L^2 test errors for different candidate library sizes N_c and training-set sizes N_T across four singularly perturbed PDE benchmarks. Bold indicates the best result among the candidate-library configurations $N_c \in \{2,4,6\}$.

N_c	Boundary Layer			Corner Layer			Interior Layer			Nonlinear RD		
	$N_T=256$	$N_T=512$	$N_T=1024$	$N_T=256$	$N_T=512$	$N_T=1024$	$N_T=256$	$N_T=512$	$N_T=1024$	$N_T=256$	$N_T=512$	$N_T=1024$
0 (FNO)	$6.90e^{-3}$	$3.30e^{-3}$	$3.50e^{-3}$	$5.90e^{-2}$	$3.05e^{-2}$	$1.85e^{-2}$	$2.75e^{-2}$	$1.30e^{-2}$	$1.09e^{-2}$	$1.55e^{-2}$	$9.80e^{-3}$	$4.80e^{-3}$
2	$3.30e^{-3}$	$2.77e^{-3}$	$2.19e^{-3}$	$1.43e^{-2}$	$\mathbf{1.09e^{-2}}$	$8.40e^{-3}$	$1.06e^{-2}$	$8.60e^{-3}$	$6.70e^{-3}$	$2.17e^{-3}$	$1.96e^{-3}$	$1.55e^{-3}$
4	$3.49e^{-3}$	$2.79e^{-3}$	$2.20e^{-3}$	$\mathbf{1.39e^{-2}}$	$1.10e^{-2}$	$\mathbf{8.17e^{-3}}$	$9.77e^{-3}$	$9.23e^{-3}$	$6.56e^{-3}$	$\mathbf{2.15e^{-3}}$	$1.98e^{-3}$	$\mathbf{1.49e^{-3}}$
6	$\mathbf{3.12e^{-3}}$	$\mathbf{2.59e^{-3}}$	$\mathbf{2.16e^{-3}}$	$1.48e^{-2}$	$\mathbf{1.09e^{-2}}$	$8.37e^{-3}$	$\mathbf{9.54e^{-3}}$	$\mathbf{8.46e^{-3}}$	$\mathbf{6.34e^{-3}}$	$2.21e^{-3}$	$\mathbf{1.95e^{-3}}$	$1.51e^{-3}$
Exact ϕ	$5.00e^{-3}$	$3.00e^{-3}$	$2.70e^{-3}$	$3.60e^{-2}$	$1.90e^{-2}$	$1.30e^{-2}$	$1.60e^{-2}$	$8.50e^{-3}$	$6.50e^{-3}$	$8.90e^{-3}$	$6.00e^{-3}$	$3.10e^{-3}$

As shown in Table 11, introducing the candidate corrector library consistently improves upon the standard FNO baseline across all four PDE benchmarks and training-set sizes. The improvement is particularly pronounced for the corner layer and nonlinear reaction–diffusion problems. For example, at $N_T = 256$, the corner-layer error decreases from 5.90×10^{-2} for FNO to 1.39×10^{-2} with $N_c = 4$, while the nonlinear reaction–diffusion error decreases from 1.55×10^{-2} to 2.15×10^{-3}.

The results also show that increasing N_c does not necessarily lead to monotonic improvement. While $N_c = 6$ performs best for the boundary and interior layer problems, smaller libraries achieve comparable or slightly better performance in several corner-layer and nonlinear reaction–diffusion settings. This suggests that performance depends not only on the number of candidate functions but also on whether the selected functions provide suitable representations of the underlying localized solution structures.

Interestingly, the randomly selected candidate-library configurations also outperform the prescribed Exact ϕ baseline in most settings. For the boundary, corner, and nonlinear reaction–diffusion problems, all candidate-library configurations achieve lower errors than Exact ϕ across the reported training-set sizes. For the interior layer problem, the comparison is closer at larger N_T, but the best candidate-library configuration remains competitive with or slightly better than Exact ϕ. These results suggest that using a flexible set of candidate profiles can be more effective than relying on a single prescribed corrector profile when the relevant localized structure is not fully represented by that profile.

C.5 Robustness against High-Frequency Source Terms

While boundary layer stiffness is the primary challenge in standard SPPDEs, the complexity of the source term can present a more significant bottleneck. For instance, in a source function of the form

$$f = m_1 \sin(Nm_2x + Nm_3y) + m_4 \cos(Nm_5x + Nm_6y),$$

an increase in N causes source-driven oscillations to outweigh the difficulty posed by the layer itself. Our results demonstrate that the model must resolve these dense global fluctuations to accurately

predict the corrector function, marking a transition from stiffness-centered to complexity-centered learning difficulty.

Table 12: Comparative Analysis of FNO, CNO, ComFNO, and LibCorrNet: Relative L_2 Error (Mean). The evaluation is conducted with a highly oscillatory source function f

Layer	Num. t	FNO	CNO	ComFNO	LibCorrNet
Boundary	64	$1.18e^{-1}$	$1.81e^{-1}$	$1.15e^{-1}$	$\mathbf{1.11e^{-1}}$
	128	$8.16e^{-2}$	$8.42e^{-2}$	$6.50e^{-2}$	$\mathbf{5.07e^{-2}}$
	256	$3.94e^{-2}$	$2.20e^{-2}$	$2.10e^{-2}$	$\mathbf{1.96e^{-2}}$
	512	$4.57e^{-2}$	$1.55e^{-2}$	$2.80e^{-2}$	$\mathbf{1.49e^{-2}}$
	1024	$1.06e^{-2}$	$2.08e^{-2}$	$8.50e^{-3}$	$\mathbf{7.10e^{-3}}$
Interior	64	$3.20e^{-1}$	$\mathbf{2.13e^{-1}}$	$2.05e^{-1}$	$2.35e^{-1}$
	128	$1.36e^{-1}$	$1.60e^{-1}$	$8.20e^{-2}$	$\mathbf{4.55e^{-2}}$
	256	$9.40e^{-2}$	$1.05e^{-1}$	$5.40e^{-2}$	$\mathbf{1.65e^{-2}}$
	512	$6.86e^{-2}$	$4.73e^{-2}$	$\mathbf{1.95e^{-2}}$	$2.20e^{-2}$
	1024	$2.41e^{-2}$	$2.76e^{-2}$	$1.20e^{-2}$	$\mathbf{9.60e^{-3}}$
Corner	64	$2.01e^{-1}$	$\mathbf{1.01e^{-1}}$	$1.50e^{-1}$	$1.30e^{-1}$
	128	$1.66e^{-1}$	$\mathbf{8.57e^{-2}}$	$8.80e^{-2}$	$9.22e^{-2}$
	256	$5.92e^{-2}$	$4.93e^{-2}$	$3.20e^{-2}$	$\mathbf{2.69e^{-2}}$
	512	$4.29e^{-2}$	$6.03e^{-2}$	$3.90e^{-2}$	$\mathbf{3.71e^{-2}}$
	1024	$2.38e^{-2}$	$\mathbf{7.80e^{-3}}$	$9.20e^{-3}$	$1.05e^{-2}$

C.6 Computational Cost

Table 13 reports the wall-clock training time and per-sample inference latency of the standard FNO and LibCorrNet at two training budgets, $N_T = 256$ and $N_T = 1024$, measured on the same hardware described in Appendix B.1.

Despite incorporating an additional coordinate-learning branch and a candidate-weighting module, LibCorrNet requires less wall-clock training time than the standard FNO at both training budgets. Specifically, the training time decreases from 185.49 s to 127.11 s for $N_T = 256$ and from 1665.83 s to 674.91 s for $N_T = 1024$. Under the same early-stopping criterion, LibCorrNet reaches its validation-loss plateau in fewer epochs. One possible explanation is that the candidate-based structural prior reduces the optimization burden on the neural backbone: localized layer structures are partially represented by the prescribed candidate library rather than being learned entirely through the Fourier layers. Consequently, the reduction in the number of training epochs can offset the additional per-epoch cost introduced by the coordinate-learning branch.

At inference time, LibCorrNet introduces a modest overhead of approximately 1.3–1.4 ms per sample compared with FNO. This increase is consistent with the additional forward computations required by the coordinate-learning and candidate-weighting modules. The inference overhead remains nearly unchanged between $N_T = 256$ and $N_T = 1024$, indicating that it primarily reflects the fixed architectural cost of the corrector branch rather than the size of the training set. Overall, LibCorrNet achieves the accuracy improvements reported in Table 1 while maintaining competitive computational efficiency, with reduced total training time and only a modest increase in per-sample inference latency.

Table 13: Comparison of training and inference times for FNO and LibCorrNet at $N_T = 256$ and $N_T = 1024$.

N_T	Model	Training Time (s)	Inference Time (ms)
256	FNO	185.49	4.79
	LibCorrNet (Ours)	**127.11**	6.15
1024	FNO	1665.83	4.71
	LibCorrNet (Ours)	**674.91**	6.12