# ON OPTIMIZATION OF POLES FOR ADAPTIVE FOURIER DECOMPOSITION-INSPIRED NEURAL LAYERS

*Zeyuan Song, Zheyu Jiang*

School of Chemical Engineering
Oklahoma State University
Stillwater, Oklahoma USA

## ABSTRACT

Spectral methods and their variants are important numerical approaches for solving PDE problems. However, the performance of these methods deteriorates when solutions contain singularities or sharp gradients. In this work, we integrate Adaptive Fourier Decomposition (AFD) with Blaschke-type bases into a neural operator architecture for solving forward and inverse PDEs. A central challenge in AFD-inspired neural operator is how to select AFD poles, which lie on the unit disk (a Riemannian manifold), and network hyperparameters, which lie in Euclidean space. We address this in an optimization framework, which alternates between Riemannian gradient steps for pole updates and Euclidean steps for network parameters. We prove descent and linear convergence to first-order critical points under standard strong-convexity and cross-block coupling conditions. A greedy outer loop adaptively increases the number of poles and is provably convergent, attaining the global optimum when the target is exactly representable. Experiments on Burgers' (forward problem) and Darcy flow (inverse problem) equations demonstrate improved accuracy and efficiency over Euclidean-only variants and competitive baseline methods, illustrating the need and benefits of manifold-aware pole optimization in AFD-based neural operators.

***Index Terms—*** Partial differential equation, adaptive Fourier decomposition, pole selection, optimization, Riemann manifold

## 1. INTRODUCTION AND MOTIVATION

Accurate and efficient solution methods for partial differential equations (PDEs) benefit a wide range of science and engineering fields [1, 2, 3]. The general form of PDEs can be expressed as

$$\mathcal{L}[u(x,t)] = f(x), \quad x \in \Omega, \tag{1}$$

where $\mathcal{L}$ denotes the differential operator, $f(x)$ is the source term, and $\Omega$ is the spatial domain [4]. Spectral methods are widely used to solve PDEs as they can accurately represent *smooth* solutions. The idea is to decompose the solution $u(x,t)$ using basis functions $\phi_k(x)$

$$u(x,t) = \sum_{k\in\mathbb{N}} \hat{u}_k \phi_k(x), \tag{2}$$

followed by projecting the differential operator $\mathcal{L}$ onto this subspace, which leads to a system of ordinary differential equations for the coefficients $\hat{u}_k(t)$. It has been shown that spectral methods achieve exponential convergence for smooth solutions [5]. However, conventional spectral methods fall short in approximating *non-smooth* solutions having *singularities* or *sharp gradients*, since the basis functions are fixed.

To address this issue, the idea is to turn the fixed basis functions into adaptive ones. One technique, adaptive Fourier decomposition (AFD), was proposed to adaptively decompose signals in Hardy space [6, 7], which can be further relaxed to reproducing kernel Hilbert space (RKHS) [8]. Assuming the basis functions $\mathscr{B}_k$ are Blaschkle-type functions

$$\mathscr{B}_k(z) = \frac{\sqrt{1-|a_k|^2}}{1-\overline{a_k}z} \prod_{j=1}^{k-1} \frac{z-a_j}{1-\overline{a_j}z}, \quad a_k \in \mathbb{D}, \tag{3}$$

where $a_k$ is called the *pole*, and $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$ is the unit disk. With this, the solution $u(x,t)$ can be obtained by:

$$u(x,t) = \sum_{k\in\mathbb{N}} \langle u(x,t), \mathscr{B}_k \rangle \mathscr{B}_k(\exp ix), \tag{4}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product in the corresponding function space, and $z = \exp ix$. The adaptive nature of AFD comes from the adaptability in pole selection. In classic AFD, poles are selected following the *maximal selection principle* (MSP), a computationally expensive procedure. In MSP, the first pole is determined by

$$a_1 = \arg\max_{a\in\mathbb{D}} (1-|a|^2)|u(a)|^2, \tag{5}$$

where $u(a)$ denotes the reconstructed solution $u$ under the poles $a$ in Equation (4). The subsequent poles are recursively selected in the same manner for the *residuals*.

Reformulating AFD into a data-driven, neural operator-based framework enables fast and accurate solutions to PDEs by leveraging the advantages of adaptive bases [9], which existing neural operator PDE solvers based on frequency approaches (e.g., Fourier and wavelet methods [10, 11, 12, 13]) fall short of. Nevertheless, two issues are impeding this endeavor. First, classic AFD can only be used to solve a limited variety of PDEs [14, 15], since the function spaces of many PDEs do not correspond to a reproducing kernel Hilbert space (RKHS). For those PDEs whose solution space is not a Hardy space, one can use neural networks to construct the *nearest* RKHS [16]. In this work, we propose simple yet effective neural layers inspired by the classic AFD theory with Blaschkle-type basis of Equation (3), whose output aligns with the result of Equation (4). The second issue is how to optimize the poles $a_k$ on the $m$-dimensional unit disk $\mathbb{D}^m$ and the hyperparameters $\theta$ of the neural layers in Euclidean space $\mathbb{R}^p$. Specifically, one needs to solve an optimization problem on two different domains, and the unit disk $\mathbb{D}$ is a Riemann surface with the Poincaré metric $ds^2 = \frac{4|dz|^2}{(1-|z|^2)^2}$ [17]. Therefore, this paper provides new perspectives on 1) designing neural architectures with rigorous theoretical justifications and 2) optimizing key parameters in Riemann surfaces and Euclidean spaces.

## 2. AFD-INSPIRED NEURAL LAYERS

AFD-inspired neural layers expand the lifted input function into a set of data-dependent rational orthogonal functions defined on a unit disk. Specifically, the computational grid $\{x_j\}_{j=1}^N \subset \Omega$ is mapped affinely to the complex unit disk $\mathbb{D}^m = \{z \in \mathbb{C}^m : |z| < 1\}$ by $x_j \mapsto z_j$ with radius $\rho < 1$. For a sequence of poles $a_1, \ldots, a_m \in \mathbb{D}$, the bases are constructed following Equation (3). Here, we use the notation $\mathscr{B}_k(z; a_{1:k})$ instead of $\mathscr{B}_k(z)$. Evaluating these basis functions on the grid yields the matrix:

$$\Phi(a) = \begin{bmatrix} \mathscr{B}_1(z_1; a_1) & \cdots & \mathscr{B}_1(z_N; a_1) \\ \vdots & \ddots & \vdots \\ \mathscr{B}_m(z_1; a_{1:m}) & \cdots & \mathscr{B}_m(z_N; a_{1:m}) \end{bmatrix} \in \mathbb{C}^{m \times N}.$$

Once the lifted input $f_{\text{lift}}(x) = \mathcal{L}_\theta([f(x), x]) \in \mathbb{R}^C$ are available, AFD-inspired neural layers compute the coefficients:

$$\langle f_{\text{lift}}, \mathscr{B}_k(\cdot; a_{1:k}) \rangle := f_{\text{lift}}^\top \mathscr{B}_k(\cdot; a_{1:k}), \tag{6}$$

and the output can be reconstructed through adaptive expansion:

$$\hat{u}_{a_{1:k}, \theta}(x) = \sum_{k=1}^m c_k \, \mathscr{B}_k(x; a_{1:k}), \tag{7}$$

which reduces to a compact low-rank factorization on discrete grids:

$$\hat{u}_{a_{1:k}, \theta}(x) = (f_{\text{lift}}^\top \Phi(a)) \, \Phi(a)^\top. \tag{8}$$

Note that when all poles are fixed ($a_k = 0$), the bases reduce to standard Fourier atoms $z^k$, making the classical spectral layers a special case.

Next, we will explore how to optimize poles and hyperparameters by minimizing the empirical loss over $n$ training pairs:

$$\min_{a \in \mathbb{D}^m, \; \theta \in \mathbb{R}^p} L(a, \theta) = \frac{1}{n} \sum_{i=1}^n \ell\big(\hat{u}_{a,\theta}(f(x^{(i)})), \, u^{(i)}\big), \tag{9}$$

which adopts the following update rules:

$$\textbf{R-step: } a^{t+1} = \operatorname{Exp}_{a^t}\Big( - \eta_t \operatorname{grad}_a L(a^t, \theta^t) \Big), \tag{10}$$

$$\textbf{E-step: } \theta^{t+1} = \theta^t - \gamma_t \nabla_\theta L(a^{t+1}, \theta^t), . \tag{11}$$

Here, $\operatorname{Exp}_z(v) = \frac{z + \tanh\left(\frac{\lambda_z \|v\|}{2}\right) \frac{v}{\|v\|}}{1 + \bar{z} \, \tanh\left(\frac{\lambda_z \|v\|}{2}\right) \frac{v}{\|v\|}}, \lambda_z = \frac{2}{1 - |z|^2}, \operatorname{grad}_a L = \frac{(1-|a|^2)^2}{4} \operatorname{grad}_a^{\text{euc}} L$, where $\operatorname{grad}_a^{\text{euc}} L$ is the Euclidean gradient of $L$ with respect to $a$.

## 3. THEORETICAL RESULTS

We outline a few reasonable assumptions in deriving the theoretical justifications of AFD-inspired neural layers.

(A1) $|a_k| \le 1 - \varepsilon$ along iterates for some $\varepsilon \in (0, 1)$.

(A2) $a \mapsto L(a, \theta)$ is $L_a$-smooth on $(\mathbb{D}^m, g)$ and $\theta \mapsto L(a, \theta)$ is $L_\theta$-smooth in Euclidean space.

(A3) $\mu_a$-geodesic strong convexity in $a$ and $\mu_\theta$-strong convexity in $\theta$.

(A4) $\|\nabla_a \nabla_\theta L\| \le \beta$ on the feasible region.

(A5) $L$ satisfies the Kurdyka-Lojasiewicz (KL) property on the feasible set.

(A6) The sequence $\{\theta^t\}$ is bounded.

Among these assumptions, we remark that the KL property is satisfied by a broad class of functions (including subanalytic and semi-algebraic functions) that cover most objective functions used in deep learning. Also, while we acknowledge that strong convexity is a strong condition for general neural network training, we only use this assumption to establish the linear convergence rate in Theorem 3.3. We clarify that our general convergence results, which show that the objective function decreases monotonically and converges to a critical point, rely on milder smoothness of the objective function rather than strong convexity. While PDE solutions may contain shocks or sharp gradients, Equation (9) generally remains smooth with respect to parameters, making our assumption appropriate for the convergence analysis of AFD-inspired neural layers.

### 3.1. Convergence results

**Theorem 3.1.** *From Assumptions (A1) and (A2), for fixed $\theta$, the R-step update of Equation* (10) *with $\eta_t \in (0, \frac{2}{L_a})$ satisfies*

$$L(a^{t+1}, \theta) \;\le\; L(a^t, \theta) - \left(\eta_t - \tfrac{L_a}{2} \eta_t^2\right) \left\|\operatorname{grad}_a L(a^t, \theta)\right\|_g^2. \tag{12}$$

*In other words, $L(a^t, \theta)$ decreases monotonically. Furthermore, $\sum_t \left\|\operatorname{grad}_a L(a^t, \theta)\right\|_g^2 < \infty$*

*Proof.* By geodesic $L_a$-smoothness on a complete Riemannian manifold, for any $v \in T_{a^t} \mathbb{D}^m$,

$$L(\operatorname{Exp}_{a^t}(v), \theta) \le L(a^t, \theta) + \langle \operatorname{grad}_a L(a^t, \theta), v \rangle_g + \tfrac{L_a}{2} \|v\|_g^2.$$

Letting $v = -\eta_t \operatorname{grad}_a L(a^t, \theta)$ yields Equation (12). Since $\eta_t \in (0, \frac{2}{L_a})$, the decrement is nonnegative. Summing Equation (12) over $t$ yields $\sum_t \|\operatorname{grad}_a L\|_g^2 < \infty$ because

$$\sum_{t=0}^N \left(\eta_t - \tfrac{L_a}{2} \eta_t^2\right) \left\|\operatorname{grad}_a L(a^t, \theta)\right\|_g^2 \le \sum_{t=0}^N \left(L(a^t, \theta) - L(a^{t+1}, \theta)\right) < L(a^0, \theta) < \infty.$$

□

**Theorem 3.2.** *Assuming (A1)-(A3) hold with $\mu_a > 0$. For $\eta \in (0, \frac{1}{L_a}]$ and the unique minimizer $a^\star$ of $a \mapsto L(a, \theta)$,*

$$L(a^{t+1}, \theta) - L(a^\star, \theta) \;\le\; (1 - \eta \mu_a) \left(L(a^t, \theta) - L(a^\star, \theta)\right). \tag{13}$$

*Proof.* Assumption (A3) implies the Polyak-Lojasiewicz inequality on manifolds [18]: $\frac{1}{2} \|\operatorname{grad}_a L\|_g^2 \ge \mu_a \big(L - L^\star\big)$. Combining it with Equation (12) for $\eta \le \frac{1}{L_a}$ gives $L(a^{t+1}, \theta) - L(a^\star, \theta) \le \big(1 - \eta \mu_a\big)\big(L(a^t, \theta) - L(a^\star, \theta)\big)$. □

**Theorem 3.3.** *Assuming (A1), (A2), and (A5) hold with step sizes $\eta_t \in (0, \frac{2}{L_a})$ and $\gamma_t \in (0, \frac{2}{L_\theta})$, the sequence $\{(a^t, \theta^t)\}$ generated by Equation* (10) *and Equation* (11) *satisfies:*

*(i) $L(a^t, \theta^t)$ is monotonically decreasing and convergent.*

*(ii) Every limit point is a first-order critical point of $L$.*

*(iii) If the sequence is bounded, it converges to a single critical point.*

*If, in addition, Assumptions (A3) and (A4) hold and $\beta$ is sufficiently small, $L(a^t, \theta^t)$ converges linearly.*

*Proof.* Let $x^t = (a^t, \theta^t)$ denote the sequence of iterates. We prove each claim in order.

To prove (i), note that the R-step update for $a$, executed on the function $L(\cdot, \theta^t)$, is a Riemannian gradient step. From Assumption (A2), we have the standard descent lemma:

$$L(a^{t+1}, \theta^t) \le L(a^t, \theta^t) - c_t \|\mathrm{grad}_a L(a^t, \theta^t)\|_g^2, \qquad (14)$$

where $c_t := \eta_t - \frac{L_a}{2}\eta_t^2 > 0$. Similarly, the E-step update for $\theta$ on the $L_\theta$-smooth function $L(a^{t+1}, \cdot)$ yields:

$$L(a^{t+1}, \theta^{t+1}) \le L(a^{t+1}, \theta^t) - d_t \|\nabla_\theta L(a^{t+1}, \theta^t)\|^2, \qquad (15)$$

where $d_t := \gamma_t - \frac{L_\theta}{2}\gamma_t^2 > 0$.

Combining (14) and (15) gives:

$$L(a^{t+1}, \theta^{t+1}) \le L(a^{t+1}, \theta^t) \le L(a^t, \theta^t).$$

From Assumption (A1), iterates $\{a^t\}$ are bounded. For the sequence $\{L(a^t, \theta^t)\}$ to converge, it must be bounded. From Assumption (A6), the full sequence of iterates $\{(a^t, \theta^t)\}$ is in a compact set. Furthermore, it is continuous from (A2). Since any continuous function on a compact set is bounded, $\{L(a^t, \theta^t)\}$ converges. This proves (i).

To prove (ii), we add Equations (14) and (15) to get:

$$\begin{aligned} 0 \le c_t \|\mathrm{grad}_a L(a^t, \theta^t)\|_g^2 + d_t \|\nabla_\theta L(a^{t+1}, \theta^t)\|^2 \\ \le L(a^t, \theta^t) - L(a^{t+1}, \theta^{t+1}). \end{aligned}$$

Summing it from $t = 0$ to $\infty$, the RHS is bounded by $L(a^0, \theta^0) - \lim_{t\to\infty} L(a^t, \theta^t) < \infty$, and the convergence of this series implies its terms must converge to 0. Assuming that step sizes are bounded away from the interval endpoints, applying the squeeze theorem gives:

$$\lim_{t\to\infty} \|\mathrm{grad}_a L(a^t, \theta^t)\|_g = 0 \text{ and } \lim_{t\to\infty} \|\nabla_\theta L(a^{t+1}, \theta^t)\| = 0.$$

From Assumption (A2), gradient $\nabla_\theta L$ is Lipschitz continuous. Since $d(a^{t+1}, a^t) = \eta_t \|\mathrm{grad}_a L(a^t, \theta^t)\|_g \to 0$, it follows that $\|\nabla_\theta L(a^t, \theta^t)\| \to 0$. As both gradient components vanish, any limit point must be a critical point. This proves (ii).

To prove (iii), we first leverage Assumption (A1), which ensures $\{a^t\}$ is bounded. Since $\{\theta^t\}$ is bounded, the sequence $\{(a^t, \theta^t)\}$ is bounded. By the Bolzano-Weierstrass theorem, there exists at least one limit point $(a^\star, \theta^\star)$, which by (ii) must be a critical point. Next, we leverage Assumption (A5). The key result for functions satisfying the KL property implies that a bounded sequence with vanishing gradients must have a finite length:

$$\sum_{t=0}^{\infty} d\big((a^{t+1}, \theta^{t+1}), (a^t, \theta^t)\big) < \infty.$$

A sequence of finite length is a Cauchy sequence, which converges in a complete metric space. Thus, the full sequence $\{(a^t, \theta^t)\}$ converges to a single critical point. This proves (iii).

Based on Assumptions (A3) and (A4), we can establish a linear convergence rate. We consider the contraction of a Lyapunov function $\Psi_t := d(a^t, a^\star)_g^2 + \|\theta^t - \theta^\star\|^2$, where $(a^\star, \theta^\star)$ is the unique minimizer. Standard gradient descent analysis on strongly convex functions indicates that each block update is a contraction perturbed by the cross-variable coupling. This leads to a recursive system of inequalities:

$$\begin{aligned} d(a^{t+1}, a^\star)^2 &\le (1 - \eta\mu_a) d(a^t, a^\star)^2 + O(\beta^2)\|\theta^t - \theta^\star\|^2, \\ \|\theta^{t+1} - \theta^\star\|^2 &\le (1 - \gamma\mu_\theta)\|\theta^t - \theta^\star\|^2 + O(\beta^2) d(a^{t+1}, a^\star)^2. \end{aligned}$$

This can be written as a vector inequality $\mathbf{v}_{t+1} \le M\mathbf{v}_t$, where $\mathbf{v}_t = [d(a^t, a^\star)^2, \|\theta^t - \theta^\star\|^2]^T$. The contraction matrix $M$ has diagonal entries less than 1 and off-diagonal entries proportional to $\beta^2$. For a sufficiently small $\beta$, the spectral radius $\rho(M)$ is less than 1. This implies $\Psi_{t+1} \le \rho\Psi_t$ for some $\rho < 1$, which indicates linear convergence. □

### 3.2. Increasing the number of poles

Let $L^{(m)}$ denote the objective with $m$ poles. At outer stage $m$, run the inner alternating algorithm to stationarity, then insert a new pole using a greedy rule and continue. This is illustrated in Algorithm 1.

**Algorithm 1** AFD-inspired neural layers

```
Input: data $\{(f^{(i)}, u^{(i)})\}$, initial $m$, stepsizes $(\eta, \gamma)$, buffer $\varepsilon$
Initialize poles $a \in \mathbb{D}^m$ and weights $\theta$
repeat                                                  ▷ outer loop at fixed $m$
    repeat                                              ▷ inner alternating
        $a \leftarrow \mathrm{Exp}_a(-\eta\, \mathrm{grad}_a L)$                          ▷ R-step
        Project $a$ to satisfy $|a_k| \le 1 - \varepsilon$
        $\theta \leftarrow \theta - \gamma\, \nabla_\theta L$                              ▷ E-step
    until $\|\mathrm{grad}_a L\| + \|\nabla_\theta L\| \le \tau$
    Add pole $a_{m+1}$ with proxy gain $\Delta_m$                     ▷ greedy growth
    Set $m \leftarrow m + 1$
until $\Delta_m \le \delta$
```

**Definition 3.4** (Greedy gain [19]). *Let $x_\star^{(m)} = (a_\star^{(m)}, \theta_\star^{(m)})$ be an inner stationary point. The warm start $x_0^{(m+1)}$ has gain $\Delta_m \ge 0$ if $L^{(m+1)}(x_0^{(m+1)}) \le L^{(m)}(x_\star^{(m)}) - \Delta_m$.*

**Theorem 3.5.** *If each inner loop reaches stationarity and the outer step has a gain $\Delta_m \ge 0$, then the sequence $V_m := L^{(m)}(x_\star^{(m)})$ is nonincreasing and convergent. If $\inf_m \Delta_m > 0$, the procedure terminates after finitely many outer steps. If the target is exactly representable at some $\bar{m}$ and the inner loops reach the unique minimizer, then the algorithm achieves the global optimum at $m = \bar{m}$.*

*Proof.* Based on Definition 3.4 and inner optimality,

$$V_{m+1} \le L^{(m+1)}(x_0^{(m+1)}) \le V_m - \Delta_m,$$

hence $(V_m)$ is nonincreasing and bounded below, and thus convergent. If $\inf_m \Delta_m > 0$, strict decrease implies finite termination. Exact representability at $\bar{m}$ and exact inner solves imply $V_{\bar{m}} = L^\star$ and stagnation thereafter. □

## 4. NUMERICAL EXPERIMENTS

In this section, we experimentally validate our proposed method in solving both forward and inverse PDE problems. We run all experiments in a Dell Precision 7920 Tower equipped with Intel Xeon Gold 6246R CPU and NVIDIA Quadro RTX 6000 GPU (with 24GB GGDR6 memory).

### 4.1. Self comparison and comparison with benchmark solvers

Table 1 presents the relative $L^2$ error results of our AFD-inspired neural operator with and without Equations (10) and (11), as well as three benchmark solvers. Clearly, with pole and hyperparameter optimization, the accuracy of AFD-inspired neural layer is significantly enhanced for both forward (Burgers' equation) and inverse (Darcy flow equation) problems. Also, our method outperforms benchmark solvers in solving the inverse Darcy flow problem. Although our model does not perform the best in Burgers' equation, we point out that our model contains only one neural layer and only takes $\approx 1.1$ seconds/epoch for training. We utilize the initial $m = 16$, step size $\eta = 0.3$, $\gamma = 2 \times 10^{-3}$, and $\varepsilon = 10^{-7}$ to train our model for 100 epochs.

| Models | Burgers' equation | Inverse Darcy flow |
|---|---|---|
| Ours (full) | 4.32E-03 | 7.65E-02 |
| Ours (Euclidean) | 1.64E-02 | 4.05E-01 |
| NAO [20] | 7.87E-03 | 7.71E-02 |
| MWT [10] | 1.17E-02 | 9.73E-01 |
| FNO [13] | 1.05E-03 | 8.01E-02 |

**Table 1**. Comparison of relative $L^2$ error among different models on Burgers' (forward problem) and Darcy flow (inverse problem) equations.

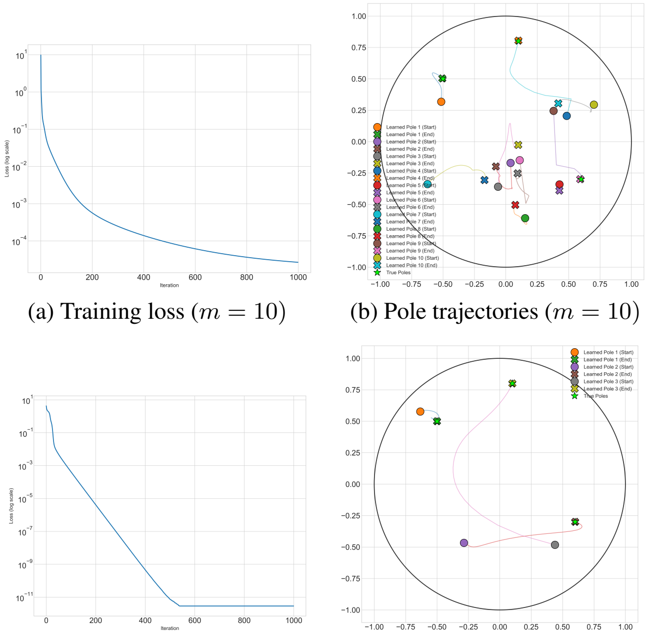
Additional experiments feature more complex geometries and realistic datasets. When solving the magnetic Schrödinger equation on a nontrivially projectable complex manifold (closed unit ball with Kähler metric) [21], our full AFD-inspired neural operator achieves a relative $L^2$ error of 3.02E-03, whereas FNO [13] has a much higher relative $L^2$ error of 5.56E-01. When learning the real-world latex glove Digital Image Correlation measurement dataset, state-of-the-art Implicit Fourier Neural Operator (IFNO) [22] has a relative $L^2$ error of 3.30E-02, whereas our full AFD-inspired neural operator achieves a lower relative $L^2$ error of 2.97E-03.

### 4.2. Validation of Theorem 3.3 and Theorem 3.5

To validate Theorem 3.3 and Theorem 3.5, we construct a synthetic target signal that is exactly representable by three poles. We fix $N = 128$ equally spaced points $t \in [-5, 5]$ and map them into the unit disk as $z_j = \rho \cdot \frac{2(t_j - \min t)}{\max t - \min t} - \rho$, where $\rho = 0.92$. Then, we select three poles inside the unit disk, $a_1 = 0.3 + 0.2i, a_2 = -0.55 + 0.1i, a_3 = 0.15 - 0.6i$, each satisfying $|a_k| < 1$. For each pole, the analytic atom is defined as $\phi(a; z) = \frac{\sqrt{1-|a|^2}}{1-\bar{a}z}$. Next, we choose complex weights $\theta_1 = 1.2 - 0.8i, \theta_2 = -0.9 + 0.2i$, and $\theta_3 = 0.6 + 0.7i$. The ground truth signal is then given by $y_{\text{true}}(z) = \theta_1 \phi(a_1; z) + \theta_2 \phi(a_2; z) + \theta_3 \phi(a_3; z)$. One can observe from Figure 1 (a) and (c) that Algorithm 1 is convergent, while linear convergence is observed only at $m = \bar{m} = 3$. We also find that the true poles can be identified when $m = \bar{m}$. Moreover, from Figure 1 (b) and (d), it is clear that having more than 3 poles may not be necessary. Finally, the nonincreasing and convergent nature of $V_m$ and the global optimum can be verified from Figure 2.

## 5. CONCLUSIONS

To summarize, we propose an AFD-inspired neural layer that explicitly incorporates Blaschke-type bases and reconstructs AFD operations in a data-driven way. The AFD theory has demonstrated remarkable effectiveness in signal processing through its adaptive selection of poles. However, in the context of neural networks, poles cannot be directly optimized during training. To address this issue, we leverage the geometric insight that the unit disk, where the poles reside, constitutes a Riemann surface (manifold) with constant negative curvature of $-1$. This perspective allows us to design an iterative optimization scheme, which consists of a R-step for updating poles along the Riemannian manifold and a E-step for updating network parameters within the Euclidean space.



(a) Training loss ($m = 10$)

(b) Pole trajectories ($m = 10$)

(c) Training loss ($m = 3$)

(d) Pole trajectories ($m = 3$)

**Fig. 1**. Numerical results of the target signal. Note that $y_{\text{true}}$ lies in the span of three poles, i.e., $\bar{m} = 3$.

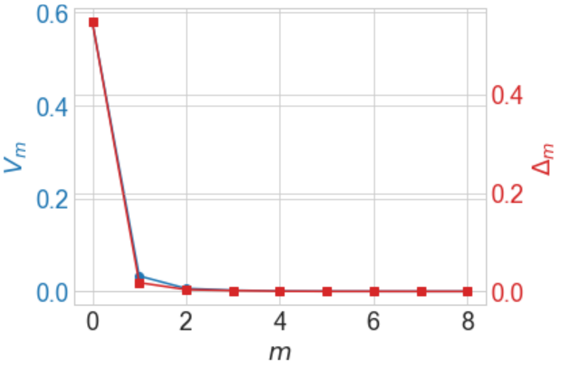


**Fig. 2**. Visualization of $V_m$ and $\Delta_m$.

Numerical experiments on both forward and inverse PDE problems demonstrate the attractiveness of our AFD-inspired neural layer to generalize across various tasks that require expressive and adaptive function representations. Thus, this work marks an initial step toward systematically embedding AFD theory into neural architectures. We believe that such a mathematically grounded neural layer design can be flexibly integrated into a wide range of existing network models, offering new opportunities for manifold-aware operator learning.

Our future work will focus on exploring strategies for identifying the minimum number of poles, $\bar{m}$, required to achieve global optima on real-world datasets. Such advancements not only offer new theoretical insights of our AFD-inspired neural layer but also enhance its computational efficiency and accuracy in solving large-scale, complex PDE problems.

## 6. ACKNOWLEDGEMENT

This work is funded by the U.S. National Science Foundation (NSF) under Award Number 2442806.

## 7. REFERENCES

[1] Zeyuan Song and Zheyu Jiang, "Mp-fvm: Enhancing finite volume method for water infiltration modeling in unsaturated soils via message-passing encoder-decoder network," *Computers and Geotechnics*, vol. 190, pp. 107745, 2026.

[2] Zeyuan Song and Zheyu Jiang, "A physics-based, data-driven numerical framework for anomalous diffusion of water in soil," *Systems & Control Transactions*, vol. 4, pp. 2391–2397, 2025.

[3] Zeyuan Song and Zheyu Jiang, "A novel bayesian framework for inverse problems in precision agriculture," *Systems & Control Transactions*, vol. 4, pp. 246–251, 2025.

[4] Zeyuan Song and Zheyu Jiang, "A data-driven modeling approach for water flow dynamics in soil," *Computer Aided Chemical Engineering*, vol. 52, pp. 819–824, 2023.

[5] Zhiping Mao and George Em Karniadakis, "A spectral method (of exponential convergence) for singular solutions of the diffusion equation with general two-sided fractional derivative," *SIAM Journal on Numerical Analysis*, vol. 56, no. 1, pp. 24–49, 2018.

[6] Tao Qian, "Intrinsic mono-component decomposition of functions: an advance of Fourier theory," *Mathematical Methods in the Applied Sciences*, vol. 33, no. 7, pp. 880–891, 2010.

[7] Tao Qian, Liming Zhang, and Zhixiong Li, "Algorithm of adaptive Fourier decomposition," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5899–5906, 2011.

[8] Zeyuan Song and Zuoren Sun, "Representing functions in $H^2$ on the Kepler manifold via WPOAFD based on the rational approximation of holomorphic functions," *Mathematics*, vol. 10, no. 15, pp. 2729, 2022.

[9] Zeyuan Song and Zheyu Jiang, "Adaptive mamba neural operators," in *The Fourteenth International Conference on Learning Representations*, 2026.

[10] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan, "Multiwavelet-based operator learning for differential equations," *Advances in neural information processing systems*, vol. 34, pp. 24048–24062, 2021.

[11] Xiongye Xiao, Defu Cao, Ruochen Yang, Gaurav Gupta, Gengshuo Liu, Chenzhong Yin, Radu Balan, and Paul Bogdan, "Coupled multiwavelet operator learning for coupled differential equations," in *The Eleventh International Conference on Learning Representations*, 2022.

[12] Tapas Tripura and Souvik Chakraborty, "Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 404, pp. 115783, 2023.

[13] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint arXiv:2010.08895*, 2020.

[14] Hongfang Bai, Ieng Tak Leong, and Pei Dang, "Reproducing kernel representation of the solution of second order linear three-point boundary value problem," *Mathematical Methods in the Applied Sciences*, vol. 45, no. 17, pp. 11181–11205, 2022.

[15] Hongfang Bai and Ieng Tak Leong, "A sparse kernel approximate method for fractional boundary value problems," *Communications on Applied Mathematics and Computation*, vol. 5, no. 4, pp. 1406–1421, 2023.

[16] Peter Y Lu, Samuel Kim, and Marin Soljačić, "Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning," *Physical Review X*, vol. 10, no. 3, pp. 031056, 2020.

[17] Étienne Ghys, "Poincaré and his disk," *The scientific legacy of Poincaré*, vol. 36, pp. 17, 2006.

[18] Siwan Boufadène and François-Xavier Vialard, "On the global convergence of wasserstein gradient flow of the coulomb discrepancy," *SIAM Journal on Mathematical Analysis*, vol. 57, no. 4, pp. 4556–4587, 2025.

[19] Jerome H Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[20] Yue Yu, Ning Liu, Fei Lu, Tian Gao, Siavash Jafarzadeh, and Stewart A Silling, "Nonlocal attention operator: Materializing hidden knowledge towards interpretable physics discovery," *Advances in Neural Information Processing Systems*, vol. 37, pp. 113797–113822, 2024.

[21] Zeyuan Song and Zheyu Jiang, "Adaptive fourier decomposition-guided neural operator design for inverse PDE problems," in *Submitted to The Fourteenth International Conference on Learning Representations*, 2025, under review.

[22] Huaiqian You, Quinn Zhang, Colton J. Ross, Chung-Hao Lee, and Yue Yu, "Learning deep implicit fourier neural operators (ifnos) with applications to heterogeneous material modeling," *Computer Methods in Applied Mechanics and Engineering*, vol. 398, pp. 115296, 2022.