Learning Orthonormal Bases for Function Spaces

Hamidreza Kamkari
MIT CSAIL
Cambridge, MA
kamkarih@mit.edu

Mohammad Sina Nabizadeh
MIT CSAIL
Cambridge, MA
sinabiz@mit.edu

Justin Solomon
MIT CSAIL
Cambridge, MA
jsolomon@mit.edu

Abstract

Infinite-dimensional orthonormal basis expansions play a central role in representing and computing with function spaces due to their favorable linear algebraic properties. However, common bases such as Fourier or wavelets are fixed and do not adapt to the structure of a given problem or dataset. In this paper, we aim to represent these bases with neural networks and optimize them. Our key idea is that any target infinite-dimensional orthonormal basis can be viewed either as a point on the Lie manifold of the orthogonal group, or equivalently, as the endpoint of a continuous path on that manifold that connects a reference basis, e.g. Fourier, to that target. Paths on the Lie manifold satisfy ordinary differential equations (ODEs) governed by skew-adjoint integral operators. Using neural networks to define finite-rank generators of such ODEs allows us to parameterize and optimize orthonormal bases in function space. While relying on finite-rank generators to model infinite operators might seem restrictive, we prove a universality result: even with a rank-2 generator, the integrated solutions of the ODE are dense in the orthogonal group under the appropriate operator topology. In other words, for any target orthonormal basis, there exists a path originating from a reference basis and driven by finite-rank generators that gets arbitrarily close to that target basis. We demonstrate the flexibility of our framework by transforming the Fourier basis into the principal components of a functional dataset, eigenfunctions of linear operators, or dynamic modes of energy-preserving physical simulations.

1 Introduction

Orthonormal basis (ONB) expansions are basic building blocks of various fields. They help us represent functions, run numerical simulations, and analyze data in machine learning. The choice of basis governs not only approximation accuracy but also the computational cost of downstream tasks. Often, the most useful basis is adapted to the problem at hand. For instance, in data analysis, principal components optimally capture variance, allowing one to analyze the data by looking at only the first few basis elements [54, 1]. In physics and geometry, the first few spectral eigenfunctions approximate the dynamic or geometric modes of the system [11, 49]. In signal processing, Fourier and wavelet series [39, 6] help pinpoint various signal characteristics.

Neural networks flexibly parameterize complex function spaces and adapt to specific tasks via gradient-based optimization. Here, we leverage their expressiveness for the space of ONBs and ask:

Can we parameterize infinite series of orthonormal and continuous bases with neural networks?

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

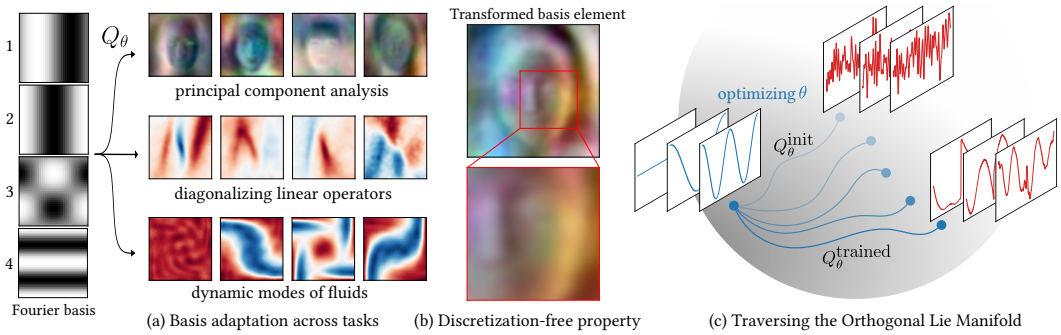


Figure 1: **Method and Applications Overview. (a)** Parameterizing a change-of-basis map Q_θ allows us to adapt a reference ONB to various applications: eigenfaces that describe the variance of the data, visualizing the training dynamics of a non-linear neural function by the eigenfunctions of its neural tangent kernel, or dynamic modes of a fluid simulation. **(b)** The learned bases inherit properties of the reference such as continuity and being discretization-free. **(c)** Our method parameterizes paths on the Lie manifold of the infinite orthogonal group and gradually transform an initial basis by traversing this manifold; by optimizing the path, we are able to implicitly learn ONBs.

Classical ONBs such as Fourier series provide a good starting point: they are infinite and orthonormal by construction, inner products decompose cleanly into coefficient sums and projection requires only a single inner product. They are also continuous: they represent functions via linear combination of continuous functions on the entire domain, rather than as discrete values on a grid. Thus, the resulting space can be evaluated, differentiated, or integrated at any point in the domain, making them discretization-free and resolution-invariant.

Any two ONBs, up to sign, are connected by a special orthogonal change-of-basis operator. Therefore, rather than directly modeling the space of ONBs, we can parameterize the equivalent orthogonal map that inputs a reference ONB—e.g. the Fourier series—and outputs a basis tailored to the application at hand. This construction allows the target ONB to inherit desirable properties of this reference: remaining infinite, continuous, and orthonormal. Figure 1 (a) shows how parameterizing this operator allows us to map the Fourier basis to the principal components of the celebrity face dataset [30], recover the spectrum of a linear operator, or extract the dynamic modes of a system.

To tractably parameterize this change-of-basis operator, we draw on Lie theory [25]. Rather than optimizing an orthogonal operator directly, we view it as a continuous path across the manifold of the orthogonal group (see Figure 1; c). Geometrically, this path corresponds to a sequence of infinitesimal rotations in function space, modeled as the flow of an ODE governed by a time-dependent, skew-adjoint infinitesimal generator. The orthogonal group forms a highly constrained manifold that is hard to parameterize, but the space of skew-adjoint generators is linear and easier to model with a neural network. The remaining challenge is that even the space of generators is infinite-dimensional. To resolve this, we prove in section 4 that any path on this infinite group can be arbitrarily well-approximated using a *finite-rank* generator. Ultimately, this reduces the task of parameterizing and optimizing an infinite ONB to the following: we use a neural network to model a finite-rank, skew-adjoint generator and integrate the resulting ODE to smoothly change a reference basis into a target basis. By optimizing the network's parameters via neural ODE backpropagation [10], we model a continuous path on the orthogonal group that adapts to our needs.

Our contributions are: (i) reformulating orthonormal bases as a rotation path on the infinite-dimensional orthogonal group, achieved by solving an ODE rather than constructing an orthonormal basis from scratch; (ii) a universality theorem establishing that a large family of rotations (specifically, those generated by Hilbert-Schmidt operators) can be approximated by the flow of an ODE governed by a low-rank skew-adjoint operator; (iii) a practical neural parameterization of this generator, trained end-to-end via ODE-based backpropagation, yielding an orthonormal, continuous, and infinite basis that can be adapted to arbitrary variational problems; (iv) a principled variational objective for diagonalizing linear operators, demonstrated in two applications by recovering the eigenfunctions of the data covariance operator and the Neural Tangent Kernel; and (v) applications beyond eigenproblems to modeling dynamical systems, demonstrating that our basis transformations capture the Koopman operator to track energy-preserving dynamics.

2 Related Work

Basis expansions. A principled way to discover a problem-adapted basis is to compute the eigenfunctions of an underlying linear operator. In physics, eigenfunctions of differential operators—such as spherical harmonics, Bessel functions, and Hermite polynomials—diagonalize the Laplacian and form the foundation of spectral methods for solving differential equations [24, 58, 36, 9]. Alternatively, operators can be data-driven rather than analytic; for example, in principal component analysis (PCA), the principal components of a stochastic process are the eigenfunctions of its covariance operator [20, 1]. Likewise, for dynamical systems, the spectral decomposition of the Koopman operator [31, 41, 7] yields spatial modes that organize the dynamics and allow for reduced order modeling; dynamic mode decomposition (DMD) and its extensions for example estimate these modes directly from data [47, 49, 59, 61, 32]. In other cases, bases are engineered rather than linked to linear operators; wavelet bases are one such example which are designed to capture local signal characteristics [39]. Despite their different use cases, basis expansions share a core principle: representing functions as linear combinations of their elements isolates their useful properties while preserving the linear algebraic structure of the underlying function space.

Eigendecomposition. Solving an eigenproblem is a standard method for obtaining custom bases. The classical approach involves discretizing the operator as a matrix on a grid or mesh and applying numerical eigensolvers [48, 23, 54]. However, these approaches inevitably incur discretization error in continuous state spaces. While solving the discretized system and applying post-hoc interpolation is a possible workaround [5], recent work instead suggests using neural networks to learn continuous eigenfunctions directly by solving variational problems. For instance, Spectral Inference Networks [45] pair this approach with an explicit orthogonalization step, whereas NeuralEF [13] avoids it by extending EigenGame [19] to function space. Further extending this paradigm, Ben-Shaul et al. [4] target differential self-adjoint operators on free-form domains, and Azmoodeh et al. [3] address continuous-time PCA via the directly representing the ONBs. While these neural methods are discretization-free, they typically recover only a small, finite number of eigenfunctions without strictly enforcing orthogonality. Our framework, by contrast, maintains exact orthonormality across the full eigenfunction sequence by always remaining on the Lie manifold.

Non-linear representations of function spaces. An alternative to using orthonormal bases is to forego the linear structure of function spaces entirely in favor of highly expressive, non-linear neural parameterizations. Neural operators [37, 38, 33], implicit neural representations (INRs) [52, 42, 62, 15], and physics-informed neural networks (PINNs) [46] loosely fall into this category and have achieved success across image processing, computer graphics, and physical simulation [43, 50]. Neural representations have also proven effective for inverse problems where the goal is to reconstruct data from sparse measurements through prior-informed image representations [51], learned measurement fields [55], or physics-consistent models [40]. However, while these representations adapt remarkably well to specific problems, they sacrifice the linear algebraic structure of the function space: linear operations in function space—such as addition or applying a differential operator—map to highly non-linear and unpredictable transformations in the network's weight space [63, 64, 12]. While recent architectures like the Fourier tensor network [2] have begun to address this by, for example, explicitly structuring INRs with orthogonal Fourier bases and low-rank tensor coefficients, our framework offers an alternative: we leverage the adaptive power of neural networks while strictly preserving the linear algebraic structure of the function space by parameterizing bases. We even demonstrate how to bridge these two worlds; in one of our applications, we take an arbitrary non-linear neural network and project it back into a structured function space by learning the continuous ONB that diagonalizes its neural tangent kernel.

3 Mathematical Preliminaries

Function spaces. We work with functions defined on a d-dimensional compact domain Ω equipped with finite measure μ. We consider the real-valued Hilbert space $H \coloneqq L^2(\Omega, \mu)$, with its inner product and norm defined in standard fashion: $\langle f, g\rangle \coloneqq \int_\Omega f(\omega)g(\omega)\,\mu(d\omega) = \mathbb{E}_{\omega\sim\mu}[f(\omega)g(\omega)]$ and $\|f\|^2 \coloneqq \langle f, f\rangle$. Since μ is finite, we may without loss of generality assume it is a probability measure. Consequently, the inner product is an expectation, allowing for Monte Carlo estimation. We assume H is *separable*, meaning it admits a countable complete ONB $\{\varphi_n\}_{n=1}^\infty$. Concretely, for any $f \in H$, $\lim_{n\to\infty} \|f - \sum_{m=1}^n \langle f, \varphi_m\rangle \cdot \varphi_m\| = 0$. For simplicity, we assume Ω possesses

a standard Euclidean metric; thus, a valid ONB $\{\varphi_n\}_{n=1}^{\infty}$ (among many) can be the d-dimensional real Fourier basis. This basis may be suboptimal for a given problem, motivating us to change it to a problem-adapted one; we will, henceforth, use $\{\varphi_i\}_{i=1}^{\infty}$ to denote the initial (reference) basis.

Linear operators. A bounded linear transformation on H is a *linear operator*. Any such operator A admits a unique adjoint denoted by A^*, defined such that: $\langle f, Ag\rangle = \langle A^* f, g\rangle, \forall f, g \in H$. When A can be written in an integral kernel form $A[\cdot,\cdot] : \Omega \times \Omega \to \mathbb{R}$, its action follows the identity:

$$(Af)(x) = \int_{\Omega} A[x,y]\, f(y)\, \mu(dy); \quad \text{consequently,} \quad \forall x, y \in \Omega : A^*[x,y] = A[y,x]. \tag{1}$$

A is *skew-adjoint* if $A^* = -A$, or equivalently, $\forall x, y \in \Omega : A[x,y] = -A[y,x]$.

The space of operators themselves have norms and topology. Here, we use the *Hilbert-Schmidt* norm:

$$\|A\|_{\text{HS}}^2 := \sum_{n=1}^{\infty} \|A\varphi_n\|^2 = \sum_{n=1}^{\infty} \langle A\varphi_n, A\varphi_n\rangle = \sum_{n=1}^{\infty} \langle \varphi_n, A^*A\varphi_n\rangle =: \text{tr}(A^*A). \tag{2}$$

where $\{\varphi_n\}_{n=1}^{\infty}$ can be any complete ONB. An operator A is Hilbert-Schmidt if this norm is finite—or equivalently, if A^*A is *trace-class*, meaning its trace is finite and well-defined.

Infinite-dimensional orthogonal group. Any two ONBs, $\{\varphi_i\}_{i=1}^{\infty}$ and $\{\phi_i\}_{i=1}^{\infty}$, are related by an orthogonal change-of-basis operator $Q : H \to H$, where $Q^*Q = I$.

To make our search for a basis tractable, we borrow a classical concept from Lie theory [25]: a large subset of orthogonal operators Q can be characterized by a smooth manifold obtained via the exponentiation of a skew-adjoint operator. Concretely, let $Q(t)$ be a family of operators obeying the following ODE:

$$\frac{d}{dt}Q(t) = KQ(t), \quad Q(0) = I, \qquad \text{where } K : H \to H \text{ is skew-adjoint.} \tag{3}$$

The solution, $Q(t)$, remains orthogonal because the inner product Q^*Q is preserved:

$$\frac{d}{dt}(Q^*Q) = \dot{Q}^*Q + Q^*\dot{Q} = Q^*K^*Q + Q^*KQ = Q^*(K^* + K)Q = 0. \tag{4}$$

Searching over the smooth family of generators K instead of Q, establishes a tractable search space. Formally, this allows us to work with the following continuous group:

Definition 1. The *Hilbert-Schmidt orthogonal group*, denoted $\mathcal{SO}(HS)$, is the space of operators that can be written as solutions to the ODE in (3), where K is additionally Hilbert-Schmidt.

4 Approximation Theory of the Orthogonal Group

Our goal is to represent an orthogonal change-of-basis operator $Q \in \mathcal{SO}(HS)$ using the parameters θ of a neural network. As a first step, rather than directly modeling Q, we model its skew-adjoint infinitesimal generator $K_\theta(t)$ as a time-evolving operator:

$$\frac{d}{dt}Q(t) = K_\theta(t)Q(t), \qquad Q(0) = I, \tag{5}$$

integrated to time $T > 0$. Even when $K_\theta(t)$ is time-dependent, (4) shows $Q(t)$ is an orthogonal time-evolving operator. In a slight abuse of notation, we will denote this operator by $Q_\theta := Q(T)$, dropping the time evaluation and using θ emphasize its parameterization.

We have reduced the problem of designing an orthogonal operator to designing a skew-adjoint operator with a neural network. If the infinitesimal generator K_θ were to have finite rank $r = 2$, we could represent its skew-adjoint kernel using two *unrestricted* multilayer perceptrons (MLPs):

$$K_\theta(t)[x,y] = \text{MLP}_1(x,t;\theta)\cdot\text{MLP}_2(y,t;\theta) - \text{MLP}_2(x,t;\theta)\cdot\text{MLP}_1(y,t;\theta). \tag{6}$$

Although designing the generator to have a finite rank might appear as a restrictive choice, integrating low-rank dynamics over time allows one to approximate *any* transformation in $\mathcal{SO}(HS)$. This is true even when the rank is as small as $r = 2$. We now make this concrete:

Theorem 1 (Approximating the Orthogonal Group via Rank-2 Generators)**.** *Let $\mathcal{Q}_T$ denote the set of all operators $Q(T)$ obtained by solving the initial value problem (5) at time $T > 0$, where the generator $K(t)$ is uniformly Hilbert-Schmidt, continuous, and rank-2. Then, $\mathcal{Q}_T$ is dense in $\mathcal{SO}(HS)$ with respect to the Hilbert-Schmidt norm topology. In other words, every rotation in this infinite orthogonal group is arbitrarily well-approximated by some time-evolving rank-2 generator.*

Proof Sketch. We first approximate an arbitrary $Q \in \mathcal{SO}(HS)$ with an operator that acts non-trivially only on a finite n-dimensional subspace. Restricted to this subspace, we have a constructive proof: this operator is isomorphic to an $n \times n$ orthogonal matrix. A classical result from linear algebra states that any finite-dimensional rotation can be factorized into a sequence of planar Givens' rotations [23]. Each planar rotation is generated by a fixed rank-2 skew-adjoint operator and by using a time-dependent scalar control function, we can chain these stationary generators in sequence; effectively "turning on" each planar rotation one after the other. This constructs a single continuous time-varying rank-2 generator whose time-ordered integral results in the target finite-dimensional rotation. Finally, taking the limit of $n \to \infty$ in the Hilbert-Schmidt operator topology extends this approximation to the entire infinite-dimensional group $\mathcal{SO}(HS)$; for full details, see Appendix A. □

Theorem 1 thus establishes that an entire change-of-basis can be implicitly parameterized through a finite-rank K_θ as in (6). Due to the universal approximation theorem of MLPs (the version that uses compactness of Ω by Hornik et al. [27]), even the rank $r = 2$ model in (6) with a sufficiently overparameterized θ can implicitly represent an arbitrary basis in function space.

5 Variational Problems: Adapting the Basis to Various Tasks

The previous sections establish how a path on the orthogonal group manifold can be parameterized with a neural network and traversed while preserving structure. Our next task is to put this parameterization to use. To that end, we derive variational objective functions that reward or penalize paths on the Lie manifold and allow us to optimize it and find useful bases.

Eigenproblems. One class of variational problems is finding Q_θ that *diagonalizes* an infinite-dimensional linear operators $A : H \to H$; this is akin to finding eigenvectors of $n \times n$ matrices in the finite case. In our formulation, we can freely search the space of operators $Q \in \mathcal{SO}(HS)$ to find one that diagonalizes A. To do so, we reformulate the eigenproblem in the variational language:

Theorem 2. *Let A be a trace-class positive semidefinite linear operator, and let $\{p(i)\}_{i=1}^{\infty}$ be a strictly decreasing sequence of positive weights such that $\sum_{i=1}^{\infty} p(i) = 1$, and $\{\varphi_i\}_{i=1}^{\infty}$ be any orthonormal basis on H. Consider Q^{opt} as the solution to the following variational problem:*

$$\max_{Q \in \mathcal{SO}(HS)} \quad \sum_{i=1} p(i) \cdot \langle Q\varphi_i, AQ\varphi_i \rangle \tag{7}$$

Then, $\phi_i^\star := Q^{opt}\varphi_i$ are the eigenfunctions of A. That is, $A\phi_i^\star = \lambda_i \phi_i^\star$ for all $i \geq 1$, where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$ are the corresponding eigenvalues.

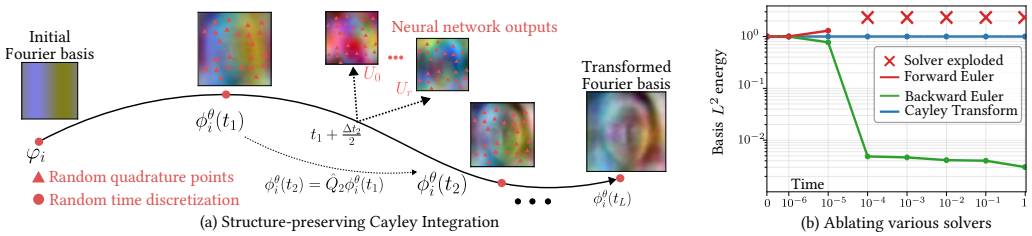
See Appendix B for a proof. Interpreting $p(\cdot)$ as a probability distribution on indices i allows us to write the objective as an expectation. Furthermore, replacing Q with Q_θ changes an otherwise intractable search over $\mathcal{SO}(HS)$ into one over parameters θ, compatible with stochastic optimization:

$$\max_{\theta} \quad \mathbb{E}_{i \sim p} \langle Q_\theta \varphi_i, AQ_\theta \varphi_i \rangle. \tag{8}$$

Substituting various operators for A allows us to solve different eigenproblems in function space. We provide an example below; Appendix B details another involving the neural tangent kernel.

PCA in function space. We formulate PCA in function space as an eigenproblem.[1] In this setting, we assume our data consists of random functions X drawn from an Ω-indexed stochastic process. This process can be formally viewed as a probability distribution $\mathbb{P}$ over the Hilbert space H. For simplicity of exposition, we assume that these functions have zero mean, $\mathbb{E}_{X \sim \mathbb{P}}[X] = 0$; we defer the treatment of the non-centered case to Appendix B.

[1] PCA in function space is sometimes referred to as the Karhunen-Loéve expansion in the literature [53, 21, 60].



(a) Structure-preserving Cayley Integration (b) Ablating various solvers

Figure 2: **Numerical Pipeline. (a)** A Fourier basis is gradually transformed into principal functions of CelebA dataset. We select quadrature points $\widehat{\Omega} = \{\omega_j\}_{j=1}^{D} \sim \mu$ and time-discretization $\{t_0 = 0, \ldots, t_L = T\}$ and perform L Cayley steps. Each step takes $\mathcal{O}(r^3 + r^2 D)$ that exploits the rank r structure of the generator. **(b)** Using off-the-shelf ODE solvers result in loss or explosion of the function norm, by contrast, Cayley integration keeps it intact.

The covariance operator $\mathcal{C} : H \to H$ is defined by its action on any function $f \in H$ as $\mathcal{C}f = \mathbb{E}_{X\sim\mathbb{P}}[X\langle X, f\rangle]$. Plugging this definition into (8) yields the following derivation:

$$\mathbb{E}_{i\sim p}\langle Q_\theta\varphi_i, \mathcal{C}Q_\theta\varphi_i\rangle = \mathbb{E}_{i\sim p}\big\langle Q_\theta\varphi_i, \mathbb{E}_{X\sim\mathbb{P}}\big[X\langle X, Q_\theta\varphi_i\rangle\big]\big\rangle = \mathbb{E}_{i\sim p, X\sim\mathbb{P}}\langle X, Q_\theta\varphi_i\rangle^2. \tag{9}$$

This objective maximizes the explained variance of the stochastic process $\mathbb{P}$. Practically, the optimization algorithm is as follows: sample $i \sim p$, retrieve the corresponding initial (Fourier) basis function φ_i, apply Q_θ, sample a random function $X \sim \mathbb{P}$, estimate the squared projection $\langle X, Q_\theta\varphi_i\rangle^2$ using random quadrature points $\widehat{\Omega}$, finally maximize via stochastic gradient ascent on θ.

Koopman operators. We also cover an application that cannot be formulated as an eigenproblem. For a dynamical system governed by a state transformation $\Psi : \Omega \to \Omega$, the associated Koopman operator $\mathcal{U}$ [31] acts on observable functions via composition $\mathcal{U}f := f \circ \Psi$. When Ψ is volume-preserving (e.g. incompressible flow), $\mathcal{U}$ becomes orthogonal. Because Q_θ is orthogonal by design, it provides a suitable inductive bias to learn these dynamics. To that end, we directly fit Q_θ to the Koopman $\mathcal{U}$ by minimizing the expected projection error:

$$\min_\theta \quad \mathbb{E}_{i\sim p}\,\|Q_\theta\varphi_i - \varphi_i \circ \Psi\|^2. \tag{10}$$

Since both Q_θ and $\mathcal{U}$ are linear, training them to match on the complete basis $\{\varphi_i\}$ is sufficient to fit Q_θ to $\mathcal{U}$. In turn, the learned ONB $\phi_i = Q_\theta\varphi_i$ represents the *dynamic modes* of the system.

This formulation becomes particularly valuable when Ψ cannot be numerically simulated in a strictly volume-preserving manner. In such cases, learning Q_θ that is orthogonal by design guarantees energy preservation in the system—$\forall f \in H : \|Q_\theta f\|^2 = \|f\|^2$; even when $\|f \circ \Psi\|^2 \neq \|f\|^2$ numerically. In Figure 6, we demonstrate this via a simple fluid dynamical flow: by repeatedly transforming a Fourier basis using Q_θ, we generate future state snapshots that strictly preserve norm, despite being trained on imperfect numerical approximations of Ψ.

6 Algorithmic Details

Our objectives require the action of Q_θ on the initial basis functions φ_i. Defining $\phi_i^\theta(t) := Q_\theta(t)\varphi_i$ and substituting into (5), we obtain

$$\frac{d}{dt}\phi_i^\theta(t) = K_\theta(t)\phi_i^\theta(t), \qquad \phi_i^\theta(0) = \varphi_i. \tag{11}$$

In principle, one could integrate (11) with off-the-shelf ODE solvers and optimize θ. However, standard solvers like forward and backward Euler accumulate numerical error over time and violate the orthonormal structure; we have included a comparison in Figure 2 with more details in Appendix C. We therefore use a structure-preserving Lie group integrator [14, 28, 8] that preserves inner products.

Structure-preserving ODE integration. Our Lie integrator is the Cayley method [28], a structure preserving method which discretizes time as $t_0 = 0 \le t_1 < \ldots < t_L = T$ and advance the system from $t_{\ell-1}$ to t_ℓ. For each ℓ, the generator is approximated by a midpoint rule $\widehat{K}_\ell = K_\theta(t_{\ell-1} + \frac{\Delta t_\ell}{2})$ and the system is evolved by applying $Q_\ell = \exp\left(\Delta t \cdot \widehat{K}_\ell\right)$. An important step in the Cayley method is that it replaces the exponential with its $(1,1)$-Padé approximant which preserves inner products:

$$\phi_i^\theta(t_\ell) \approx \widehat{Q}_\ell\,\phi_i^\theta(t_{\ell-1}), \qquad \widehat{Q}_\ell := \left(I - \tfrac{\Delta t}{2}\widehat{K}_\ell\right)^{-1}\left(I + \tfrac{\Delta t}{2}\widehat{K}_\ell\right) \approx Q_\ell. \tag{12}$$

One can verify algebraically that if $\widehat{K}_\ell$ is skew-adjoint, then $\widehat{Q}_\ell$ is orthogonal [23, 18], so the Cayley update preserves the norms and inner products of the basis exactly at each step.

Applying (12) requires discretization at some quadrature points $\widehat{\Omega} = \{\omega_j\}_{j=1}^D \sim \mu$, where the ω_j are drawn via stratified sampling. To avoid overfitting to any discretization, we resample the points before each ODE solve. Hence, discretely, $\widehat{\phi}_i^\theta(t) \in \mathbb{R}^D$, $\widehat{Q}_\ell \in \mathbb{R}^{D\times D}$, and $\widehat{K}_\ell \in \mathbb{R}^{D\times D}$ are no longer functions and operators, but random vectors and matrices. Similarly, we resample in each training iteration $t_\ell \sim \mathrm{Unif}(0, T)$ to avoid overfitting to any particular time discretization.

Low-rank parameterization. While it is mathematically valid to parameterize $K_\theta(t)$ using a rank-2 generator in (6), we generalize it to higher ranks $r \geq 2$ for more expressiveness. In particular,

$$K_\theta(t) = U_\theta^*(t)\big(M_\theta(t) - M_\theta(t)^\top\big)U_\theta(t), \tag{13}$$

where $U_\theta(t) : H \to \mathbb{R}^r$ has r rows (i.e. r functions in H) and $M_\theta(t) \in \mathbb{R}^{r\times r}$ is an arbitrary matrix. By construction, $K_\theta(t)$ is skew-adjoint, has rank $\leq r$, and depends smoothly on θ. In practice, U_θ is implemented as a network that inputs both time t and coordinate $\omega \in \Omega$ and outputs r values, and M_θ is implemented as another network that inputs time t and outputs an $r \times r$ matrix. Numerically, K_θ is only evaluated on the midpoints $t_\ell + \frac{\Delta t_\ell}{2}$ and quadrature points $\widehat{\Omega}$, producing $\widehat{K}_\ell, \widehat{U}_\ell$, and $\widehat{U}_\ell^*$.

Efficient Solver. The quadrature set must be large enough ($1024 \leq D \leq 4096$ for most experiments) to accurately approximate the underlying function-space operations. Naïvely applying $\widehat{Q}_\ell$ to $\widehat{\phi}_i^\theta$ would require $\mathcal{O}(D^3)$ matrix vector products and linear solves; however, leveraging the low-rank structure of $\widehat{K}_\ell$ allows us to be much more efficient: Each step amounts to applying $\widehat{Q}_\ell$ to the input $\widehat{\phi}_i^\theta$. Due to the low-rank structure, the forward action of applying $\big(I + \frac{1}{2}\widehat{K}_\ell\big)$ can be done in $\mathcal{O}(r^2 + rD)$ time. On the other hand, for the inverse action $(I - \frac{1}{2}\widehat{K}_\ell)^{-1}$, we apply the Woodbury identity [26]:

$$\left(I - \tfrac{1}{2}\widehat{K}_\ell\right)^{-1} = I + \tfrac{1}{2}\widehat{U}_\ell^*\widehat{S}_\ell\left(I_{r\times r} - \tfrac{1}{2}\widehat{U}_\ell\widehat{U}_\ell^*\widehat{S}_\ell\right)^{-1}\widehat{U}_\ell, \tag{14}$$

where $\widehat{S}_\ell := M_\theta(t_\ell + \frac{\Delta t_\ell}{2}) - M_\theta(t_\ell + \frac{\Delta t_\ell}{2})^\top$. The Gram matrix $U_\theta U_\theta^* \in \mathbb{R}^{r\times r}$ is a small matrix of inner products, which we estimate on $\widehat{\Omega}$ with $\mathcal{O}(r^2 D)$ operations. Hence, the total backward action reduces to $\mathcal{O}(r^3 + r^2 D)$ linear solves and matrix vector products, which are all linear in D.

Figure 2 illustrates the algorithm in action. For full details on the integrator, the values of L, r, and D, ablations, and the architecture used for each component, see Appendix C. The algorithm presented here only involves real-valued functions, for experiments involving multiple channels, such as the CelebA dataset with RGB, we use a multichannel variant explained in the same appendix.

7 Experiments

In this section, we present experiments that evaluate our learned bases. Each task requires specific configuration choices. We focus on the primary takeaways and defer details and additional experiments to Appendix D. While we use the real Fourier basis as the reference basis in almost all experiments, our architecture is not tied to this choice. In fact, in Appendix D.1, we show that it also works when the Fourier reference is replaced by a Haar wavelet basis. All experiments share the Fourier indexing and index prior described in Appendix D.2, the generator architecture in Appendix D.3, and the optimization setup in Appendix D.4; per-experiment hyperparameters are listed in Table 1. Our codebase is available at [https://github.com/HamidrezaKmK/Learning-ONBs].

1D functional PCA. We evaluate our approach using a synthetic 1D experiment on the domain $\Omega = [0, 1]$ equipped with a uniform measure μ. To create a deliberately challenging scenario for standard Fourier methods, we sample functions with a sharp discontinuity at $x = 0.5$ (see Figure 3):

$$f(x) = \sigma \begin{cases} \sin(2\pi k x) + 2x & \text{if } x \leq 0.5, \\ -\sin(2\pi k x) + 2x - 2 & \text{if } x > 0.5, \end{cases} \tag{15}$$

where the scale factor $\sigma \sim$ Rademacher takes values in $\{-1, 1\}$ with equal probability, and the frequency multiplier $k \sim$ Geometric$(1/3)$ is a strictly positive integer. While discontinuous signals are notoriously difficult for standard Fourier reconstruction, our method overcomes this issue, also surpassing the baseline finite PCA reconstruction with as few as 10 basis functions. Moreover, as

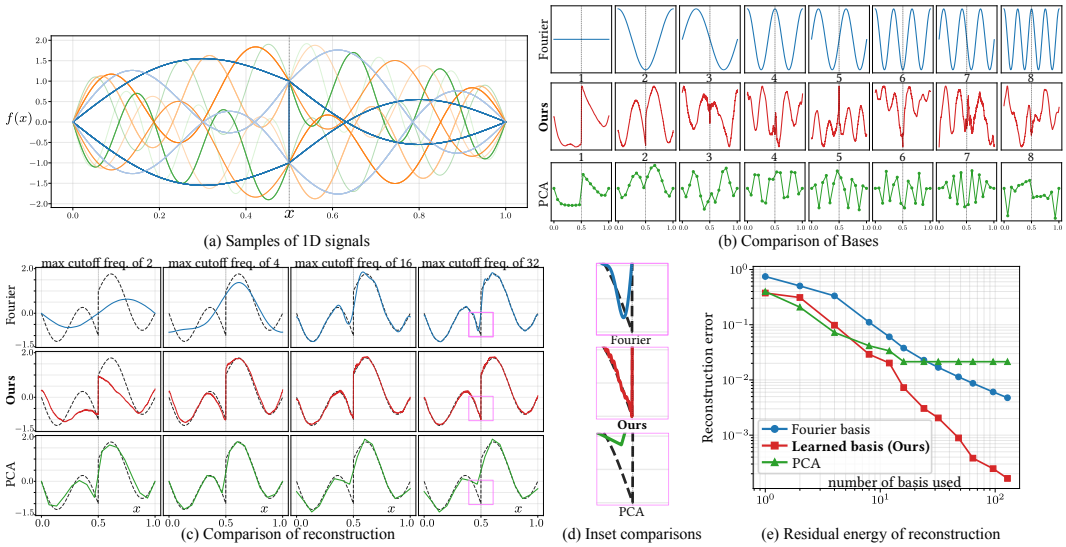


Figure 3: **1D Synthetic PCA on Function Space. (a)** Samples from a distribution of signals described by (15), exhibiting a jump at $x = 0.5$ on a 1D domain. **(b)** The Fourier basis is transformed to diagonalize the covariance operator, allowing the basis to better capture the variance of signals. Our results improve upon the traditional finite-dimensional PCA construction, which is limited by the Nyquist cutoff inherent to that setting. **(c)** Reconstruction of several sample signals from the distribution. Our learned basis yields better reconstructions across all cutoff frequencies. **(d)** In particular, finite PCA is limited by Nyquist limits and the Fourier basis exhibits the Gibbs ringing phenomenon at the discontinuity. **(e)** A quantitative comparison of reconstructions averaged across the distribution of signals. The reconstruction error is shown on a log scale. We achieve an order-of-magnitude improvement using roughly 100 basis elements.

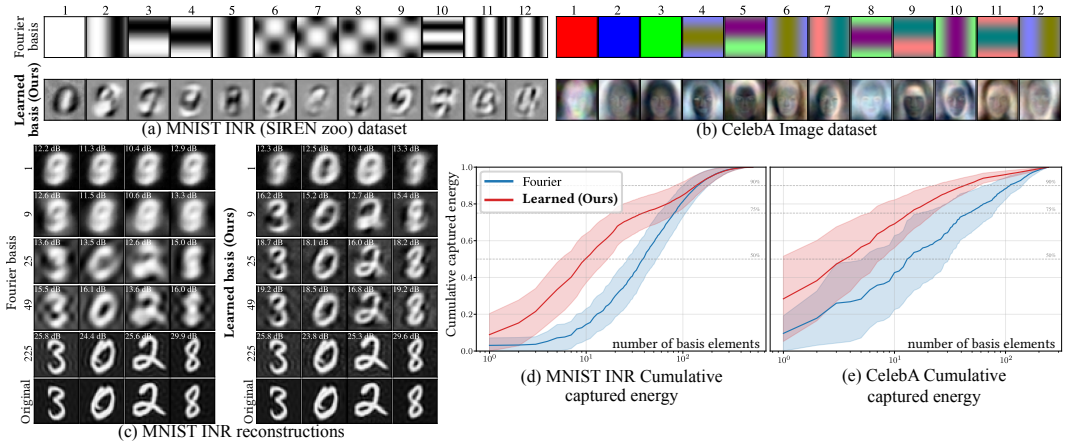
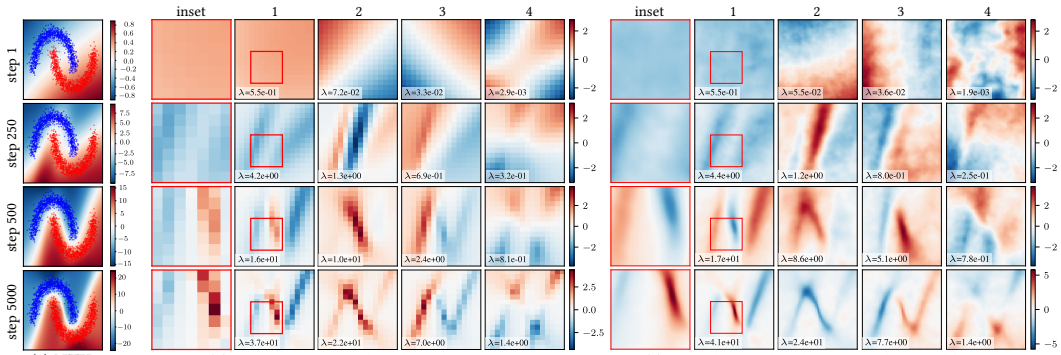


Figure 4: **2D PCA in Function Space. (a)** Fourier basis transformed to explain the INR zoo of SIREN networks representing the MNIST dataset. Note that the resulting basis represents a dataset that is itself discretization-free. **(b)** Transformed bases for the CelebA dataset. **(c)** Reconstruction of several datapoints using the Fourier basis and our transformed, optimized basis. With only a few basis elements, our basis reconstructs the signals more faithfully and captures the shape of the digits better than the Fourier baseline. **(d)** Quantitative comparison of the cumulative captured energy (mean $\pm$ std across the dataset). Our learned basis captures more energy than the Fourier baseline. Because our basis is infinite, it gradually achieves full reconstruction without a Nyquist limit; any residual error stems from numerical integration.

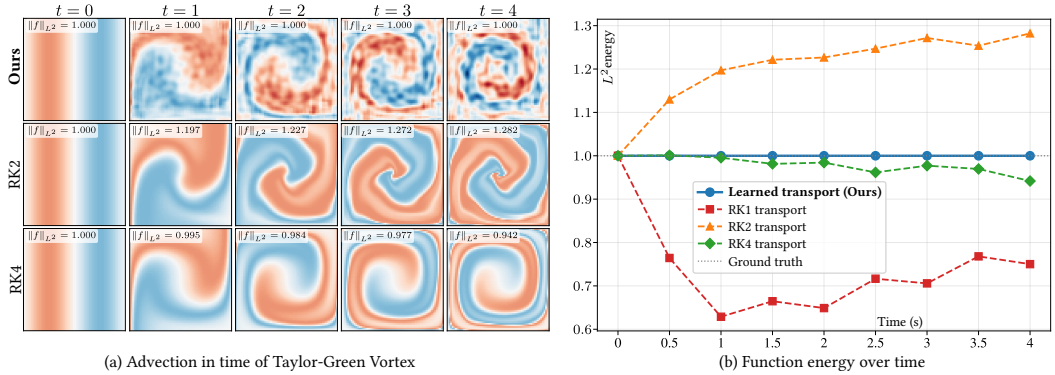
shown in Figure 3, while finite PCA is optimal for discrete data, its grid-based discretization prevents it from fitting signals beyond the Nyquist rate. Our formulation bypasses this bottleneck. Ablations over the generator rank, a comparison with a neural eigenfunction method, and computational-cost measurements are provided in Appendix D.5.

Real Data PCA. To maintain a discretization-free setting, we use the INR Zoo [44], an MNIST [35] dataset where each datapoint is a SIREN network [52] overfitted to an image. We have a similar setup for a subset of 1024 images from CelebA, using quadrature points on a 64×64 grid since INRs were unavailable. In both cases, our method learns to transform the Fourier basis into principal functions of these datasets in Figure 4. As illustrated, the learned basis achieves better reconstruction than the Fourier baseline and the first few basis elements capture most of the signal energy of the dataset.



(a) NTK (b) NTK diagonalized with grid solver (c) NTK diagonalized with **our learned basis**

Figure 5: **Two-moon NTK.** Diagonalizing the NTK over the course of training a two-moon classifier (steps 1, 250, 500, and 5000). **(a)** The evolving logit landscape and training data. **(b)** Top eigenfunctions recovered by a discrete grid-solver; the grid's fixed resolution leads to noticeable pixelation and a loss of high-frequency information. **(c)** Eigenfunctions obtained using our proposed discretization-free method. Our approach matches the global structure of the leading eigenfunctions while resolving the fine, intricate details near the decision boundary. Note that the eigenspaces are comparable up to sign which explains why the colormaps sometimes appear swapped between the two methods.



(a) Advection in time of Taylor-Green Vortex (b) Function energy over time

Figure 6: **Learning the Koopman Operator.** **(a)** Visual comparison of fluid-flow trajectories generated by our learned Koopman operator vs. the composition operator integrated via RK methods; our method is noisier but conserves energy. **(b)** L^2 energy tracking over time, comparing standard solvers (1st, 2nd, and 4th-order RK) against iterative rollouts from Q_θ.

Results on a CIFAR10 SIREN zoo, further CelebA results, and visualizations of the learned bases and of the generator dynamics are provided in Appendix D.6.

Diagonalizing the neural tangent kernel (NTK). Moving beyond PCA, we diagonalize the NTK operator at various stages of training a two-moon classifier. A short summary on what diagonalization of the NTK means is provided in Appendix B. Figure 5 compares against a grid-based eigensolver. While the discrete solver isolates eigenfunctions with lower eigenvalues slightly better, its fixed resolution fails to capture fine spatial details, particularly at the sharp center of the decision boundary. Our discretization-free method successfully resolves these localized features, highlighting a fundamental advantage of solving eigenproblems entirely in function space.

Koopman operators. We train Q_θ to approximate the Koopman operator, by optimizing the objective in (10). Here, the underlying dynamical system Ψ is the Taylor-Green vortex on a flat torus [57], simulated for a discrete time step of $\Delta t = 0.5s$. Specifically, the discrete flow map Ψ is obtained by integrating the governing ODE $\dot{x} = v(x)$ over Δt using a fourth-order Runge-Kutta (RK4) scheme, where v parameterizes the Taylor-Green vortices. After training, we evaluate the model on a sample initial condition (see Figure 6) by iteratively applying the learned operator to generate extended fluid-flow trajectory snapshots. We compare the energy conservation of these predicted rollouts against various solvers. Although baselines that run Ψ may produce more visually plausible dynamics, they fail to preserve energy, evaluated as the Hilbert norm of the signal. In contrast, despite being noisier, our learned Koopman operator yields plausible results and strictly conserves energy.

8 Conclusions, Limitations, & Future Work

In this work, we introduced a neural framework for learning continuous orthonormal bases in function spaces. Our approach allows us to optimize these bases for various tasks while preserving their essential linear-algebraic properties, such as continuity and orthonormality. The Lie ODE formulation allowed us to prove theoretical guarantees on the expressiveness of these networks. However, optimizing and backpropagating through any ODE trajectory is well-known to be practically challenging [22, 17]. Reaching our theoretical limits often demands long integration horizons, which can complicate optimization dynamics and increase training times. Moreover, our theory only concerns the expressivity of the neural family, optimization convergence analysis and finite-sample bounds remain open and important avenues to analyze the practicality of our method. This motivates future work on tailored architectures and optimizers to scale our method.

Several exciting directions remain for future research. Here, we showed how one can diagonalize the covariance and NTK operators; a natural extension is to diagonalize elliptic operators in physics or the Laplace-Beltrami operator in differential geometry. Additionally, we showed how our change-of-basis operator can match the Koopman operator, which is also an integral component of the more general family of DMD methods. It is natural to extend our method in the context of DMD with more complicated physical systems and their Koopman operators.

In conclusion, we hope that parameterizing orthonormal bases as a continuous, optimizable neural network will offer a powerful new lens for modeling complex systems involving function spaces.

Acknowledgments and Disclosure of Funding

The MIT Geometric Data Processing Group acknowledges the generous support of National Science Foundation grants IIS2335492 and OAC2403239, from the CSAIL Future of Data and FinTechAI programs, from the MIT–IBM Watson AI Laboratory, from the Wistron Corporation, from the MIT Generative AI Impact Consortium, from the Toyota–CSAIL Joint Research Center, and from Schmidt Sciences. The authors acknowledge further support from SideFX Software. The authors additionally declare that they have no competing interests.

We thank Clément Jambon, Artem Lukoianov, Christopher Scarvelis, Ana Dodik, Alice Petrov, Dolores Garcia Marti, and Ahmed Mahmoud for providing valuable feedback on our work. Hamidreza Kamkari extends a special thank you to Yeganeh Gharedaghi for her instrumental contributions during the early stages of this project and her invaluable help with ideation.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Pooya Ashtari, Pourya Behmandpoor, Nikos Deligiannis, and Aleksandra Pizurica. FUTON: Fourier tensor network for implicit neural representations. In *Advances in Neural Information Processing Systems*, 2026.

[3] Shayan K Azmoodeh, Krishna Subramani, and Paris Smaragdis. Continuous-time signal decomposition: An implicit neural generalization of pca and ica. In *2025 IEEE 35th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2025.

[4] Ido Ben-Shaul, Leah Bar, Dalia Fishelov, and Nir Sochen. Deep learning solution of the eigenvalue problem for differential operators. *Neural Computation*, 35(6):1100–1134, 2023.

[5] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems*, volume 16, 2003.

[6] Pierre Brémaud. *Mathematical principles of signal processing: Fourier and wavelet analysis*. Springer, 2002.

[7] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012. doi: 10.1063/1.4772195.

[8] Elena Celledoni, Håkon Marthinsen, and Brynjulf Owren. An introduction to lie group integrators–basics, new developments and applications. *Journal of Computational Physics*, 257: 1040–1061, 2014.

[9] Yue Chang, Otman Benchekroun, Maurizio M Chiaramonte, Peter Yichen Chen, and Eitan Grinspun. Shape space spectra. *ACM Transactions on Graphics (TOG)*, 44(4):1–16, 2025. doi: 10.1145/3731148.

[10] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[11] Richard Courant and David Hilbert. *Methods of Mathematical Physics, Vol. I*. Interscience Publishers, 1953.

[12] Luca De Luigi, Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Deep learning on implicit neural representations of shapes. In *International Conference on Learning Representations*, 2023.

[13] Zhijie Deng, Jiaxin Shi, and Jun Zhu. Neuralef: Deconstructing kernels by deep neural networks. In *International Conference on Machine Learning*, pages 4976–4992. PMLR, 2022.

[14] Fasma Diele, Luciano Lopez, and R Peluso. The cayley transform in the numerical solution of unitary differential systems. *Advances in computational mathematics*, 8(4):317–334, 1998.

[15] Emilien Dupont, Hyunjik Kim, S. M. Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5694–5725. PMLR, 2022.

[16] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[17] Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International conference on machine learning*, pages 3154–3164. PMLR, 2020.

[18] Jean Gallier. Remarks on the cayley representation of orthogonal matrices and on perturbing the diagonal of a matrix to make it invertible. *arXiv preprint math/0606320*, 2006.

[19] Ian Gemp, Brian McWilliams, Claire Vernade, and Thore Graepel. EigenGame: PCA as a Nash equilibrium. In *International Conference on Learning Representations*, 2021.

[20] Jan J Gerbrands. On the relationships between svd, klt and pca. *Pattern recognition*, 14(1-6): 375–381, 1981.

[21] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[22] Amir Gholami, Kurt Keutzer, and George Biros. ANODE: Unconditionally accurate memory-efficient gradients for neural ODEs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 730–736, 2019. doi: 10.24963/ijcai.2019/103.

[23] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[24] David Gottlieb and Steven A Orszag. *Numerical analysis of spectral methods: theory and applications*. SIAM, 1977.

[25] Brian C Hall. Lie groups, lie algebras, and representations. In *Quantum Theory for Mathematicians*, pages 333–366. Springer, 2013.

[26] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[27] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[28] Arieh Iserles. On cayley-transform methods for the discretization of lie-group equations. *Foundations of Computational Mathematics*, 1(2):129–160, 2001.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[30] Andreas M Kist. CelebA Dataset cropped with Haar-Cascade face detector, 2021. URL https://doi.org/10.5281/zenodo.5561092.

[31] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[32] Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018. doi: 10.1007/s00332-017-9423-0.

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

[34] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[35] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[36] Bruno Lévy. Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13. IEEE, 2006. doi: 10.1109/SMI.2006.21.

[37] Zongyi Li, Nikola 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.

[38] Lu Lu, Pengzhan Jin, Guofeng 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.

[39] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[40] Ana Mantilla-Dulcey, Yeganeh Gharedaghi, Daniela Quintero, Antonio Ortega, and Henry Arguello. 3D unsupervised sparse gravimetry imaging guided by a physics-consistent neural field. In *2026 IEEE International Conference on Image Processing (ICIP)*, pages 1–6, 2026. doi: 10.1109/ICIP61757.2026.11630192.

[41] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1–3):309–325, 2005. doi: 10.1007/s11071-005-2824-x.

[42] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[43] Vismay Modi, Nicholas Sharp, Or Perel, Shinjiro Sueda, and David IW Levin. Simplicits: Mesh-free, geometry-agnostic elastic simulation. *ACM Transactions on Graphics (TOG)*, 43(4):1–11, 2024.

[44] Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25790–25816. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/navon23a.html.

[45] David Pfau, Stig Petersen, Ashish Agarwal, David GT Barrett, and Kimberly L Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. In *International Conference on Learning Representations*, 2019.

[46] 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.

[47] Clarence W. Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009. doi: 10.1017/S0022112009992059.

[48] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011.

[49] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[50] Nicholas Sharp, Cristian Romero, Alec Jacobson, Etienne Vouga, Paul Kry, David IW Levin, and Justin Solomon. Data-free learning of reduced-order kinematics. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.

[51] Liyue Shen, John Pauly, and Lei Xing. NeRP: Implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):770–782, 2024. doi: 10.1109/TNNLS.2022.3177134.

[52] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[53] Henry Stark and John W Woods. *Probability, random processes, and estimation theory for engineers*. Prentice-Hall, Inc., 1986.

[54] Petre Stoica and Randolph L Moses. *Spectral Analysis of Signals*. Pearson Prentice Hall, 2005.

[55] Yu Sun, Jiaming Liu, Mingyang Xie, Brendt Wohlberg, and Ulugbek S. Kamilov. CoIL: Coordinate-based internal learning for tomographic imaging. *IEEE Transactions on Computational Imaging*, 7:1400–1412, 2021. doi: 10.1109/TCI.2021.3125564.

[56] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[57] Geoffrey Ingram Taylor and Albert Edward Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 158(895):499–521, 1937.

[58] Lloyd N Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.

[59] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014. doi: 10.3934/jcd.2014.1.391.

[60] Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional data analysis. *Annual Review of Statistics and its application*, 3:257–295, 2016.

[61] Matthew O. Williams, Clarence W. Rowley, and Ioannis G. Kevrekidis. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015. doi: 10.3934/jcd.2015005.

[62] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer graphics forum*, volume 41, pages 641–676. Wiley Online Library, 2022.

[63] Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. Signal processing for implicit neural representations. *Advances in Neural Information Processing Systems*, 35: 13404–13418, 2022.

[64] Gizem Yüce, Guillermo Ortiz-Jiménez, Beril Besbinar, and Pascal Frossard. A structured dictionary perspective on implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19228–19238, 2022.

[65] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The core contributions are clearly enumerated at the end of the Introduction. Our theoretical claim regarding the universality of finite-rank generators is formally proven in the Appendix. Meanwhile, our practical claims regarding the framework's flexibility are empirically validated across diverse tasks (functional PCA, NTK diagonalization, and fluid dynamics) in the experiments section, supported by a fully reproducible pipeline.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations of the paper are either discussed in the main body or as part of our final Conclusion section, where we explicitly address the computational challenges of long integration horizons and scaling our approach.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All necessary assumptions are clearly stated in the main text alongside the first time the theorems appear. Complete formal proofs for both our primary theoretical contribution (Theorem 1) and our secondary result (Theorem 2) are detailed in Appendix A and Appendix B, respectively.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: An overview of the main experimental results and key takeaways is provided in section 7. To ensure full reproducibility, comprehensive experimental details—including architectural choices, hyperparameters, and additional experiments—are detailed in Appendix D, and our complete codebase is also referenced in the main text.

   Guidelines:

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

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

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

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

   Answer: [Yes]

   Justification: We provide open access to all data and code. The datasets used in this code are fully public and accessible. Comprehensive implementation details and instructions to faithfully reproduce the main results are detailed in section 7, Appendix D, and our accompanying Python codebase.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [No]

   Justification: The core experimental settings and key takeaways are summarized in the main text. Exhaustive details—including all key hyperparameters—are delineated in Appendix D. For complete transparency and exact training pipelines, however, the reader should ultimately view the full codebase attached with this submission.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: For experiments reporting captured energy, our figures plot the mean values across random datapoints of the dataset, with error bars denoting one standard deviation.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The compute details are provided in Appendix D.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work focuses on foundational theory for basis learning via Lie geometry and validates these methods using standard public datasets. To the best of our knowledge, neither the proposed methodology nor the empirical experiments raise any ethical concerns or violate the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper focuses entirely on foundational mathematical and algorithmic research for representation learning in function spaces. Because our contributions are decoupled from specific, real-world deployments or societal applications, a direct discussion of societal impacts is not applicable.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper focuses on foundational mathematical frameworks and algorithmic development for learning bases for function spaces. We do not construct, train, or release any high-risk models (e.g., large language models or image generators) or sensitive scraped datasets, so specific safeguards are not applicable.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All existing assets, including public datasets and open-source software libraries utilized in our implementation, are properly cited. The corresponding licenses and terms of use are explicitly documented and respected in Appendix D.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not introduce or release any new assets, such as new datasets or pre-trained models. We rely entirely on existing, publicly available datasets for our empirical validation and train our own novel architecture models for evaluating the method.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not applicable. Our research is purely theoretical and algorithmic, and does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: Our research is entirely theoretical and algorithmic, relying solely on standard, publicly available datasets for empirical validation. It does not involve any crowdsourcing, user studies, or research with human subjects, making IRB approval and related disclosures inapplicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: Large Language Models are entirely outside the scope of our research topic and core methodology. We use them strictly as standard auxiliary tools to assist with routine drafting, formatting, and code maintenance, which does not require declaration under the current guidelines.

    Guidelines:

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

A Proof of Theorem 1

We first restate the theorem for completeness:

Theorem 1 (Approximating the Orthogonal Group via Rank-2 Generators)**.** *Let $\mathcal{Q}_T$ denote the set of all operators $Q(T)$ obtained by solving the initial value problem (5) at time $T > 0$, where the generator $K(t)$ is uniformly Hilbert-Schmidt, continuous, and rank-2. Then, $\mathcal{Q}_T$ is dense in $\mathcal{SO}(HS)$ with respect to the Hilbert-Schmidt norm topology. In other words, every rotation in this infinite orthogonal group is arbitrarily well-approximated by some time-evolving rank-2 generator.*

Here, we repeatedly use the fact that any rank-2 operator $K(t)$ can be written in the following form:

$$K(t) = \alpha(t) \otimes \beta(t) - \beta(t) \otimes \alpha(t), \tag{16}$$

where $\alpha, \beta : [0, T] \to H$ are H-valued functions of time. Our goal is to ultimately show that for any operator $Q \in \mathcal{SO}(HS)$, there exist α, β such that the solution to the following initial value problem closely approximates Q at $t = T$:

$$\dot{Q}(t) = [\alpha(t) \otimes \beta(t) - \beta(t) \otimes \alpha(t)]\, Q(t), \qquad Q(0) = I. \tag{17}$$

To do so, we first restrict our space to the following space of operators:

Definition 2. Let $\mathcal{SO}(\text{fin}) \subset \mathcal{SO}(HS)$ denote the group of "finite-like" orthogonal Hilbert Schmidt operators that act as identity outside of some finite-dimensional linear subspace of V of H. That is, if $Q \in \mathcal{SO}(\text{fin})$ then there exists some finite n and a finite set $S = \{e_1, \ldots, e_n\} \subset H$ such that

$$Qv = v \quad \text{for all } v \in V^\perp = \text{span}\{e_1, \ldots, e_n\}^\perp.$$

We now constructively prove that the approximation theorem holds for $\mathcal{SO}(\text{fin})$:

Lemma 3. *For any $Q \in \mathcal{SO}(\text{fin})$, there exists continuous maps $\alpha, \beta : [0, T] \times \Omega \to \mathbb{R}$ such that the solution $Q(T)$ of (17) satisfies $Q(T) = Q$.*

Proof. Let $Q \in \mathcal{SO}(\text{fin})$. Then, by definition, there exists a finite-dimensional subspace V alongside its orthonormal basis $\{e_1, \ldots, e_n\} \subset H$ such that Q acts trivially on $V^\perp$. We can therefore restrict the representation of Q to V and get a finite representation $Q_V \in SO(n)$. By classical results in numerical linear algebra and Lie group theory [25], any element of $SO(n)$ can be written as a finite product of Givens' rotations acting on coordinate planes.

In other words, for any Q_V there exist M angles $\{\theta_m\}_{m=1}^M$ and pairwise orthogonal directions $\{a_m\}_{m=1}^M, \{b_m\}_{m=1}^M \subset \mathbb{S}^{n-1}$ such that:

$$Q_V = \prod_{m=1}^{M} R^{\text{Givens}}_{a_m, b_m}(\theta_m) = \prod_{m=1}^{M} \exp\Big(\theta_m (a_m b_m^\top - b_m a_m^\top)\Big). \tag{18}$$

We can now define $\alpha_m = \sqrt{\theta_m} \cdot \sum_{i=1}^n a_{mi} e_i \in H$ and $\beta_m = \sqrt{\theta_m} \cdot \sum_{i=1}^n b_{mi} e_i \in H$ which allows us to lift the representation back from the coefficient space to H and write:

$$Q = \prod_{m=1}^{M} \exp\Big(\alpha_m \otimes \beta_m - \beta_m \otimes \alpha_m\Big). \tag{19}$$

Now, the idea is to construct a continuous field in H, that once integrated, results in the operator above. We achieve this by stitching together M time-evolving functions below:

$$\alpha_m(t) = r_m(t)\, \alpha_m, \qquad \beta_m(t) = r_m(t)\, \beta_m. \tag{20}$$

We define $r_m : \mathbb{R} \to \mathbb{R}$ as a continuous control function in the form of a "bump" that has the following properties: $r_m(0) = r_m(1) = 0$ and $\int_0^1 r_m^2(\tau) d\tau = 1$. One such control function can be the square root of the probability density of the Beta distribution.

These properties on $r_m(t)$ ensure $\alpha_m(1) = \alpha_{m+1}(0)$ and $\beta_m(1) = \beta_{m+1}(0)$, allowing us to stitch generators continuously and form a bigger generator. Furthermore, the construction above allows the self-adjoint operator to always self-commute over all $t \in [0, 1]$:

$$\alpha_m(t) \otimes \beta_m(t) - \beta_m(t) \otimes \alpha_m(t) = r_m^2(t)\, [\alpha_m \otimes \beta_m - \beta_m \otimes \alpha_m] \tag{21}$$

Next, w.l.o.g. assume $T = M$ and define the following M initial value problems:

$$\dot{Q}_m = [\alpha_m(t) \otimes \beta_m(t) - \beta_m(t) \otimes \alpha_m(t)]\, Q_m(t), \quad Q_m(0) = \begin{cases} Q_{m-1}(1) & \text{if } m > 1 \\ I & \text{otherwise} \end{cases}. \tag{22}$$

Using the fact that the generator above is skew-adjoint and self-commutes, we can write the solution of the ODE as an exponential map and simplify:

$$Q_m(1) = \exp\left([\alpha_m \otimes \beta_m - \beta_m \otimes \alpha_m] \cdot \underbrace{\int_0^1 r_m^2(\tau) d\tau}_{(=1) \text{ by construction}} \cdot \right) Q_m(0) \tag{23}$$

$$\Rightarrow \quad Q_M(1) = \prod_{m=1}^{M} \exp\left(\alpha_m \otimes \beta_m - \beta_m \otimes \alpha_m\right) \tag{24}$$

Hence, chaining them yields a time-evolving $Q(t)$ for the form (17) where $Q(T) = Q_M(1) = Q$. □

Lemma 4. *$\mathcal{SO}$(fin) is dense in $\mathcal{SO}(HS)$ with respect to the Hilbert-Schmidt operator norm.*

Proof. Let $Q \in \mathcal{SO}(HS)$, then from the definition, it can be written as the exponential map of a Hilbert-Schmidt skew-adjoint operator, i.e., $Q = \exp(K)$ for some $K = -K^*$ where $\|K\|_{HS} < \infty$.

Since K is Hilbert-Schmidt, using the spectral theorem for operators, there exists a sequence of real values $\lambda_n > 0$ and an orthonormal set of vectors in H, $\{u_n, v_n\}_{n=1}^{\infty}$, such that:

$$K = \sum_{j=1}^{\infty} \lambda_j (v_j \otimes u_j - u_j \otimes v_j). \tag{25}$$

Next, we can define the sequence of finite rank operators $\{K_n\}_{n=1}^{\infty}$ as the spectral truncation:

$$K_n = \sum_{j=1}^{n} \lambda_j (v_j \otimes u_j - u_j \otimes v_j). \tag{26}$$

Importantly, $\lim_{n\to\infty} K_n = K$ in $\|\cdot\|_{HS}$ and K commutes with all K_n: $KK_n = K_nK$.

Since K_n is finite rank, its exponential acts as identity on range$(K_n)^{\perp}$; this means that $Q_n := \exp(K_n) \in \mathcal{SO}(\text{fin})$. Furthermore, we have that:

$$\lim_{n\to\infty} \|Q_n - Q\|_{HS} \tag{27}$$

$$= \lim_{n\to\infty} \|\exp(K_n) - \exp(K)\|_{HS} \tag{28}$$

$$= \lim_{n\to\infty} \|\exp(K)\left[\exp(K_n - K) - I\right]\|_{HS} \qquad \text{Exponentiation of Commutative Operators} \tag{29}$$

$$\leq \|\exp(K)\|_{HS} \cdot \|\exp\left(\lim_{n\to\infty} K_n - K\right) - I\|_{HS} \qquad \text{Continuity of norms and exponentials} \tag{30}$$

$$= \|\exp(K)\|_{HS} \cdot \|\exp(\mathbf{0}) - I\|_{HS} = 0. \tag{31}$$

Thus, we have found for any $Q \in \mathcal{SO}(HS)$, a sequence of operators $Q_n \in \mathcal{SO}(\text{fin})$ such that $Q_n \overset{n\to\infty}{\Longrightarrow} Q$ in $\|\cdot\|_{HS}$; this is precisely the definition of $\mathcal{SO}(\text{fin})$ being dense in $\mathcal{SO}(HS)$. □

Collectively, the two lemmas above prove Theorem 1.

Finally, a universal approximation result with MLPs is straightforward: with a sufficiently overparameterized θ, any two MLPs can approximate both functions α and β on the compact domain $[0, T] \times \Omega$ [27]. In turn, by approximating the continuous $\alpha : [0, T] \times \Omega \to \mathbb{R}$ and $\beta : [0, T] \times \Omega \to \mathbb{R}$, these MLPs implicitly approximate an arbitrary $Q \in \mathcal{SO}(HS)$.

B Diagonalizing Linear Operators

B.1 Proof of Theorem 2

Theorem 2. *Let A be a trace-class positive semidefinite linear operator, and let $\{p(i)\}_{i=1}^{\infty}$ be a strictly decreasing sequence of positive weights such that $\sum_{i=1}^{\infty} p(i) = 1$, and $\{\varphi_i\}_{i=1}^{\infty}$ be any orthonormal basis on H. Consider Q^{opt} as the solution to the following variational problem:*

$$\max_{Q \in \mathcal{SO}(HS)} \quad \sum_{i=1} p(i) \cdot \langle Q\varphi_i, AQ\varphi_i \rangle \tag{7}$$

Then, $\phi_i^\star := Q^{opt}\varphi_i$ are the eigenfunctions of A. That is, $A\phi_i^\star = \lambda_i \phi_i^\star$ for all $i \geq 1$, where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$ are the corresponding eigenvalues.

Proof. Let Υ and Λ be diagonal operators with entries $p(i)$ and λ_i, respectively, and let $\Phi \in \mathcal{SO}(\infty)$ be the orthogonal operator whose i-th column corresponds to $Q\varphi_i$. We can compactly rewrite the objective as $\mathrm{tr}(\Upsilon\Phi^* A\Phi)$. Let $\sigma_i(\cdot)$ an operator functional that inputs an operator and returns its ith largest eigenvalue. Since both Υ and A are trace-class, by applying von Neumann's trace inequality and the Poincaré separation theorem, respectively, we upper-bound the objective as follows:

$$\mathrm{tr}(\Upsilon\Phi^* A\Phi) \leq \sum_{i=1}^{\infty} p(i)\sigma_i(\Phi^* A\Phi) \qquad \text{Von-Neumann inequality for trace-class operators} \tag{32}$$

$$\leq \sum_{i=1}^{\infty} p(i)\sigma_i(A) \qquad \Phi \text{ is unitary, so: } \sigma_i(\Phi^* A\Phi) \leq \sigma_i(A) \tag{33}$$

$$= \sum_{i=1}^{\infty} p(i)\lambda_i. \tag{34}$$

The upper bound is exactly achieved when $\Phi \in \mathcal{SO}(HS)$ diagonalizes A. □

B.2 Diagonalizing the Covariance Operator

An important detail omitted from the main text is that, in practice, many datasets $X \sim \mathbb{P}$ used for principal component analysis are not necessarily zero-mean. To properly diagonalize the true covariance operator $\mathcal{C}$ using Q_θ, we must shift the data by its empirical mean. Rather than computing the empirical mean beforehand, we accomplish the same effect by concurrently fitting a spatial mean function—parameterized as a neural field $\nu_\psi : \Omega \to \mathbb{R}$ using random Fourier features [56].

This leads to a joint objective where we maximize the captured energy of the centered data while minimizing the L^2 reconstruction error of the mean:

$$\max_{\theta} \quad \mathbb{E}_{i\sim p, X\sim\mathbb{P}} \langle X - \nu_\psi, Q_\theta \varphi_i \rangle^2, \tag{35}$$

$$\min_{\psi} \quad \mathbb{E}_{X\sim\mathbb{P}} \|\nu_\psi - X\|^2. \tag{36}$$

Because the parameters ψ and θ are decoupled across these two objectives, we can optimize them in parallel using independent optimizers for each task.

For clarity, the algorithm is summarized as a pseudo-code in algorithm 1.

B.3 Diagonalizing the Neural Tangent Kernel

Let $\mathrm{MLP}(\cdot;\xi) : \Omega \to \mathbb{R}$ a non-linear neural function parameterized by ξ. We restrict our attention to networks with a scalar output; this encompasses the experimental setting in section 7 like our Two Moons experiment where the output of the network represents the logit of a binary classifier.

The Neural Tangent Kernel (NTK) is defined by the inner product of the network's parameter gradients. We can express this as an integral operator κ_ξ acting on the function space H with the kernel $\kappa_\xi[x,y] = \nabla_\xi \mathrm{MLP}(x;\xi)^\top \nabla_\xi \mathrm{MLP}(y;\xi)$. Specifically, the action of the NTK operator on any function $f \in H$ is defined as:

$$(\kappa_\xi f)(x) = \int_\Omega \nabla_\xi \mathrm{MLP}(x;\xi)^\top \nabla_\xi \mathrm{MLP}(y;\xi) f(y)\, \mu(dy). \tag{37}$$

Algorithm 1: Continuous Principal Component Analysis

Input: Dataset $\mathcal{X}$, reference basis $\{\varphi_i\}_{i=1}$, distribution $p(i)$, domain Ω with measure μ.
Output: Optimized basis Q_θ and mean function ν_ψ.
Initialize orthogonal network parameters θ and mean function parameters ψ;
while *not converged* **do**
  Sample a function $X \sim \mathcal{X}$, a basis index $i \sim p$, and quadrature points $\widehat{\Omega} \sim \mu$;
  $\mathcal{L}_{\text{mean}} \leftarrow \frac{1}{|\widehat{\Omega}|} \sum_{\omega \in \widehat{\Omega}} \|\nu_\psi(\omega) - X(\omega)\|^2$;
  $\mathcal{J}_{\text{PCA}} \leftarrow \frac{1}{|\widehat{\Omega}|} \sum_{\omega \in \widehat{\Omega}} \langle X(\omega) - \nu_\psi(\omega), (Q_\theta \varphi_i)(\omega) \rangle^2$;
  $\psi \leftarrow \text{update}(\psi, \nabla_\xi \mathcal{L}_{\text{mean}})$;
  $\theta \leftarrow \text{update}(\theta, \nabla_\theta \mathcal{J}_{\text{PCA}})$;

To diagonalize the NTK using our framework, we substitute κ_ξ for A in the variational objective of Theorem 2. For a basis element $\phi_i = Q\varphi_i$, we can simplify the quadratic form $\langle \phi_i, \kappa_\xi \phi_i \rangle$:

$$\langle \phi_i, \kappa_\xi \phi_i \rangle \tag{38}$$

$$= \int_\Omega \phi_i(x) \left(\int_\Omega \nabla_\xi \text{MLP}(x;\xi)^\top \nabla_\xi \text{MLP}(y;\xi) \phi_i(y)\, \mu(dy) \right) \mu(dx) \tag{39}$$

$$= \int_\Omega \int_\Omega \left[\phi_i(x) \cdot \nabla_\xi \text{MLP}(x;\xi)\right]^\top \left[\nabla_\xi \text{MLP}(y;\xi) \cdot \phi_i(y)\right] \mu(dy)\mu(dx) \quad \text{(Fubini)} \tag{40}$$

$$= \| \mathbb{E}_{x\sim\mu} \left[\phi_i(x) \nabla_\xi \text{MLP}(y;\xi)\right] \|_2^2 \quad \text{pulling pack into the parameter space} \tag{41}$$

Thus, the quadratic form is in fact equal to the squared norm of the expected parameter gradient. Substituting this simplified quadratic form back into the general objective, we arrive at our specialized loss function for diagonalizing the NTK:

$$\max_\theta \ \mathbb{E}_{i\sim p} \| \mathbb{E}_{x\sim\mu} \left[(Q_\theta \varphi_i)(x) \nabla_\xi \text{MLP}(x;\xi)\right] \|_2^2 . \tag{42}$$

In practice, the optimization proceeds by sampling a target index $i \sim p$ and a batch of quadrature points $\widehat{\Omega} \sim \mu$. At these points, we evaluate both our trial basis function $(Q_\theta \varphi_i)(x)$ and the per-sample parameter gradients $\nabla_\xi \text{MLP}(x;\xi)$. We then compute the empirical expectation over the batch by taking the weighted average of these evaluated gradients, yielding a single vector in the network's parameter space. Finally, we compute the squared norm of this expectation and backpropagate to update the basis parameters θ, keeping the MLP parameters ξ frozen. Upon convergence, the learned functions $\phi_i = Q_\theta \varphi_i$ faithfully represent the dominant eigenmodes of the training dynamics.

For clarity, the algorithm is summarized as a pseudo-code in algorithm 2.

Algorithm 2: Diagonalizing the Neural Tangent Kernel

Input: Domain Ω with measure μ, reference basis $\{\varphi_i\}_{i=1}$, distribution $p(i)$, frozen $\text{MLP}(\cdot;\xi)$.
Output: Optimized NTK eigenfunctions $\phi_i = Q_\theta \varphi_i$.
Initialize orthogonal network parameters θ;
while *not converged* **do**
  Sample a batch of quadrature points $\widehat{\Omega} \sim \mu$ **and** a basis index $i \sim p$;
  `// Step 1: Evaluate trial eigenfunctions and parameter gradients`
  for *each* $\omega \in \widehat{\Omega}$ **do**
    $g(\omega) \leftarrow \nabla_\xi \text{MLP}(\omega;\xi)$;
    $q(\omega) \leftarrow (Q_\theta \varphi_i)(\omega)$;
  `// Step 2: Compute empirical expected gradient and NTK objective`
  $v \leftarrow \frac{1}{|\widehat{\Omega}|} \sum_{\omega \in \widehat{\Omega}} q(\omega) g(\omega)$;
  $\mathcal{J}_{\text{NTK}} \leftarrow \|v\|_2^2$;
  `// Step 3: Update basis generator`
  $\theta \leftarrow \text{update}(\theta, \nabla_\theta \mathcal{J}_{\text{NTK}})$;

C Cayley Integration Details and Multichannel Details

This appendix expands on section 6. We provide a pseudo-code for the Cayley integration in algorithm 3; this method is internally implemented whenever $Q_\theta \varphi_i(\omega)$ is called for $\omega \in \widehat{\Omega} \subset \Omega$. Notably, we set $T = 1$ for all experiments.

We also extend the Cayley integration to multichannel function spaces and provide more details on the forward and backward Euler ODE solvers used in our calculations in Figure 2.

Algorithm 3: Continuous Change-of-Basis via Cayley Integration

Input: Initial basis function φ_i, parameters θ, rank r.
Output: Transformed basis evaluation $\widehat{\phi}_i^\theta(T) = Q_\theta \varphi_i$.
Sample random time discretization $0 = t_0 < t_1 < \cdots < t_L = T$;
Sample random dense spatial samples $\widehat{\Omega} = \{\omega_j\}_{j=1}^D \sim \mu$;
Initialize $\widehat{\phi} \leftarrow \varphi_i(\widehat{\Omega})$;
for $\ell = 1, \ldots, L$ **do**
  $\Delta t_\ell \leftarrow t_\ell - t_{\ell-1}$ **and** $t_{\text{mid}} \leftarrow t_{\ell-1} + \Delta t_\ell / 2$;
  $\widehat{U} \leftarrow \sqrt{\Delta t_\ell}\, U_\theta(t_{\text{mid}}, \widehat{\Omega}) \in \mathbb{R}^{r \times D}$ **and** $\widehat{S} \leftarrow M_\theta(t_{\text{mid}}) - M_\theta(t_{\text{mid}})^\top \in \mathbb{R}^{r \times r}$;
  `// Step 1: Estimate` $r \times r$ `Gram matrix and Woodbury inverse` M
  $G \leftarrow [\widehat{U}\widehat{U}^*]_{i,j} \approx \frac{1}{D} \sum_{\omega \in \widehat{\Omega}} \widehat{U}^i(\omega) \widehat{U}^j(\omega) \in \mathbb{R}^{r \times r}$ **and** $M \leftarrow \left(I_{r \times r} - \frac{1}{2} G\, \widehat{S} \right)^{-1}$;
  `// Step 2: Apply Forward Action` $\phi \mapsto \phi + \frac{1}{2} \widehat{U}^* \widehat{S} \widehat{U} \phi$
  $c \leftarrow [\widehat{U}\phi]_r \approx \frac{1}{D} \sum_{\omega \in \widehat{\Omega}} \widehat{U}^r(\omega) \phi(\omega) \in \mathbb{R}^r$;
  $y(\widehat{\Omega}) \leftarrow \phi(\widehat{\Omega}) + \frac{1}{2} \widehat{U}^*\, \widehat{S}\, c$;
  `// Step 3: Apply Backward Action via Woodbury`
  $\tilde{c} \leftarrow [\widehat{U} y]_r \approx \frac{1}{D} \sum_{\omega \in \widehat{\Omega}} \widehat{U}^r(\omega) y(\omega) \in \mathbb{R}^r$;
  $w \leftarrow \widehat{S} M\, \tilde{c} \in \mathbb{R}^r$;
  $\phi(\widehat{\Omega}) \leftarrow y(\widehat{\Omega}) + \frac{1}{2} \widehat{U}^*\, w$;

C.1 Multichannel Hilbert Spaces

The multichannel-valued Hilbert space $H_C = L^2(\Omega, \mathbb{R}^C)$, is defined through the following inner product that integrates over the spatial domain and sums across channels:

$$\langle f, g \rangle = \sum_{c=1}^{C} \int_\Omega f_c(x) g_c(x)\, d\mu(x). \tag{43}$$

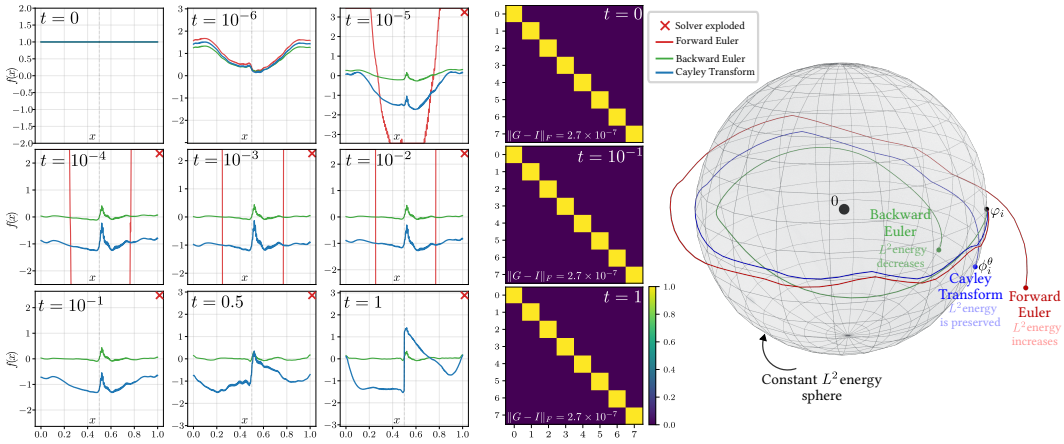
Consequently, H_C is a single unified Hilbert space rather than C independent ones. Naturally, when $C = 1$, H_C simplifies to the standard real-valued Hilbert space denoted as H in the main text.

For experiments involving multichannel data (e.g. RGB images in CelebA), we only need a minor adjustment to U_θ. Instead of the single-channel formulation $U_\theta(t, \omega) \in \mathbb{R}^r$, we use $U_\theta(t, \omega) \in \mathbb{R}^{rC}$ and index by $U_\theta(t, \omega)_c^i$ to denote the cth channel of the ith column of U_θ. Inner product approximations are updated to account for the channel dimension, meaning the Gram matrix and matrix-vector products become:

$$[U_\theta(t) U_\theta(t)^*]_{ij} \approx \frac{1}{|\widehat{\Omega}|} \sum_{\substack{\omega \in \widehat{\Omega} \\ 1 \le c \le C}} U_\theta(t, \omega)_c^i \cdot U_\theta(t, \omega)_c^j, \tag{44}$$

$$[U_\theta(t) f]_i \approx \frac{1}{|\widehat{\Omega}|} \sum_{\substack{\omega \in \widehat{\Omega} \\ 1 \le c \le C}} U_\theta(t, \omega)_c^i f_c(\omega). \tag{45}$$

Applying generators independently per channel forces a block-diagonal structure that isolates the color channels, preventing the transfer of signal energy between them. By instead operating on the



(a) Forward integration of constant Fourier basis (b) Gram matrix during evolution (c) Cartoon of different time integrations

Figure 7: **Full Cayley Ablation.** Comparison of the Cayley method against backward and forward Euler. **(a)** Example evolution of a basis function $\phi_i^\theta(t)$ over time. The Cayley solver perfectly maintains signal energy, while forward Euler rapidly explodes and backward Euler dissipates. **(b)** Beyond norm preservation, the inner products of various basis functions, visualized as a gram matrix, also remain preserved through various snapshots of time. **(c)** A conceptual diagram of the Lie group integrator, which strictly remains on the orthogonal manifold, compared to methods that either overshoot or decay. Overall, Cayley integration is critical for stable optimization.

unified multichannel Hilbert space H_C, our formulation allows for cross-channel interactions and color mixing. Empirically, this is what allows us to map a blue Fourier basis element to a multicolored function, as shown in Figure 2. Moreover, since H_C itself is a proper Hilbert space, the approximation theorem Theorem 1 also holds in the multichannel case.

C.2 Cayley vs. Off-the-Shelf ODE Solvers

Figure 7 compares our Cayley integrator with two classical alternatives, holding the number of steps L and the skew-symmetric operator $\widehat{K}_\ell$ constant:

- **Forward (explicit) Euler**: The update formula can be written as $\widehat{\phi}_\ell = (I + \Delta t\, \widehat{K}_\ell)\widehat{\phi}_{\ell-1}$. For a skew-symmetric $\widehat{K}_\ell$, this is unstable as each step amplifies the norm by $\sqrt{1 + (\Delta t\, \sigma_K)^2} > 1$, where σ_K represents the singular values of $\widehat{K}_\ell$. This leads to the explosive growth shown in Figure 2 and Figure 7.
- **Backward (implicit) Euler**: The update rule can be written as $\widehat{\phi}_\ell = (I - \Delta t\, \widehat{K}_\ell)^{-1}\widehat{\phi}_{\ell-1}$. This approach is stable but dissipative, with the norm decaying by $1/\sqrt{1 + (\Delta t\, \sigma_K)^2} < 1$ at each step. For a fair comparison, we implement this using the same Woodbury identity as our Cayley solver.
- **Cayley (ours)**: Norm-preserving, ensuring $\|\widehat{\phi}_\ell\| = 1$ at every integration step.

D Experimental Details

This appendix collects the configurations used for every experiment in section 7. All runs share the same Cayley integrator (Appendix C) and differ only in their variational objective (PCA, NTK diagonalisation, or Koopman fitting). Table 1 also summarized some key hyperparameters that are different across the different experiments.

D.1 Choice of Reference Basis

All experiments in the main text use the Fourier basis as the reference. Since Theorem 1 concerns only the change of basis Q_θ, any orthonormal reference can be used instead. In Figure 8 we replace the initial basis with the Haar wavelet basis on the 1D FPCA problem.

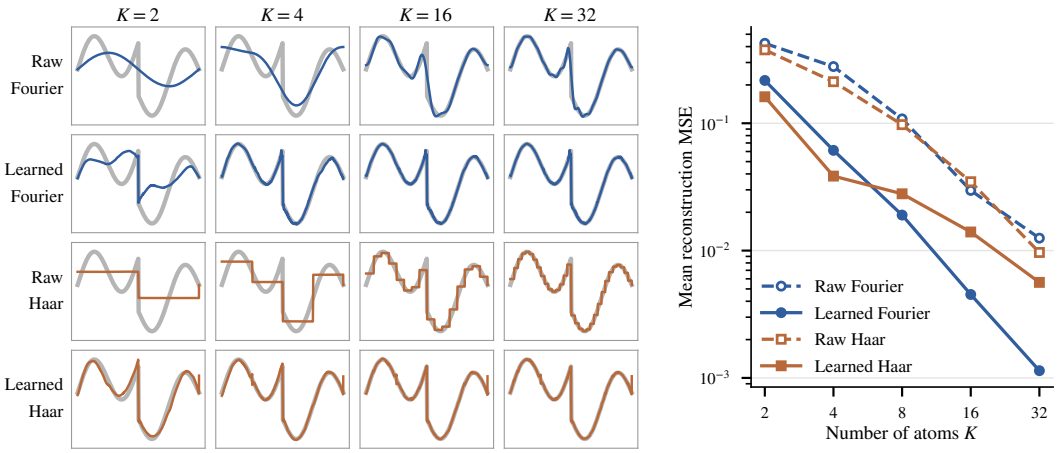


Figure 8: **Different reference bases comparison. Left:** one held-out signal (grey) reconstructed from its first K atoms under the raw and learned Fourier and Haar bases. **Right:** reconstruction MSE averaged over 2,048 held-out signals. Learning improves both references at every truncation level; from eight atoms on, the learned Fourier basis is clearly better, suggesting that the discontinuous Haar elements are harder to transport.

D.2 Indexing the Infinite Bases and their Prior $p(i)$

All variational problems involve a real Fourier basis φ_i defined on $\Omega = [0,1]^d$ and an index prior $p(i)$. The index i is represented as a vector of size d, $\mathbf{i} = [i_1, i_2, \ldots, i_d]$, for single-channel tasks, and extended to size $d+1$, $\mathbf{i} = [i_1, i_2, \ldots, i_d, c]$, for multi-channel tasks. For each spatial dimension j, the absolute value $|i_j|$ denotes the Fourier frequency. The sign of i_j determines the trigonometric function: $i_j > 0$ yields a cosine, $i_j < 0$ yields a sine, and $i_j = 0$ yields the constant function.

For example, in a 2D single-channel setting, the index $i = [2, -1]$ corresponds to the scalar basis function $\varphi_i(x_1, x_2) \propto \cos(2 \times 2\pi x_1) \sin(1 \times 2\pi x_2)$. In a multi-channel setting (e.g., a 3-channel RGB image), the extended index $i = [2, -1, 3]$ applies this exact same spatial feature strictly to the third channel. This yields the vector-valued basis function $\varphi_i(x_1, x_2) \propto [0, \quad 0, \quad \cos(2 \times 2\pi x_1) \sin(1 \times 2\pi x_2)]$.

We quantify the overall spatial frequency of a Fourier basis element using the squared ℓ^2-norm of its spatial multi-index $\mathbf{i} = [i_1, \ldots, i_d]$, given by $\|\mathbf{i}\|_2^2 = \sum_{j=1}^d i_j^2$. To prioritize lower-frequency features, we define the index prior p as a strictly decreasing power law over this norm:

$$p(i_1, \ldots, i_d, c) = \frac{1}{C\, Z_\alpha} \left(1 + \|\mathbf{i}\|_2^2\right)^{-\alpha}, \qquad Z_\alpha := \sum_{\mathbf{k} \in \mathbb{Z}^d} \left(1 + \|\mathbf{k}\|_2^2\right)^{-\alpha}, \tag{46}$$

where C is the total channel count and $\alpha > d/2$ ensures the partition function Z_α is finite.

Objective Variance Reduction. Naïve Monte-Carlo estimation of variational objectives involving the expectation $\mathbb{E}_{\mathbf{i} \sim p}[\cdot]$ can suffer from high variance. To mitigate this, we use a *stratified-with-tail*

estimator that decomposes the expectation into two parts: an *exact stratum* $S = \{\mathbf{i} : p(\mathbf{i}) \geq \tau\}$ comprising all high-probability indices, and a *Monte-Carlo tail* consisting of N_τ independent and identically distributed (i.i.d.) draws from p. Concretely, for an integrand f, the estimator is

$$\widehat{\mathbb{E}}[f] = \underbrace{\sum_{\mathbf{i}\in S} p(\mathbf{i})\, f(\mathbf{i})}_{\text{exact stratum}} + \underbrace{\frac{1}{N_\tau}\sum_{k=1}^{N_\tau} f(\mathbf{i}^{(k)})\, \mathbf{1}[\mathbf{i}^{(k)} \notin S]}_{\text{Monte-Carlo tail}}, \qquad \mathbf{i}^{(k)} \overset{\text{i.i.d.}}{\sim} p, \tag{47}$$

where $\mathbf{1}(\cdot)$ is the indicator function of an event. By construction, this estimator is unbiased: the exact stratum is evaluated with zero variance, while the variance of the sampled tail scales as $\mathcal{O}(1/N_\tau)$. The probability threshold τ and the tail sample size N_τ for each experiment are detailed in Table 1.

Table 1: Hyperparameters for every experiment shown in section 7. Rank r, time-discretisation L, and quadrature size D are the parameters that control accuracy of the function-space ODE; τ is the tail cutoff and N_τ the tail size for the stratified energy estimator.

Experiment	r	L	D	τ	N_τ
1-D PCA (Figure 3)	30	20	128	10^{-3}	16
INR-Zoo MNIST (Figure 4)	20	20	128^2	10^{-4}	64
CelebA (Figure 4)	10	20	64^2	10^{-5}	32
NTK two-moons (Figure 5)	10	20	32^2	7×10^{-4}	32
Fluid Vortices (Figure 6)	10	50	32^2	10^{-4}	32

D.3 Architecture

Every experiment parameterizes the generator in (13) with two small networks that share a sinusoidal time embedding $\gamma(t) \in \mathbb{R}^{512}$ with $F = 256$ frequencies log-spaced in $[2^0, 2^3]$:

$$\gamma(t) = [\sin(2\pi\nu_1 t), \cos(2\pi\nu_1 t), \ldots, \sin(2\pi\nu_F t), \cos(2\pi\nu_F t)]\,, \quad \nu_f = 2^{3(f-1)/(F-1)}. \tag{48}$$

Mixing matrix $M_\theta(t)$. An MLP maps $\gamma(t)$ to $\mathbb{R}^{r\times r}$, with two hidden layers of width 64, SiLU activations [16] and RMSNorm [65]. Since $A_\theta(t) = M_\theta(t) - M_\theta(t)^\top$ is easily skew-symmetrized the network output can be fully unconstrained. That said, for stability, $A_\theta(t)$ is rescaled in the forward pass to control its Frobenius norm.

Basis field $U_\theta(t, \omega)$. The r functions of $U_\theta(t)$ come from a single NeRF-style MLP [42] applied to $[\gamma(t), \omega, \eta(\omega)]$, where $\eta(\omega)$ are random Fourier features [56] at 8 scales log-spaced in $[1, 64]$, with $\max(1, 64/2^\ell)$ Gaussian projections at scale ℓ (256 sine and cosine features in total). The MLP has three bias-free hidden layers of width 256 with SiLU and RMSNorm, and outputs $r \times C$ values for a C-channel signal. Before each Cayley step, every row of $\widehat{U}_\ell$ is normalized to unit norm under the quadrature, so the $r \times r$ Gram matrix in the solver has a unit diagonal.

D.4 Compute and Optimization

All experiments run on a single NVIDIA RTX 5090 (32 GB) using the Adam optimizer [29] with default hyperparameters, gradient clipping at $\|\nabla\|_2 \leq 1$, and gradient checkpointing through the Cayley step ladder to keep VRAM constant in L. Each Cayley step costs $\mathcal{O}(r^3 + r^2 D)$; for $D \leq 128^2$ and $10 \leq r \leq 30$ this is well below the cost of a single MLP forward pass through U_θ, so wall-clock is dominated by the L network evaluations of $U_\theta(t_\ell, \widehat{\Omega})$ per training step.

D.5 FPCA Ablations and Comparisons

On the 1D FPCA problem of (15), we ablate the generator rank and the cost of integration (Figure 9) and compare with a NeuralEF-style baseline [13] that fits eight MLPs to the leading eigenfunctions of the covariance operator (Figure 10). All errors are held-out reconstruction MSEs on 2,048 signals.

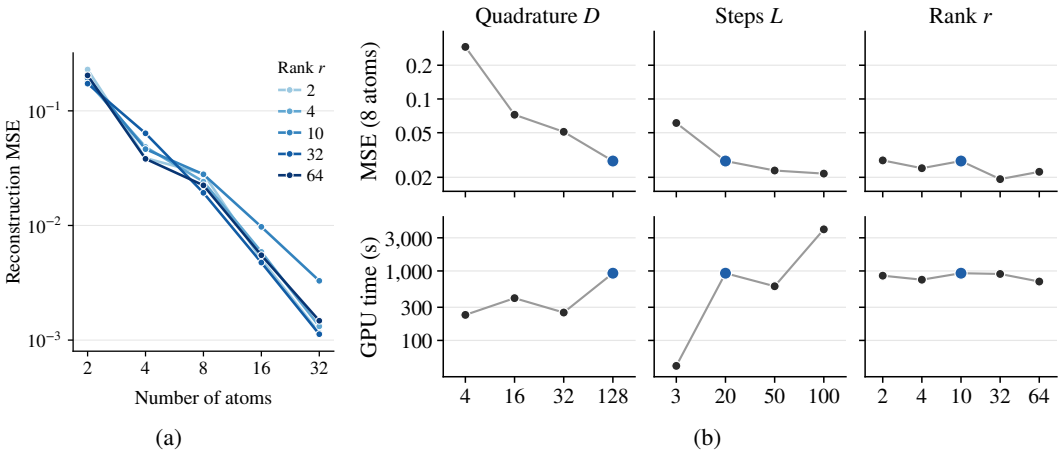
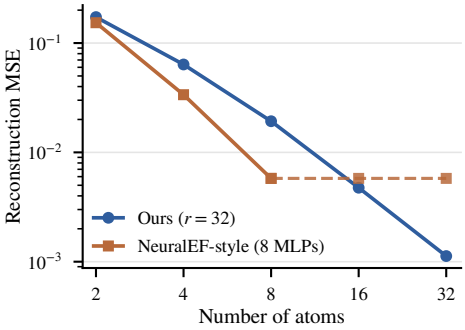


Figure 9: **Generator rank and integration cost on 1D FPCA. (a)** Error versus the number of retained atoms; darker lines are larger ranks. Rank 2, the case covered by Theorem 1, already reconstructs well; $r = 32$ is best at most truncation levels, but as we increase rank more, we see diminishing returns. **(b)** Varying one parameter at a time around $(D, L, r) = (128, 20, 10)$ (blue). Top: error with eight atoms. Bottom: estimated GPU-active time on an NVIDIA RTX 5090. A finer quadrature gives the largest gain, more integration steps help with diminishing returns at a much higher cost, and the rank barely affects cost.

NeuralEF-style baseline. NeuralEF [13] was not proposed for FPCA, so we adapt its objective to the covariance operator of the data. We train $K = 8$ MLPs $\psi_1, \ldots, \psi_K$, each normalized to unit empirical L^2 norm on the training coordinates. For a minibatch of $B = 16$ signals centered by the current mean estimate μ (held fixed in this loss), we form the coefficients $a_{bi} = \langle f_b - \mu, \psi_i \rangle$ and the covariance bilinear forms $R_{ij} = \frac{1}{B} \sum_b a_{bi} a_{bj}$, and minimize Eq. (17) of Deng et al. [13]:

$$\mathcal{L}_{\text{NeuralEF}} = -\sum_{i=1}^{K} R_{ii} + \sum_{j=2}^{K} \sum_{i<j} \frac{R_{ij}^2}{\text{sg}(\max(R_{ii}, \epsilon))}, \tag{49}$$

where sg is the stop-gradient and $\epsilon = 10^{-8}$; four minibatches are accumulated per step. The first term captures variance and the second decorrelates each function from the ones before it, so orthogonality is encouraged by a soft penalty rather than enforced by construction.

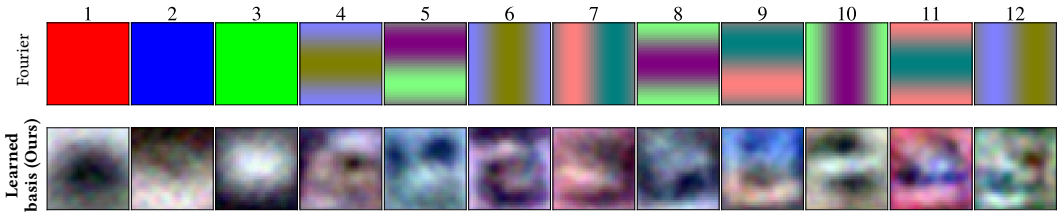


Method	$\max \lvert G - I \rvert$	$\lVert G - I \rVert_F$
NeuralEF	1.50×10^{-1}	3.16×10^{-1}
Ours ($r = 32$)	4.70×10^{-4}	1.57×10^{-3}

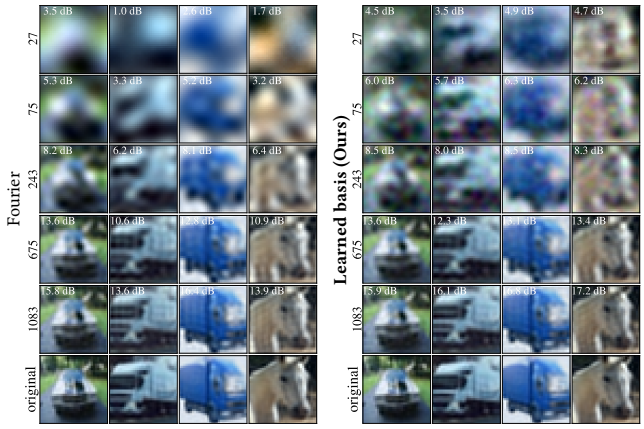
Figure 10: **Comparison with a NeuralEF-style baseline. Left:** held-out reconstruction MSE versus the number of retained atoms. The baseline has eight functions, so its error cannot improve beyond eight atoms (dashed). **Right:** deviation from the identity of the Gram matrix G of the first eight learned functions; smaller is more orthonormal. The baseline is more accurate with up to eight atoms, but our basis keeps improving beyond them and stays orthonormal to numerical precision.

D.6 Additional Experiments

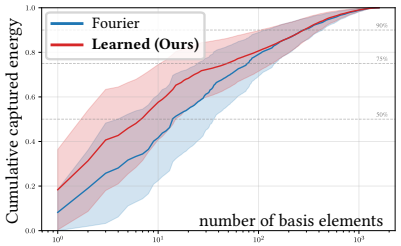
Additional experiments and visualizations are provided in Figure 11–Figure 14. Figure 11 evaluates the learned basis on the CIFAR10 SIREN Zoo [34], showing that the learned bases resemble smooth principal components of the dataset and yield higher PSNRs and greater captured energy than the Fourier basis. Figure 12 presents analogous results for CelebA, where the learned bases take the form of smooth eigenfaces and achieve better reconstruction quality and higher cumulative captured energy



(a) CIFAR10 INR (SIREN zoo) dataset



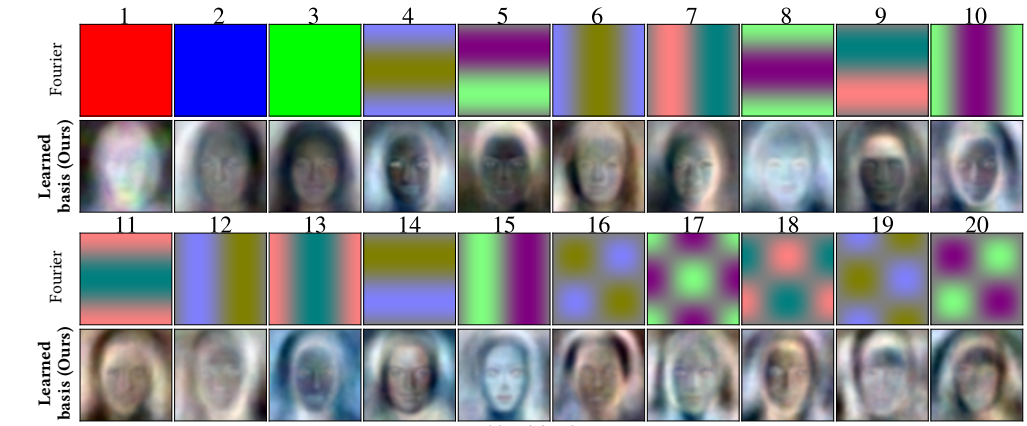
(b) CIFAR10 INR reconstruction



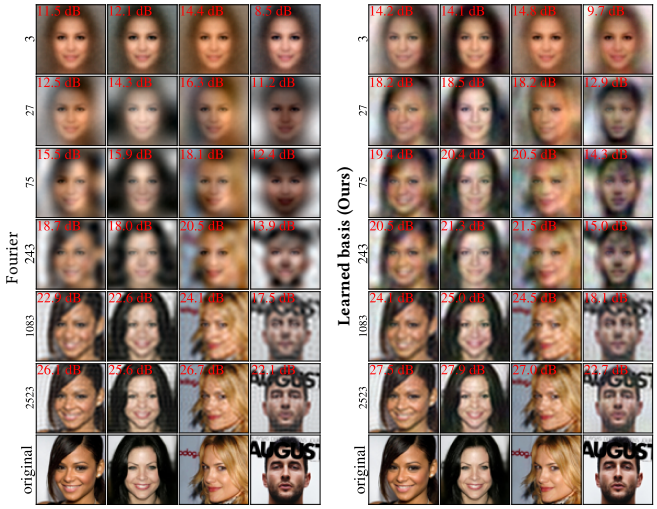
(c) CIFAR10 INR cumulative captured energy

Figure 11: **PCA for CIFAR10 [34] SIREN Zoo. (a)** Visualizes the reference and learned basis after training; the learned bases resemble smooth principal components of the CIFAR10 dataset. **(b)** Reconstruction with the learned basis results in higher PSNRs. **(c)** Quantitatively, the captured energy is higher for our learned basis compared to the Fourier basis.

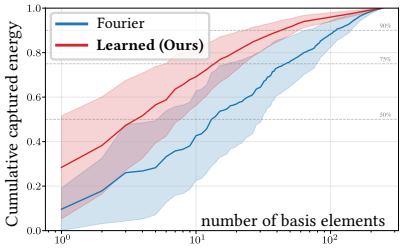
with fewer basis elements. Figure 13 visualizes a truncated set of initial and learned bases across applications—including MNIST, Taylor-Green vortices, and CelebA—demonstrating that while the initialization corresponds to a random path on the Lie manifold, training recovers the eigenmodes of the covariance operator. Finally, Figure 14 illustrates the internal network dynamics for the vortex experiment, visualizing the learned neural fields $U_\theta^{0\cdots4}$ and spatial rotations $\exp(M_\theta - M_\theta^\top)$ that define the generator $K_\theta(t)$ and enable simulation of the Koopman operator via rank-r planar rotations of the Fourier basis.



(a) CelebA dataset

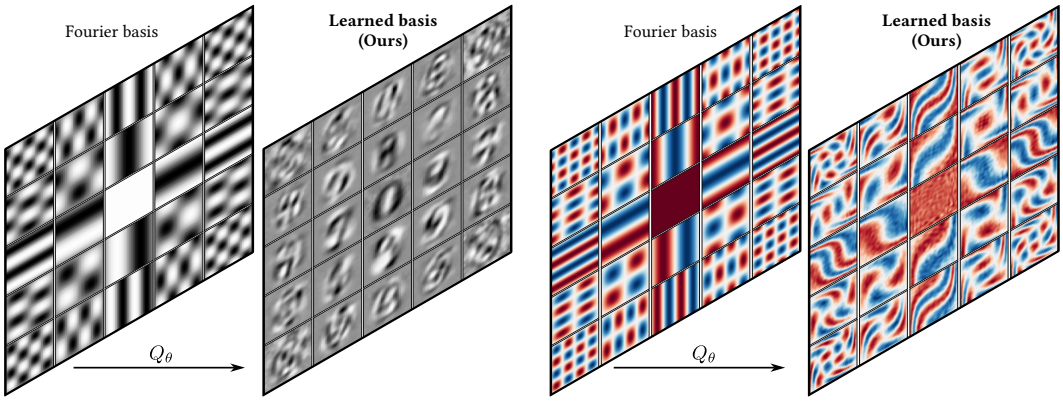


(b) CelebA reconstruction



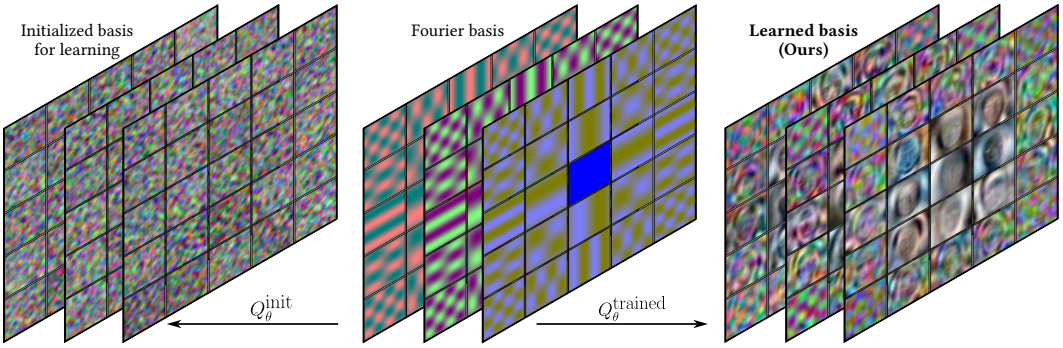
(c) CelebA cumulative captured energy

Figure 12: **PCA for CelebA. (a)** Visualizes the reference and learned basis; the learned bases resemble smooth "eigenfaces" of CelebA. **(b)** Reconstruction with the learned basis obtains higher PSNRs and captures the structure of the face better with fewer elements. **(c)** Quantitatively, the cumulative captured energy is higher for our learned basis compared to Fourier basis.



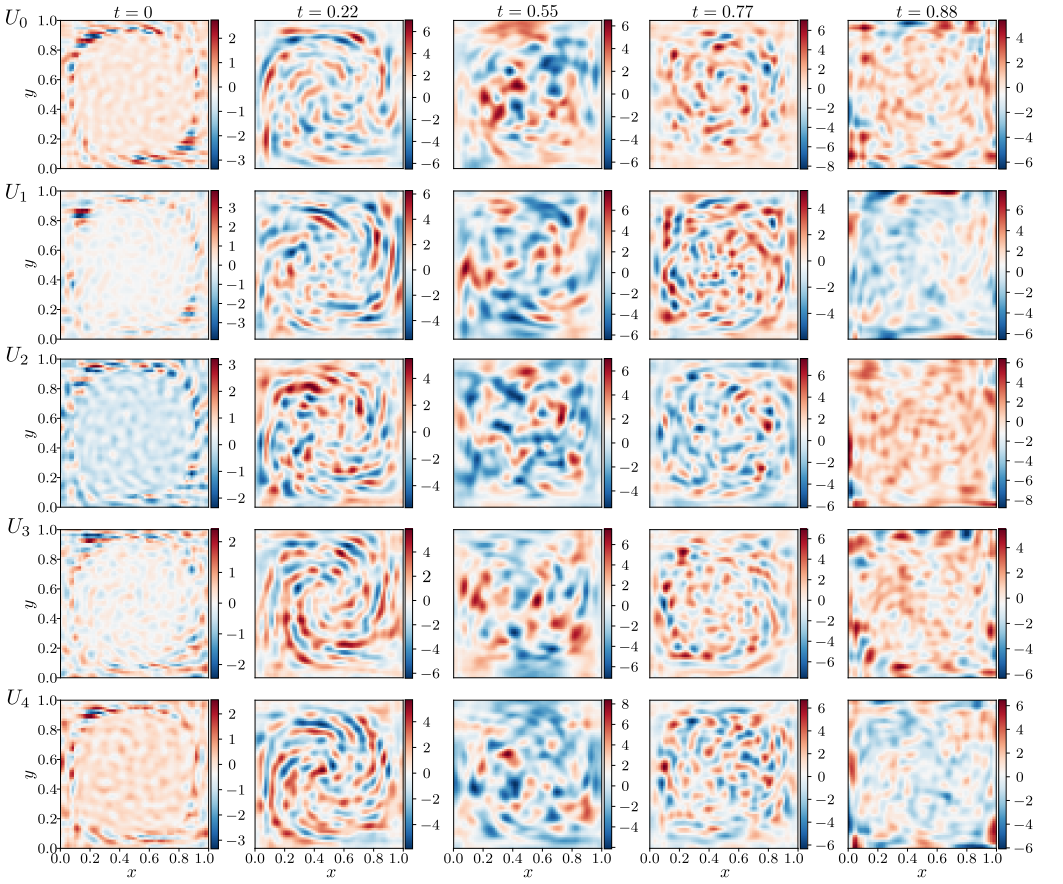
(a) MNIST INR dataset covariance operator learning basis mosaic

(b) Taylor-Green Koopman operator learning basis mosaic

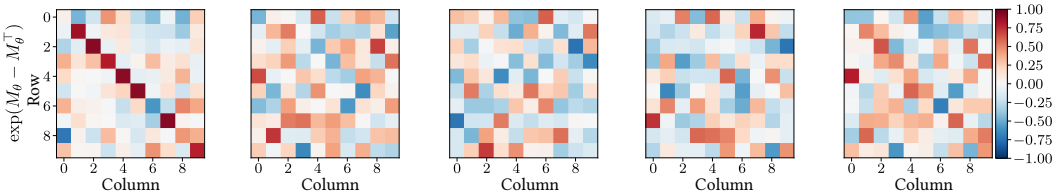


(c) CelebA dataset multi-channel covariance operator learning basis mosaic

Figure 13: **Initial and Learned 2D and Multichannel Bases.** Visualization a truncated set of initial and learned bases across applications. See Appendix D.2 for our indexing scheme used to visualize the 2D and multi-channel bases; the center of the mosaic corresponds to the direct current (DC) Fourier function. **(a)** 2D Fourier and a learned bases fit to the principal components of the MNIST SIREN Zoo; the bases are visualized as a 2D mosaic indexed as φ_{i_1,i_2} for Fourier and $Q_\theta\varphi_{i_1,i_2}$ for the learned basis where i_1 and i_2 correspond to the horizontal and vertical frequencies. **(b)** Similar results for the Vortices where each $Q_\theta\varphi_{i_1,i_2}$ corresponds to a dynamic mode of the Taylor-Green vortex field. **(c)** The 2D RGB-channel Fourier and learned basis for CelebA. Here the bases are visualized as a 2D mosaic with an extra channel dimension and indexed as $\varphi_{i_1,i_2,c}$. The basis $Q_\theta^{\text{init}}\varphi_{i_1,i_2,c}$ at initialization is obtained from a random path on the Lie manifold, but once trained, it learns to map to the eigenmodes of the covariance operator.



(a) Learned generators U_θ (rows: ranks, columns: time)



(b) Rank r rotation matrices

Figure 14: **Internal Network Dynamics.** This figure illustrates the learned neural network outputs that transform the Fourier basis into the dynamic modes of the Taylor-Green vortices. Specifically, we visualize the first five neural fields, $U_\theta^{0\cdots4}$, across various time steps **(a)**, alongside the spatial rotations $\exp(M_\theta - M_\theta^\top)$ **(b)** that jointly characterize the generator $K_\theta(t)$. Passing the Fourier basis through these rank-r planar rotations allows us to simulate the Koopman operator.