SLiK: Learned Sturm–Liouville Spectral Geometry for Sequence Modeling

Anonymous authors
Paper under double-blind review

Abstract

A central challenge in sequence modeling is to represent local structure and long-range dependencies within a unified architecture. Spectral methods provide compact global representations, but typically operate in predefined coordinate systems, such as Fourier or polynomial bases, learning only transformations within them and leaving the retained global subspace largely fixed. We ask whether the coordinate-defining operator can instead be learned from the task. We introduce SLiK, a local–global architecture that learns a continuous, self-adjoint Sturm–Liouville operator end-to-end. Its low-order eigenspace defines task-adapted global coordinates, within which eigenvalue-conditioned spectral responses transform sequence information, while a parallel multi-scale convolutional path preserves local and high-frequency structure. For a fixed number of retained modes, spectral projection and reconstruction scale linearly with sequence length, and the continuous operator can be re-discretized across sequence lengths without introducing length-specific learned parameters. Across four long-term forecasting benchmarks, SLiK is competitive with strong Transformer, Mamba, and MLP baselines. Under matched 180M-parameter, 100B-token language-model pretraining, SLiK remains competitive with strong Transformer and state-space baselines while achieving 80.6% average needle-in-a-haystack retrieval accuracy. On long-read RNA splicing classification, SLiK achieves 0.9063 accuracy with the fastest runtime among evaluated models. Controlled ablations against fixed spectral bases, generic learned orthogonal bases, and frozen operators, with closely matched parameter counts, show that structured operator learning provides gains beyond generic basis adaptation. Together, these results support learning spectral geometry itself as a general design principle for sequence modeling across modalities.

1 Introduction

Sequence modeling provides a common abstraction for data as different as multivariate time series, natural language, and biological sequences. Across these domains, useful representations must resolve structure at multiple scales: local temporal patterns are informative for forecasting (Nie, 2022), linguistic meaning depends on contextual interactions among tokens (Vaswani et al., 2017), and RNA sequence representations must encode signals relevant to both structure and function (Penić et al., 2025). These settings differ in their input alphabets, objectives, and causal constraints, but they share a need to combine local patterns with information distributed over much longer spans. This common structure has motivated the search for sequence backbones that remain expressive and computationally tractable across modalities and sequence lengths (Gu et al., 2021; Gu & Dao, 2024).

Recent architectures expand this design space through efficient attention and long convolutions (Poli et al., 2023; Katharopoulos et al., 2020), as well as structured and selective state-space dynamics (Gu et al., 2021; Gu & Dao, 2024). In parallel, Fourier token mixers show that a compact set of global modes can support efficient interaction (Lee-Thorp et al., 2022; Guibas et al., 2021), while polynomial projection methods use structured modes to summarize sequence history (Gu et al., 2020; 2022). However, these spectral and projection-based models typically learn transformations within a prescribed family of coordinates. This leaves a complementary representation-design question: can the coordinate-defining operator itself be learned from the task, while preserving a compact orthogonal representation whose parameters are not tied to one discrete sequence length?

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

We investigate this trade-off with the Sturm–Liouville spectral kernel (SLiK), a local–global sequence architecture built around a task-learned differential operator. SLiK parameterizes positive coefficient fields $p_\theta(u)$, $q_\theta(u)$, and $w_\theta(u)$ as continuous functions of a normalized coordinate $u \in [0, 1]$. At each requested length, these fields define a symmetric generalized eigenproblem, and a differentiable low-mode solver extracts a compact weighted eigenbasis. Importantly, we do not assume that low eigenvalues are intrinsically task-relevant. Low-order truncation instead provides a compactness and regularity bias; task adaptation arises because learning the Sturm–Liouville operator changes the functions that occupy the retained low-order eigenspace. The model projects token features onto this basis, applies eigenvalue-conditioned spectral responses, and reconstructs a low-rank global update. A parallel multi-scale convolutional path preserves local and high-frequency information that may be attenuated by spectral truncation. The operator is learned across the training set and shared across samples, so it represents task-level rather than input-conditioned spectral geometry.

Our contributions are:

1. We introduce a sequence-modeling formulation in which a continuous, self-adjoint Sturm–Liouville operator is learned end-to-end, allowing the spectral coordinate system itself, rather than only transformations within a fixed basis, to adapt to the task.

2. We develop an eigenvalue-conditioned low-rank spectral mixer coupled to a multi-scale local convolutional path. For fixed spectral rank, projection and reconstruction scale linearly with sequence length, while the continuous operator can be re-discretized across sequence lengths without introducing length-specific learned parameters.

3. We evaluate SLiK across forecasting, language modeling, and long-read RNA sequence classification, supporting learned spectral geometry as a general sequence-modeling principle across modalities.

2 Related Work

Efficient models. Self-attention compares token pairs to provide content-dependent global communication, but its conventional formulation has quadratic cost in the sequence length (Vaswani et al., 2017). Linear-attention methods reduce this cost by expressing attention through kernel feature maps and reordering the resulting products (Katharopoulos et al., 2020). Other subquadratic backbones impose different structures on long-range interaction: Hyena uses implicitly parameterized long convolutions (Poli et al., 2023), whereas S4 compresses history through structured state-space dynamics that admit both recurrent and convolutional evaluation (Gu et al., 2021). Mamba makes these dynamics input-dependent (Gu & Dao, 2024), and Mamba-3 increases the expressivity of the recurrent state update for retrieval, state tracking, and language modeling (Lahoti et al., 2026). These methods adapt pairwise interactions, convolutional filters, or state transitions. SLiK explores a complementary mechanism: global information is routed through a compact spectral geometry learned at the task level. Unlike attention and selective state-space dynamics, this geometry is shared across samples rather than conditioned on each input.

Spectral sequence representations. Spectral models use global basis functions to communicate across an entire domain. FNet replaces attention with an unparameterized Fourier transform (Lee-Thorp et al., 2022), while adaptive Fourier neural operators learn token mixing within Fourier coordinates (Guibas et al., 2021). Reduced-order frequency-domain models similarly learn transformations after projecting onto selected Fourier modes (Poli et al., 2022). In time-series forecasting, FEDformer combines Fourier representations with seasonal–trend decomposition (Zhou et al., 2022b), and FiLM combines Legendre projection, Fourier filtering, and low-rank approximation (Zhou et al., 2022a). HiPPO instead represents streaming history through projection onto orthogonal polynomial bases (Gu et al., 2020); its generalized formulation interprets structured state-space models through warped polynomial and Fourier projections (Gu et al., 2022). These approaches demonstrate the value of compact global coordinates, but the Fourier, Legendre, or related basis family is specified before learning, even when mode weights, filters, or dynamics are trainable. In SLiK, task adaptation also changes the coordinate-defining differential operator and hence the weighted orthogonal basis itself.

Learnable operator bases. Beyond fixed Fourier and polynomial systems, several operator-learning architectures learn or adapt the functions used for projection and reconstruction. DeepONet learns output coordinate functions through its trunk network (Lu et al., 2021), while BelNet and BasisONet more explicitly learn continuous basis representations (Zhang et al., 2022; Hua & Lu, 2023). BasisONet additionally promotes orthonormality through a training penalty. More recent approaches adapt spectral representations by calibrating preset eigenfunctions with pointwise gates (Yue et al., 2024) or by constructing a data-adaptive Parseval frame (Zhao & Aviles-Rivero, 2026). These methods demonstrate that adaptation can extend beyond spectral coefficients, but they do not derive shared sequence coordinates from the low eigenfunctions of a task-learned one-dimensional self-adjoint differential operator. Deep Sturm–Liouville constructs learnable orthogonal basis functions for continuous regularization along field lines in input space (Vigouroux et al., 2025). In contrast, SLiK learns a Sturm–Liouville operator over normalized sequence coordinates, whose leading eigenfunctions define a low-rank global mixing basis and whose eigenvalues modulate the spectral response. A parallel convolutional branch captures local and high-frequency information, while continuous coefficient fields enable re-discretization across sequence lengths. Unlike Deep Sturm–Liouville, which uses the operator as a regularization basis over input representations, SLiK uses it to define the spectral geometry of global sequence mixing.

3 METHOD

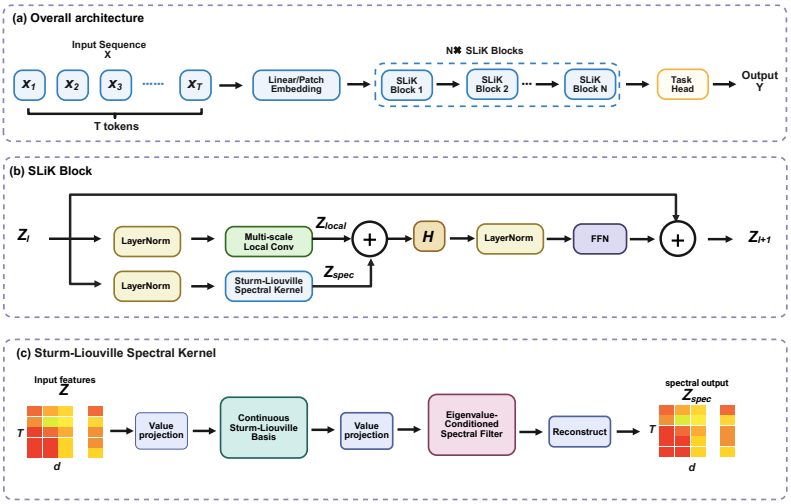

Figure 1: **SLiK architecture.** (a) An input sequence is embedded and processed by L SLiK blocks followed by a task-specific head. (b) Each block combines a multi-scale depthwise-convolution branch and a Sturm–Liouville spectral-memory branch through LayerScale residual updates, followed by a pre-norm feed-forward network. (c) Continuous positive coefficient fields define a generalized Sturm–Liouville eigenproblem.

3.1 PROBLEM FORMULATION AND ARCHITECTURE OVERVIEW

Let $\mathbf{X} \in \mathbb{R}^{B\times T\times C_{\text{in}}}$ denote a batch of sequences, where B, T, and C_{in} are the batch size, sequence length, and number of input channels, respectively.For patch-based forecasting, T denotes the number of patches and the same formulation is applied after patch embedding. Our goal is to construct a sequence representation that captures local patterns and global dependencies without tying the model parameters to a particular value of T. A linear input map produces

$$\mathbf{Z}_0 = \mathbf{X}\mathbf{W}_{\text{in}} + \mathbf{b}_{\text{in}} \in \mathbb{R}^{B\times T\times D}, \tag{1}$$

where D is the hidden dimension.

The overall architecture of SLiK is illustrated in Fig. 1(a). For an input sequence of length T, SLiK constructs the length-specific eigensystem

$$\mathcal{E}_\theta(T) = \big(\mathbf{\Lambda}, \mathbf{\Phi}, \mathbf{M}\big), \tag{2}$$

from a length-independent learnable Sturm–Liouville operator. Here, $\mathbf{\Lambda}$ contains the retained eigenvalues, $\mathbf{\Phi}$ contains the corresponding eigenfunctions evaluated on the length-T grid, and $\mathbf{M}$ is the discrete mass matrix induced by the Sturm–Liouville weight function. The resulting eigensystem is computed once for a dense sequence and shared by all L SLiK blocks. Each block combines a multi-scale local branch with a global spectral memory branch, followed by a feed-forward residual update. The final output is

$$\widehat{\mathbf{Y}} = h_{\mathrm{task}}(\mathrm{LN}_{\mathrm{out}}(\mathbf{Z}_L)), \tag{3}$$

where h_{task} is a task-specific classification, forecasting, or sequence-modeling head. The backbone itself returns token-wise features and does not require a fixed sequence length.

3.2 Continuous Learnable Sturm–Liouville Basis

Fixed Fourier or cosine bases impose a task-independent geometry on the sequence domain. We instead learn a variable-coefficient Sturm–Liouville operator on the normalized coordinate $t \in [0, 1]$,

$$\mathcal{L}_\theta \phi(t) = -\frac{\mathrm{d}}{\mathrm{d}t}\left(p_\theta(t)\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}\right) + q_\theta(t)\phi(t), \tag{4}$$

with the weighted eigenproblem

$$\mathcal{L}_\theta \phi_k(t) = \lambda_k w_\theta(t)\phi_k(t). \tag{5}$$

Here, $p_\theta(t) > 0$ controls coordinate-dependent diffusion, $q_\theta(t) > 0$ acts as a potential term, and $w_\theta(t) > 0$ defines the weighted geometry of the sequence domain. Eigenvalue ordering provides a structured compactness and regularity bias for truncation, rather than an a priori measure of task relevance. Task adaptation instead arises from learning the operator that defines the retained eigenspace. The coefficient fields are generated by a small coordinate network rather than by one parameter per sequence position. Specifically, with Fourier coordinate features

$$\boldsymbol{\psi}(t) = \left[2t - 1, \{\sin(\pi f_j t), \cos(\pi f_j t)\}_{j=1}^{F}\right], \tag{6}$$

A learnable coordinate network $g_\theta : \mathbb{R}^{2F+1} \to \mathbb{R}^3$ maps these features to three coefficient-specific modulations:

$$\boldsymbol{\delta}_\theta(t) = \tanh(g_\theta(\boldsymbol{\psi}(t))) \in [-1, 1]^3. \tag{7}$$

Positive coefficients are then obtained as

$$p_\theta(t) = \mathrm{softplus}\left(\bar{p} + s\delta_{\theta,p}(t)\right) + \epsilon, \tag{8}$$

$$q_\theta(t) = \mathrm{softplus}\left(\bar{q} + s\delta_{\theta,q}(t)\right) + \epsilon, \tag{9}$$

$$\widetilde{w}_\theta(t) = \mathrm{softplus}\left(\bar{w} + s\delta_{\theta,w}(t)\right) + \epsilon, \qquad w_\theta(t) = \frac{\widetilde{w}_\theta(t)}{T^{-1}\sum_{i=0}^{T-1}\widetilde{w}_\theta(t_i)}. \tag{10}$$

The normalization of w_θ removes the redundant global scale of the generalized mass matrix while retaining its coordinate-dependent shape.

For a requested length T, we sample the continuous fields on a uniform grid and obtain a symmetric tridiagonal operator matrix $\mathbf{A}_{\theta,T}$ and a positive diagonal mass matrix $\mathbf{M}_{\theta,T}$. We solve the generalized eigenproblem

$$\mathbf{A}_{\theta,T}\mathbf{\Phi} = \mathbf{M}_{\theta,T}\mathbf{\Phi}\mathbf{\Lambda}, \qquad \mathbf{\Phi}^\top \mathbf{M}_{\theta,T}\mathbf{\Phi} = \mathbf{I}, \tag{11}$$

and retain the $K_T = \min(K, T)$ lowest modes. Thus, $\mathbf{\Phi} \in \mathbb{R}^{T\times K_T}$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{K_T})$. These low modes provide a compact set of global coordinates adapted to the learned operator. Because the network parameters describe continuous functions on $[0, 1]$, the same parameters can construct $\mathcal{E}_\theta(T)$ for different sequence lengths. The exact discretization and differentiable low-mode solver are given in Appendices B and C.

3.3 Eigenvalue-Conditioned Spectral Memory

The spectral branch in Fig. 1(c) maps token features to a compact set of global memory slots, transforms them, and reconstructs a token-wise update. Given a normalized block input $\mathbf{Z}^{\mathrm{spec}} \in \mathbb{R}^{B\times T\times D}$, we first form values

$$\mathbf{V} = \mathbf{Z}^{\mathrm{spec}}\mathbf{W}_v. \tag{12}$$

Weighted projection onto the learned eigenbasis gives

$$\mathbf{C} = \mathbf{\Phi}^\top \mathbf{M}_{\theta,T} \mathbf{V} \in \mathbb{R}^{B \times K_T \times D}. \tag{13}$$

Although $\mathbf{C}$ contains only K_T modes, every coefficient aggregates information over the complete sequence domain.

The eigenvalues provide an intrinsic coordinate for the spectral modes. We normalize their log scale as

$$\eta_k = \frac{\log(1 + \max(\lambda_k, 0))}{\max(\log(1 + \max(\lambda_{K_T}, 0)), \epsilon)}, \tag{14}$$

and generate R response gates

$$\mathbf{g}_k = \sigma(f_{\text{gate}}(\eta_k)) \in (0, 1)^R. \tag{15}$$

Let $\mathbf{G}_r = \operatorname{diag}(g_{1r}, \ldots, g_{K_T,r})$ and let $\mathbf{W}_r \in \mathbb{R}^{D \times D}$ be the channel map of response r. The filtered coefficients are

$$\widetilde{\mathbf{C}} = \mathcal{N}_{\text{spec}} \left[\mathbf{C} + \mathcal{D}_{\text{mode}} \left(\frac{1}{\sqrt{R}} \sum_{r=1}^{R} \mathbf{G}_r \mathbf{C} \mathbf{W}_r \right) \right], \tag{16}$$

where $\mathcal{D}_{\text{mode}}$ shares its dropout mask across the mode axis. The normalization $\mathcal{N}_{\text{spec}}$ uses a global RMS over the mode and channel axes followed by a learned channel scale. These choices make the filter equivariant to admissible rotations within repeated eigenspaces; Appendix D gives the corresponding argument.

Finally, the global token update is reconstructed as

$$\mathbf{Z}_{\text{global}} = \mathcal{D}_{\text{out}} \left(\mathbf{\Phi} \widetilde{\mathbf{C}} \mathbf{W}_o \right). \tag{17}$$

Projection and reconstruction communicate information globally without materializing a dense $T \times T$ interaction matrix.

3.4 Local-Residual SLiK Block

Truncation to low spectral modes provides compact global context but can omit high-frequency local details. Each SLiK block therefore applies a parallel multi-scale convolutional branch, as shown in Fig. 1(b). For odd kernel sizes $\mathcal{K} = \{\kappa_1, \ldots, \kappa_m\}$, the local update is

$$\mathbf{Z}_{\text{local}} = \mathcal{D}_{\text{local}} \left(\text{PWConv} \left[\Big\Vert_{\kappa \in \mathcal{K}} \text{GELU}(\text{DWConv}_\kappa \left(\text{LN}_{\text{local}}(\mathbf{Z}_\ell) \right)) \right] \right), \tag{18}$$

where $\Vert$ denotes channel-wise concatenation. Depthwise convolutions model local patterns at multiple scales, and the pointwise convolution fuses the resulting branches.

The local and spectral updates are combined through independently normalized pre-norm branches and channel-wise LayerScale parameters:

$$\mathbf{H}_\ell = \mathbf{Z}_\ell + \boldsymbol{\gamma}_{\text{local},\ell} \odot \mathbf{Z}_{\text{local}} + \boldsymbol{\gamma}_{\text{spec},\ell} \odot \text{SM}_\ell(\text{LN}_{\text{spec},\ell}(\mathbf{Z}_\ell); \mathbf{\Lambda}, \mathbf{\Phi}, \mathbf{M}_{\theta,T}), \tag{19}$$

$$\mathbf{Z}_{\ell+1} = \mathbf{H}_\ell + \boldsymbol{\gamma}_{\text{ffn},\ell} \odot \text{FFN}_\ell(\text{LN}_{\text{ffn},\ell}(\mathbf{H}_\ell)). \tag{20}$$

Here, SM_ℓ denotes the spectral memory operation in Eqs. equation 12–equation 17, and $\odot$ is channel-wise multiplication. The LayerScale vectors are initialized to a small positive value, allowing the network to begin near the identity map while gradually introducing local, spectral, and feed-forward updates.

3.5 Variable-Length Processing and Computational Complexity

The continuous operator parameters do not depend on T. Consequently, two calls to the same model may use different sequence lengths, with the eigensystem rebuilt at the requested resolution. For a prefix-padded batch with true lengths $\{T_b\}_{b=1}^{B}$, we group examples by their true length, crop each group to its unpadded prefix, construct one eigensystem per distinct length, and pad only the returned

features. This prevents the padded suffix from changing either the spectral domain or the prediction. This procedure assumes prefix-valid sequences; arbitrary internal missingness requires a different masking or discretization strategy.

Let H_c be the coefficient-network width, $K_s = \min(T, K + r)$ the oversampled subspace size, J the number of subspace iterations, and R the number of spectral responses. For a dense equal-length batch, basis construction costs

$$\mathcal{O}\big(TH_c^2 + JTK_s^2 + K_s^3\big)\,, \tag{21}$$

and a stack of L blocks costs

$$\mathcal{O}\big(LB\left(TK_TD + K_TRD^2 + TD^2\right)\big)\,. \tag{22}$$

The total forward complexity is the sum of Eqs. equation 21 and Eq. equation 22. With fixed H_c, K_s, K_T, R, D, J, L, and batch size B, both basis construction and block computation are linear in T. This statement applies to the truncated long-sequence regime; if $K_s = T$, the reduced QR and eigendecomposition can scale cubically in T. During no-gradient evaluation, an eigensystem may be cached and reused for repeated calls with the same length and unchanged operator parameters. A complete derivation and memory analysis are provided in Appendix F.

3.6 Causal SLiK for Autoregressive Language Modeling

The original SLiK encoder assumes that the complete input window is available. This assumption is appropriate for time-series forecasting because every position in the historical context is observed before the forecast is made. It is not valid for autoregressive language modeling, where the representation at position t must be independent of tokens at positions $s > t$.

For language modeling, the input representation is initialized from token embeddings,

$$\mathbf{Z}_t^{(0)} = \mathbf{E}[x_t], \tag{23}$$

where $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}|\times D}$ is the token-embedding matrix. We retain the learned continuous Sturm–Liouville basis, eigenvalue-conditioned spectral responses, and local–global residual structure. We replace the information flow within both token-mixing branches by causal operations.

Given the normalized representation $\mathbf{Z}_t$, the spectral branch first computes

$$\mathbf{V}_t = \mathbf{Z}_t\mathbf{W}_v. \tag{24}$$

Instead of projecting the complete sequence into one coefficient tensor as in Eq. equation 13, causal SLiK maintains a prefix-indexed modal memory,

$$\mathbf{C}_{t,k} = \mathbf{C}_{t-1,k} + m_t\phi_{tk}\mathbf{V}_t = \sum_{s=1}^{t} m_s\phi_{sk}\mathbf{V}_s, \qquad k = 1, \ldots, K_T, \tag{25}$$

with $\mathbf{C}_{0,k} = \mathbf{0}$. The eigenvalue-conditioned gates and channel maps are applied independently to the memory available at each position:

$$\widetilde{\mathbf{C}}_{t,k} = \mathcal{N}_t\left[\mathbf{C}_{t,k} + \sum_{r=1}^{R} g_{kr}\mathbf{C}_{t,k}\mathbf{W}_r\right], \tag{26}$$

where $\mathcal{N}_t$ computes its RMS statistic only over the mode and channel dimensions at position t. The causal global update is reconstructed as

$$\mathbf{S}_t = \left(\sum_{k=1}^{K_T}\phi_{tk}\widetilde{\mathbf{C}}_{t,k}\right)\mathbf{W}_o. \tag{27}$$

The local branch is causalized independently. For a convolution of width κ, we apply $\kappa - 1$ zeros on the left and no padding on the right. Its output at position t therefore depends only on positions $\max(1, t - \kappa + 1), \ldots, t$. The feed-forward transformation remains position-wise. Consequently, every output representation satisfies

$$\frac{\partial \mathbf{Z}_t^{(L)}}{\partial \mathbf{Z}_s^{(0)}} = \mathbf{0}, \qquad s > t. \tag{28}$$

Table 1: Long-term forecasting results on four benchmark datasets. The best results are highlighted in **bold**, and the second-best results are underlined. Lower values indicate better performance.

Dataset	Horizon	SLiK		Affirm		DTMamba		TSLANet		iTransformer		PatchTST		Crossformer		RLinear		DLinear	
		MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓
ETTh2	96	**0.244**	**0.336**	0.276	0.343	0.290	0.340	0.280	0.341	0.297	0.349	0.285	0.340	0.745	0.584	0.288	0.338	0.289	0.353
	192	**0.286**	**0.361**	0.316	0.372	0.366	0.392	0.330	0.375	0.380	0.400	0.356	0.386	0.877	0.656	0.374	0.390	0.383	0.418
	336	0.328	0.389	0.341	0.383	0.380	0.409	**0.317**	**0.374**	0.428	0.432	0.350	0.395	1.043	0.731	0.415	0.426	0.448	0.465
	720	0.414	0.442	**0.391**	0.429	0.416	0.437	0.404	0.440	0.427	0.445	0.395	**0.427**	1.104	0.763	0.420	0.440	0.605	0.551
ETTm2	96	**0.152**	0.257	0.167	0.260	0.177	0.259	0.169	0.259	0.180	0.264	0.169	**0.254**	0.287	0.366	0.182	0.265	0.167	0.260
	192	**0.190**	**0.288**	0.221	0.296	0.240	0.300	0.224	0.297	0.250	0.309	0.230	0.294	0.414	0.492	0.246	0.304	0.224	0.303
	336	**0.231**	**0.318**	0.271	0.328	0.310	0.345	0.275	0.329	0.311	0.348	0.280	0.329	0.597	0.542	0.307	0.342	0.281	0.342
	720	**0.293**	**0.357**	0.351	0.377	0.395	0.394	0.354	0.380	0.412	0.407	0.378	0.386	1.730	1.042	0.407	0.398	0.397	0.421
Weather	96	**0.144**	**0.193**	0.146	0.196	0.171	0.218	0.148	0.197	0.174	0.214	0.160	0.204	0.158	0.230	0.192	0.232	0.176	0.237
	192	**0.190**	**0.239**	0.192	**0.239**	0.220	0.257	0.193	0.241	0.221	0.254	0.204	0.245	0.206	0.277	0.240	0.271	0.220	0.282
	336	**0.244**	0.281	**0.244**	**0.278**	0.274	0.296	0.245	0.282	0.278	0.296	0.257	0.285	0.272	0.335	0.292	0.307	0.265	0.319
	720	0.321	0.334	0.321	**0.332**	0.349	0.346	0.325	0.337	0.358	0.349	0.329	0.338	**0.298**	0.418	0.364	0.353	0.323	0.362
Electricity	96	0.139	0.227	**0.129**	**0.223**	0.166	0.256	0.136	0.229	0.148	0.240	0.138	0.230	0.219	0.314	0.201	0.281	0.140	0.237
	192	0.151	0.242	**0.146**	**0.239**	0.178	0.268	0.152	0.244	0.162	0.253	0.149	0.243	0.231	0.322	0.201	0.283	0.153	0.249
	336	0.163	0.260	**0.162**	**0.252**	0.197	0.289	0.168	0.262	0.178	0.269	0.169	0.262	0.246	0.337	0.215	0.298	0.169	0.267
	720	0.210	**0.288**	**0.191**	**0.288**	0.243	0.326	0.205	0.293	0.225	0.317	0.211	0.299	0.280	0.363	0.257	0.331	0.203	0.301

The SL basis depends on the learned coefficient fields and positional domain, but not on token identities. In the language-model implementation, we solve the eigensystem on the maximum context domain and use the corresponding prefix for shorter sequences. This convention ensures that an identical token prefix receives the same positional basis during full-sequence training and token-by-token decoding.

We place the causal SLiK mixer within a pre-RMSNorm residual language-model block and use a position-wise SwiGLU feed-forward network. These components form the shared language-model scaffold used in our parameter-matched comparisons. The input and output embedding matrices are tied. Given the final hidden representation $\mathbf{h}_t$, the next-token distribution is

$$p_\theta(x_{t+1} \mid x_{\le t}) = \mathrm{softmax}\left(\mathbf{h}_t \mathbf{E}^\top\right)_{x_{t+1}}. \tag{29}$$

The model is optimized with the standard next-token negative log-likelihood,

$$\mathcal{L}_{\mathrm{LM}} = -\frac{1}{T}\sum_{t=1}^{T} \log p_\theta(x_{t+1} \mid x_{\le t}). \tag{30}$$

The cumulative modal update and reconstruction require $\mathcal{O}(BTK_TD)$ operations. Applying R full-rank channel responses at every prefix requires $\mathcal{O}(BTRK_TD^2)$ operations, while the remaining output projections and feed-forward transformations require $\mathcal{O}(BTD^2)$. The recurrent modal state has size $\mathcal{O}(BK_TD)$, although parallel training retains $\mathcal{O}(BTK_TD)$ modal activations for backpropagation. For fixed K_T, R, and D, computation remains linear in sequence length and no dense $T \times T$ token-interaction matrix is materialized.

4 Experiments

4.1 Experiments on Time Series Forecasting

Data. To validate our model, we evaluate SLiK on 4 benchmarks: 2 ETTs (ETTh2 and ETTm2), Electricity (ECL), and Weather (Wu et al. (2021)). We trained and tested our model with a 60/20/20 train/validation/test split for ETTs, and 70/10/20 for other datasets. We use a look-back window of 336 for ETTs, 512 for Weather, and 96 for Electricity, similar to the Wu et al. (2025) setting. All datasets are normalized during training.

Baseline. We assessed SLiK against eight baselines, including: Transformers (iTransformer Liu et al. (2023), PatchTST Nie et al. (2023), and Crossformer Zhang & Yan (2023)), CNNs (TSLANet Eldele et al. (2024)), MLPS (Dlinear Zeng et al. (2023) and Rlinear Li et al. (2026)), and Mambas (DTMamba Wu et al. (2024) and Affirm Wu et al. (2025))

Results. Table 1 reports the forecasting results across four datasets and four prediction horizons. Overall, SLiK ranked first or second in 26 of the 32 dataset–horizon–metric settings. Compared with Mamba-based models, SLiK achieved lower errors than DTMamba in 31 settings and reduced the average MSE and MAE over Affirm by 3.9% and 0.5%, respectively. Its advantage was most evident

Table 2: Ablation results on ETTm2 with a prediction horizon of 96. The remaining columns report model complexity, runtime, and numerical diagnostics. Basis orthogonality is measured as $\|\mathbf{\Phi}^\top \mathbf{M} \mathbf{\Phi} - \mathbf{I}\|_\infty$, with $\mathbf{M} = \mathbf{I}$ for the generic and DCT bases; "–" denotes an unavailable metric.

Model variant	MSE↓	MAE↓	Params (M)	FLOPs (G)	Inf. (s)	Basis build (s)	Basis orth. err.	SL eig. resid.
Full	0.152	0.257	0.404846	0.062416	0.004129	0.004894	1.78×10^{-15}	1.24×10^{-7}
w/o sl operator	0.169	0.270	0.353568	0.042865	0.001933	0.000000	–	–
generic learned orthogonal	0.157	0.261	0.404040	0.062416	0.003879	0.000365	8.88×10^{-16}	–
fixed dct	0.162	0.266	0.403368	0.062416	0.004170	0.000317	1.03×10^{-15}	–
frozen sl operator	0.159	0.263	0.404846	0.062416	0.004232	0.005858	1.55×10^{-15}	9.06×10^{-12}
constant coefficients	0.164	0.267	0.404846	0.062416	0.004173	0.004624	2.44×10^{-15}	2.58×10^{-15}
classical sl	0.163	0.266	0.404846	0.062416	0.004122	0.005760	1.55×10^{-15}	2.42×10^{-15}

Table 3: Language modeling and zero-shot commonsense reasoning results at the 180M-parameter scale. Lower perplexity (↓) and higher accuracy (↑) are better.

Model	**FW-Edu**	**LAMB.**		**HellaS.**	**PIQA**	**Arc-E**	**Arc-C**	**WinoGr.**	**OBQA**	**Avg.**	**Params.**	**Train token/s.**
	PPL ↓	PPL ↓	Acc. ↑	Acc._n ↑	Acc. ↑	Acc._n ↑	Acc. ↑	Acc. ↑	Acc. ↑	↑	↑	↑
Transformer-180M	16.89	45.0	32.5	39.0	**67.1**	59.8	27.9	51.2	21.8	42.8	186,993,408	578,920
GDN-180M	16.52	40.8	31.3	40.2	66.3	**62.3**	**28.2**	51.7	22.0	43.2	186,829,272	539,437
Mamba-2-180M	16.76	41.8	30.9	40.1	66.8	60.1	27.3	52.0	23.2	42.9	187,046,496	571,790
Mamba-3-SISO-180M	16.59	37.7	32.5	40.8	66.1	61.5	27.9	52.0	22.8	43.4	186,849,600	**608,579**
Mamba-3-MIMO-180M	**16.46**	**32.1**	**34.0**	**41.0**	66.7	60.6	27.7	**52.9**	22.0	43.5	187,291,968	378,233
SLiK-180M	16.50	35.7	33.9	40.7	66.5	60.8	27.5	52.3	**23.4**	**43.6**	187,209,206	598,210

on ETTm2, where it achieved the best MSE at every horizon and the best MAE at three horizons; at horizon 720, SLiK obtained 0.293/0.357 MSE/MAE, compared with 0.351/0.377 for Affirm. SLiK also outperformed iTransformer, PatchTST, and Crossformer in 32, 27, and 31 comparisons, respectively, while exhibiting substantially less long-horizon degradation than Crossformer on ETTm2. Moreover, SLiK ranked first or second in every Weather setting. Electricity was the main exception, where Affirm performed better overall, indicating that SLiK provides broadly competitive, but dataset-dependent, forecasting performance.

Table 2 isolates the principal design choices in SLiK. The full model achieved the lowest MSE and MAE, whereas removing the Sturm–Liouville operator produced the largest accuracy degradation. Learning a generic orthogonal basis improved MSE/MAE from $0.162/0.266$ for the fixed DCT basis to $0.157/0.261$. The full model further reduced the errors to $0.152/0.257$ with a comparable parameter count. Freezing the SL operator yielded $0.159/0.263$. These comparisons separate the benefit of generic basis adaptation from the additional benefit of the learned SL parameterization. Additional component-wise ablations and numerical diagnostics are provided in Appendix J.1.

4.2 Language Modeling

Data. We use LM Evaluation Harness Gao et al. (2024) to test the zero-shot language modeling capabilities of our pretrained model on LAMBADA (OpenAI version) Paperno et al. (2016), HellaSwag Zellers et al. (2019), PIQA Bisk et al. (2020), Arc-Easy/Arc-Challenge Clark et al. (2018), WinoGrande Sakaguchi et al. (2021), and OpenBookQA Mihaylov et al. (2018).

Baseline. We assessed SLiK against four baselines, including: Transformer Vaswani et al. (2017), Mamba2 Dao & Gu (2024), Gated DeltaNet (GDN) Schlag et al. (2021); Yang et al. (2024) and Mamba3 Lahoti et al. (2026).

Results. All models are pretrained with 100B tokens of the FineWeb-Edu dataset Penedo et al. (2024) with the Llama-3.1 tokenizer Grattafiori et al. (2024) at a 2K context length with the same standard training protocol. Table 3 compares all models at the 180M-parameter scale. SLiK achieves the highest average downstream accuracy of 43.6, surpassing the Transformer, Mamba-3-SISO, and Mamba-3-MIMO by 0.8, 0.2, and 0.1 percentage points, respectively. It performs best on OpenBookQA (23.4) and second best on LAMBADA accuracy (33.9) and WinoGrande (52.3). SLiK also improves over Mamba-3-SISO in language modeling, reducing perplexity from 16.59 to 16.50 on FineWeb-Edu and from 37.7 to 35.7 on LAMBADA. Although Mamba-3-MIMO achieves lower perplexities of 16.46 and 32.1 using rank-4 multi-input, multi-output projections under a comparable parameter budget (Lahoti et al., 2026), SLiK remains highly competitive with both Mamba-3 variants. Our model is also tested for retrieval capabilities in Appendix K.1.

Table 4: Ablation study of SLiK. Lower perplexity (PPL) is better. ΔPPL denotes the increase relative to the full model.

Model variant	**PPL ↓**	**ΔPPL**
SLiK (full)	**16.50**	0.00
w/o SL operator	18.72	+2.22
Fixed DCT	17.29	+0.79
Frozen SL operator	<u>17.03</u>	+0.53
Identity spectral filter	17.41	+0.91
Generic learned orthogonal	17.44	+0.94
w/o eigenvalue conditioning	17.99	+1.49
Constant coefficients	17.47	+0.97

Table 5: Classification performance and runtime on the seven-class long-read RNA splicing dataset.

Model	Accuracy ↑	Macro-F1 ↑	Macro-AUROC ↑	Macro-AUPRC ↑	Runtime (min/epoch) ↓	Params(M)
SLiK	**0.9063**	<u>0.9022</u>	0.9845	<u>0.9386</u>	**2.96**	2.27
Transformer	0.9003	0.8966	**0.9877**	**0.9430**	<u>4.79</u>	1.50
Mamba	<u>0.9043</u>	**0.9026**	<u>0.9870</u>	0.9339	6.84	1.55

Table 4 shows that the full model achieved the lowest perplexity of 16.50. Removing the SL operator caused the largest degradation (18.72), followed by removing eigenvalue conditioning (17.99). All other variants increased PPL by 0.53–0.97, confirming the joint contribution of operator learning, adaptive spectral filtering, and learned coordinate-dependent coefficients.

4.3 Long-read long RNA sequence modeling

Data. We use the E-MTAB-7334 long-read RNA-seq dataset from ONT MinION-sequenced human iPSC-derived cardiomyocytes (Zhu et al., 2021). After preprocessing and class balancing, it contains 36,764 transcripts across seven splicing classes: reference transcripts (REF), novel alternative splice sites (ASS), exon skipping (ES), novel exons (NE), intron retention (IR), new combinations of existing exons (NC), and multiple novel events (MN). Inputs are fixed at 2,600 nucleotides, using a gene-disjoint 80/20 training/test split with no gene overlap between partitions.

Benchmark. To evaluate the predictive capabilities and computational efficiency of SLiK, we benchmark it against Transformer Vaswani et al. (2017) and Mamba Gu & Dao (2024) on the transcript splicing classification task.

Results. Table 5 compares the models on the seven-class long-read RNA splicing dataset. SLiK achieved the highest accuracy of 0.9063, compared with 0.9043 for Mamba and 0.9003 for Transformer. Its Macro-F1 score of 0.9022 closely matched Mamba's best result of 0.9026. Transformer achieved the highest Macro-AUROC and Macro-AUPRC, while SLiK ranked second in Macro-AUPRC and outperformed Mamba. SLiK also achieved the shortest runtime at 2.96 minutes/epoch. It was $1.62\times$ faster than Transformer and $2.31\times$ faster than Mamba. These results show that SLiK provides a favorable balance between classification performance and computational efficiency.

5 Conclusion

We introduce SLiK, a local–global sequence model that learns a continuous Sturm–Liouville operator for task-adaptive spectral mixing. By combining learned spectral geometry with eigenvalue-conditioned filtering and local convolutions, SLiK enables efficient global modeling without quadratic token interactions. Across time-series forecasting, language modeling, and long-read RNA sequence classification, SLiK achieves consistently competitive performance, supporting learned Sturm–Liouville geometry as a general design principle for efficient sequence modeling.

REFERENCES

Aryaman Arora, Neil Rathi, Nikil Roashan Selvam, Róbert Csordás, Dan Jurafsky, and Christopher Potts. Mechanistic evaluation of transformers and state space models. *arXiv preprint arXiv:2505.15105*, 2025.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv preprint arXiv:2402.18668*, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, and Xiaoli Li. Tslanet: Rethinking transformers for time series representation learning. *arXiv preprint arXiv:2404.08472*, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng

Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia,

Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Ning Hua and Wenlian Lu. Basis operator network: A neural network-based model for learning nonlinear operators via neural basis. *Neural Networks*, 164:21–37, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, volume 2026, pp. 86173–86202, 2026.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pp. 4296–4313, 2022.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: an investigation on affine mapping. *Academia AI and Applications*, 2(2), 2026.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Y Nie. A time series is worth 64words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Rafael Josip Penić, Tin Vlašić, Roland G Huber, Yue Wan, and Mile Šikić. Rinalmo: general-purpose rna language models can generalize well on structure prediction tasks. *Nature Communications*, 16(1):5671, 2025.

Michael Poli, Stefano Massaroli, Federico Berto, Jinkyoo Park, Tri Dao, Christopher Ré, and Stefano Ermon. Transform once: Efficient operator learning in frequency domain. *Advances in Neural Information Processing Systems*, 35:7947–7959, 2022.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

David Vigouroux, Joseba Dalmau, Louis Béthune, and Victor Boutin. Deep sturm–liouville: From sample-based to 1d regularization with learnable orthogonal basis functions. *arXiv preprint arXiv:2504.07151*, 2025.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 548–558. IEEE, 2021.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Yuhan Wu, Xiyu Meng, Huajin Hu, Junru Zhang, Yabo Dong, and Dongming Lu. Affirm: Interactive mamba with adaptive fourier filters for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21599–21607, 2025.

Zexue Wu, Yifeng Gong, and Aoqian Zhang. Dtmamba: Dual twin mamba for time series forecasting. *arXiv preprint arXiv:2405.07022*, 2024.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37:115491–115522, 2024.

Xihang Yue, Yi Yang, and Linchao Zhu. Holistic physics solver: Learning pdes in a unified spectral-physical space. *arXiv preprint arXiv:2410.11382*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Zecheng Zhang, Wing Tat Leung, and Hayden Schaeffer. Belnet: Basis enhanced learning, a mesh-free neural operator, 2022. URL https://arxiv.org/abs/2212.07336.

Xuxiang Zhao and Angelica I. Aviles-Rivero. Don't fix the basis – learn it: Spectral representation with adaptive basis learning for pdes, 2026. URL https://arxiv.org/abs/2605.10451.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022a.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022b.

Chenchen Zhu, Jingyan Wu, Han Sun, Francesca Briganti, Benjamin Meder, Wu Wei, and Lars M Steinmetz. Single-molecule, full-length transcript isoform sequencing reveals disease-associated rna isoforms in cardiomyocytes. *Nature communications*, 12(1):4203, 2021.

A Additional Theoretical Analysis of SLiK

This section establishes the mathematical properties used by the Sturm–Liouville spectral memory in SLiK. We distinguish properties of the exact continuous and discrete operators from properties of the finite-step Ritz approximation used in the implementation. The continuous and discrete positivity statements apply to the full model, for which p, q, and w are strictly positive. The special case $q \equiv 0$, used in one ablation, is positive semidefinite and is discussed separately below.

A.1 Notation and weighted geometry

For a sequence of length $T \geq 2$, let $h = 1/(T-1)$ and $t_i = i/(T-1)$ for $i = 0, \ldots, T-1$. The coefficient fields are sampled as $p_{i+\frac{1}{2}} > 0$ on edges and $q_i, w_i > 0$ on nodes. We denote the discrete stiffness–potential matrix and mass matrix by $\mathbf{A}, \mathbf{M} \in \mathbb{R}^{T\times T}$, respectively. For vectors, the weighted inner product and norm are

$$\langle \mathbf{x}, \mathbf{y}\rangle_{\mathbf{M}} = \mathbf{x}^\top \mathbf{M}\mathbf{y}, \qquad \|\mathbf{x}\|_{\mathbf{M}}^2 = \mathbf{x}^\top \mathbf{M}\mathbf{x}. \tag{31}$$

For a feature matrix $\mathbf{V} \in \mathbb{R}^{T\times D}$, the corresponding weighted Frobenius norm is

$$\|\mathbf{V}\|_{\mathbf{M},F}^2 = \mathrm{tr}(\mathbf{V}^\top \mathbf{M}\mathbf{V}). \tag{32}$$

Throughout the theoretical analysis, eigenpairs are ordered as $0 < \lambda_1 \leq \cdots \leq \lambda_T$ unless the semidefinite $q \equiv 0$ case is stated explicitly.

A.2 CONTINUOUS OPERATOR PROPERTIES

Consider the regular Sturm–Liouville operator

$$\mathcal{T}u = \frac{1}{w(t)}\left[-\frac{\mathrm{d}}{\mathrm{d}t}\left(p(t)\frac{\mathrm{d}u}{\mathrm{d}t}\right) + q(t)u\right] \tag{33}$$

on $[0,1]$, with the natural zero-flux boundary conditions $p(0)u'(0) = p(1)u'(1) = 0$. The coordinate network is continuous, and the softplus parameterization gives $p(t) > 0$, $q(t) > 0$, and $w(t) > 0$. On the compact interval $[0,1]$, these coefficient functions therefore have finite positive lower and upper bounds.

Proposition A.1 (continuous well-posedness). The operator $\mathcal{T}$ is positive and self-adjoint in $L^2_w([0,1])$ with inner product $\langle u, v\rangle_w = \int_0^1 w(t)u(t)v(t)\,\mathrm{d}t$. Its eigenvalues are real and positive, and its eigenfunctions may be chosen as a complete w-orthonormal basis.

Proof. For functions satisfying the zero-flux boundary conditions, integration by parts gives

$$\langle u, \mathcal{T}v\rangle_w = \int_0^1 \left[p(t)u'(t)v'(t) + q(t)u(t)v(t)\right]\,\mathrm{d}t \tag{34}$$

$$= \langle \mathcal{T}u, v\rangle_w. \tag{35}$$

Thus, $\mathcal{T}$ is symmetric. Moreover, for nonzero u,

$$\langle u, \mathcal{T}u\rangle_w = \int_0^1 \left[p(t)|u'(t)|^2 + q(t)|u(t)|^2\right]\,\mathrm{d}t > 0. \tag{36}$$

The regular Sturm–Liouville problem on a compact interval has compact resolvent. The spectral theorem therefore gives a discrete positive spectrum and a complete w-orthonormal eigenbasis. □

Semidefinite ablation. If $q \equiv 0$ under the same zero-flux boundary conditions, constant functions belong to the nullspace. The operator remains self-adjoint and nonnegative, but its first eigenvalue is zero rather than strictly positive. Accordingly, the strict-positivity part of Proposition A.1 should not be applied to the `zero_q` or fully classical Neumann-Laplacian ablations.

A.3 DISCRETE GENERALIZED EIGENPROBLEM

Let $\mathbf{D} \in \mathbb{R}^{(T-1)\times T}$ be the first-difference matrix, so that $(\mathbf{D}\mathbf{x})_i = x_{i+1} - x_i$. Define

$$\mathbf{P} = \operatorname{diag}\left(p_{\frac{1}{2}}/h, \ldots, p_{T-\frac{3}{2}}/h\right), \qquad \mathbf{Q} = h\operatorname{diag}(q_0, \ldots, q_{T-1}). \tag{37}$$

The tridiagonal matrix implemented by SLiK admits the factorization

$$\mathbf{A} = \mathbf{D}^\top\mathbf{P}\mathbf{D} + \mathbf{Q}, \qquad \mathbf{M} = h\operatorname{diag}(w_0, \ldots, w_{T-1}). \tag{38}$$

The explicit entries and endpoint convention are given in Appendix B.

Proposition A.2 (discrete positive definiteness). For the full model with $q_i > 0$ and $w_i > 0$, both $\mathbf{A}$ and $\mathbf{M}$ are symmetric positive definite for every $T \geq 2$.

Proof. For any nonzero $\mathbf{x} \in \mathbb{R}^T$,

$$\mathbf{x}^\top\mathbf{A}\mathbf{x} = \frac{1}{h}\sum_{i=0}^{T-2} p_{i+\frac{1}{2}}(x_{i+1} - x_i)^2 + h\sum_{i=0}^{T-1} q_i x_i^2 > 0, \tag{39}$$

$$\mathbf{x}^\top\mathbf{M}\mathbf{x} = h\sum_{i=0}^{T-1} w_i x_i^2 > 0. \tag{40}$$

Symmetry follows from Eq. equation 38, and strict positivity follows from $q_i > 0$, $w_i > 0$, and $\mathbf{x} \neq \mathbf{0}$. □

Corollary A.3 (real positive generalized spectrum). The generalized eigenproblem

$$\mathbf{A}\mathbf{\Phi} = \mathbf{M}\mathbf{\Phi}\mathbf{\Lambda} \tag{41}$$

has T real positive eigenvalues and an $\mathbf{M}$-orthonormal eigenbasis.

Proof. Define the standardized operator

$$\mathbf{S} = \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{M}^{-\frac{1}{2}}. \tag{42}$$

Proposition A.2 implies that $\mathbf{S}$ is symmetric positive definite. Hence $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ with $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ and $\lambda_k > 0$. Setting $\mathbf{\Phi} = \mathbf{M}^{-\frac{1}{2}}\mathbf{U}$ gives

$$\mathbf{A}\mathbf{\Phi} = \mathbf{M}\mathbf{\Phi}\mathbf{\Lambda}, \qquad \mathbf{\Phi}^\top\mathbf{M}\mathbf{\Phi} = \mathbf{I}. \tag{43}$$

□

If $q_i = 0$ for every node, $\mathbf{A}$ is positive semidefinite and the constant vector spans its nullspace. The generalized eigenvalues are then real and nonnegative, and the remaining weighted spectral construction is unchanged. Any bound containing $1/\lambda_{K+1}$ requires $\lambda_{K+1} > 0$.

A.4 Optimal weighted projection and truncation error

Let $\mathbf{\Phi}_K = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_K]$ contain the first K exact generalized eigenvectors. For one value-feature matrix $\mathbf{V} \in \mathbb{R}^{T\times D}$, define

$$\mathbf{C}_K = \mathbf{\Phi}_K^\top\mathbf{M}\mathbf{V}, \qquad \widehat{\mathbf{V}}_K = \mathbf{\Phi}_K\mathbf{C}_K. \tag{44}$$

Proposition A.4 (best approximation in the retained eigenspace). The reconstruction $\widehat{\mathbf{V}}_K$ is the unique minimizer of

$$\min_{\mathbf{Y}\in\mathbb{R}^{K\times D}} \|\mathbf{V} - \mathbf{\Phi}_K\mathbf{Y}\|^2_{\mathbf{M},F}. \tag{45}$$

Proof. Weighted orthogonality gives $\mathbf{\Phi}_K^\top\mathbf{M}(\mathbf{V} - \widehat{\mathbf{V}}_K) = \mathbf{0}$. Hence, for any $\mathbf{Y}$,

$$\|\mathbf{V} - \mathbf{\Phi}_K\mathbf{Y}\|^2_{\mathbf{M},F} = \|\mathbf{V} - \widehat{\mathbf{V}}_K\|^2_{\mathbf{M},F} \tag{46}$$

$$+ \|\mathbf{\Phi}_K(\mathbf{C}_K - \mathbf{Y})\|^2_{\mathbf{M},F} \tag{47}$$

$$= \|\mathbf{V} - \widehat{\mathbf{V}}_K\|^2_{\mathbf{M},F} + \|\mathbf{C}_K - \mathbf{Y}\|^2_F. \tag{48}$$

The second term is uniquely minimized by $\mathbf{Y} = \mathbf{C}_K$. □

Proposition A.5 (exact truncation identity). Let $\mathbf{c}_k = \boldsymbol{\phi}_k^\top\mathbf{M}\mathbf{V}$. Then

$$\|\mathbf{V} - \widehat{\mathbf{V}}_K\|^2_{\mathbf{M},F} = \sum_{k=K+1}^{T} \|\mathbf{c}_k\|_2^2. \tag{49}$$

Proof. Completeness and weighted orthogonality give $\mathbf{V} = \sum_{k=1}^{T} \boldsymbol{\phi}_k\mathbf{c}_k$. Therefore, $\mathbf{V} - \widehat{\mathbf{V}}_K = \sum_{k=K+1}^{T} \boldsymbol{\phi}_k\mathbf{c}_k$, and the result follows from the weighted Parseval identity. □

Proposition A.6 (operator-energy truncation bound). For $K < T$ and $\lambda_{K+1} > 0$,

$$\|\mathbf{V} - \widehat{\mathbf{V}}_K\|^2_{\mathbf{M},F} \le \frac{\mathrm{tr}(\mathbf{V}^\top\mathbf{A}\mathbf{V})}{\lambda_{K+1}}. \tag{50}$$

Proof. The generalized eigendecomposition yields

$$\mathrm{tr}(\mathbf{V}^\top\mathbf{A}\mathbf{V}) = \sum_{k=1}^{T} \lambda_k\|\mathbf{c}_k\|_2^2. \tag{51}$$

Because $\lambda_k \geq \lambda_{K+1}$ for $k > K$,

$$\sum_{k=K+1}^{T} \|\mathbf{c}_k\|_2^2 \leq \lambda_{K+1}^{-1} \sum_{k=K+1}^{T} \lambda_k \|\mathbf{c}_k\|_2^2 \tag{52}$$

$$\leq \lambda_{K+1}^{-1} \operatorname{tr}(\mathbf{V}^\top \mathbf{A}\mathbf{V}). \tag{53}$$

Combining this inequality with Proposition A.5 proves the claim. □

Propositions A.4–A.6 characterize the raw weighted projection and reconstruction, before the learned spectral filter is applied. They do not claim that every input is concentrated in the lowest modes. Instead, the energy bound states the condition under which truncation is accurate: the value features must have low energy under the learned operator relative to λ_{K+1}.

A.5 Properties of the approximate low-mode solver

SLiK avoids a full $T \times T$ eigendecomposition by applying orthogonal subspace iteration to $\mathbf{S}$ and then solving a reduced Rayleigh–Ritz problem. Let K_s denote the oversampled subspace width. At the end of the iteration, $\mathbf{Q} \in \mathbb{R}^{T \times K_s}$ has orthonormal columns. Let $\mathbf{U}_K \in \mathbb{R}^{K_s \times K}$ contain the first K orthonormal eigenvectors of $\mathbf{Q}^\top \mathbf{S}\mathbf{Q}$, and define

$$\widetilde{\mathbf{\Phi}} = \mathbf{M}^{-\frac{1}{2}} \mathbf{Q}\mathbf{U}_K. \tag{54}$$

Proposition A.7 (weighted orthogonality of the Ritz modes). In exact arithmetic, the approximate modes satisfy

$$\widetilde{\mathbf{\Phi}}^\top \mathbf{M} \widetilde{\mathbf{\Phi}} = \mathbf{I}. \tag{55}$$

Proof. Using $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$ and $\mathbf{U}_K^\top \mathbf{U}_K = \mathbf{I}$,

$$\widetilde{\mathbf{\Phi}}^\top \mathbf{M} \widetilde{\mathbf{\Phi}} = \mathbf{U}_K^\top \mathbf{Q}^\top \mathbf{M}^{-\frac{1}{2}} \mathbf{M} \mathbf{M}^{-\frac{1}{2}} \mathbf{Q}\mathbf{U}_K \tag{56}$$

$$= \mathbf{U}_K^\top \mathbf{Q}^\top \mathbf{Q}\mathbf{U}_K = \mathbf{I}. \tag{57}$$

□

Thus, weighted orthogonality is preserved by the Ritz construction in exact arithmetic. Finite-step approximation affects the eigen-equation residual and the distance to the exact low-eigenvalue subspace, while floating-point arithmetic introduces the measured orthogonality error.

Proposition A.8 (conditional subspace convergence). Let $0 < \lambda_1 \leq \cdots \leq \lambda_T < \alpha$, define $\mathbf{G} = \mathbf{I} - \mathbf{S}/\alpha$, and suppose that $K_s < T$, $\lambda_{K_s} < \lambda_{K_s+1}$. If the initial DCT subspace has full-rank overlap with the first K_s eigenvectors of $\mathbf{S}$, then orthogonal iteration converges to the lowest-K_s eigenspace of $\mathbf{S}$ with asymptotic factor

$$\rho_s = \frac{1 - \lambda_{K_s+1}/\alpha}{1 - \lambda_{K_s}/\alpha} < 1. \tag{58}$$

Proof sketch. The matrices $\mathbf{G}$ and $\mathbf{S}$ share eigenvectors, and the eigenvalues of $\mathbf{G}$ are $\mu_i = 1 - \lambda_i/\alpha$. Therefore, $\mu_1 \geq \cdots \geq \mu_T > 0$, and the dominant K_s-dimensional eigenspace of $\mathbf{G}$ is the lowest-K_s eigenspace of $\mathbf{S}$. Standard orthogonal-iteration analysis bounds the tangent of the largest principal angle by a constant multiple of $(\mu_{K_s+1}/\mu_{K_s})^j$ after j iterations, yielding Eq. equation 58. The constant depends on the initial subspace overlap. □

This proposition is conditional rather than a uniform finite-step guarantee. Convergence can be slow when the relevant eigengap is small. When $K_s = T$, the subspace is full and no subspace-convergence argument is required; the remaining error is numerical. The implementation uses a fixed number of iterations and therefore reports the eigen-equation residual explicitly.

B Discretization of the Sturm–Liouville Operator

B.1 Coordinate network and positive coefficient fields

We use the normalized grid

$$t_i = ih, \qquad i = 0, \dots, T-1, \qquad h = \frac{1}{T-1}, \tag{59}$$

and edge midpoints $t_{i+1/2} = (t_i + t_{i+1})/2$. The coordinate frequencies $\{f_j\}_{j=1}^{F}$ are stored independently of the sequence resolution. The coefficient network g_θ contains two hidden linear layers with $\tanh$ activations and a three-dimensional output layer. A final $\tanh$ bounds its three outputs. The coordinate features and bounded perturbations are

$$\boldsymbol{\psi}(t) = \left[2t - 1, \{\sin(\pi f_j t), \cos(\pi f_j t)\}_{j=1}^{F}\right], \tag{60}$$

$$\boldsymbol{\delta}_\theta(t) = \tanh(g_\theta(\boldsymbol{\psi}(t))). \tag{61}$$

We evaluate the first output at edge midpoints and the second and third outputs at nodes:

$$p_{i+\frac{1}{2}} = \mathrm{softplus}\Big(\bar{p} + s\delta_{\theta,p}(t_{i+\frac{1}{2}})\Big) + \epsilon, \tag{62}$$

$$q_i = \mathrm{softplus}(\bar{q} + s\delta_{\theta,q}(t_i)) + \epsilon, \tag{63}$$

$$\widetilde{w}_i = \mathrm{softplus}(\bar{w} + s\delta_{\theta,w}(t_i)) + \epsilon, \qquad w_i = \frac{\widetilde{w}_i}{T^{-1}\sum_{j=0}^{T-1}\widetilde{w}_j}. \tag{64}$$

The softplus parameterization guarantees positivity, and normalizing w removes the redundant global mass scale without restricting its spatial shape.

The final output-layer weights of g_θ are initialized with a small zero-mean Gaussian distribution and its bias is initialized to zero. The operator therefore begins close to a constant-coefficient system while all coordinate-network layers can receive gradients from the first optimization step. Casting the network parameters to the solver dtype is differentiable, so the outer model may use reduced-precision activations while the eigensystem is constructed in higher precision.

B.2 Tridiagonal stiffness and diagonal mass matrices

Let $\mathbf{A} \equiv \mathbf{A}_{\theta,T}$ and $\mathbf{M} \equiv \mathbf{M}_{\theta,T}$. The diagonal entries of $\mathbf{A}$ are

$$A_{ii} = \begin{cases} q_0 h + p_{1/2}/h, & i = 0, \\ q_i h + (p_{i-1/2} + p_{i+1/2})/h, & 1 \le i \le T-2, \\ q_{T-1} h + p_{T-3/2}/h, & i = T-1, \end{cases} \tag{65}$$

and the off-diagonal entries are

$$A_{i,i+1} = A_{i+1,i} = -\frac{p_{i+1/2}}{h}, \qquad i = 0, \dots, T-2. \tag{66}$$

The mass matrix is diagonal,

$$\mathbf{M} = \mathrm{diag}(w_0 h, \dots, w_{T-1} h). \tag{67}$$

This construction uses only $\mathcal{O}(T)$ stored matrix entries. Positivity of p, q, and w yields a symmetric positive operator and a positive mass matrix under the implemented endpoint convention.

For numerical solution, we convert the generalized problem to the symmetric standard problem

$$\mathbf{S} = \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}, \qquad \mathbf{S}\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda}, \qquad \boldsymbol{\Phi} = \mathbf{M}^{-1/2}\mathbf{U}. \tag{68}$$

The matrix $\mathbf{S}$ remains symmetric tridiagonal, with

$$S_{ii} = \frac{A_{ii}}{M_{ii}}, \qquad S_{i,i+1} = \frac{A_{i,i+1}}{\sqrt{M_{ii}M_{i+1,i+1}}}. \tag{69}$$

If $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, then $\boldsymbol{\Phi}^\top\mathbf{M}\boldsymbol{\Phi} = \mathbf{I}$ follows directly.

C Differentiable Low-Mode Eigensolver

We approximate the lowest $K_T = \min(K, T)$ eigenmodes using an oversampled subspace of size

$$K_s = \min(T, K_T + r), \tag{70}$$

where r is the oversampling parameter. The initial subspace is formed from orthonormal DCT-II modes,

$$[\mathbf{Q}_0]_{ik} = a_k \cos\left(\frac{\pi(i+1/2)k}{T}\right), \tag{71}$$

where $a_0 = T^{-1/2}$ and $a_k = (2/T)^{1/2}$ for $k > 0$. This initialization is close to the low-frequency eigenspace of a constant-coefficient operator.

We obtain a conservative spectral upper bound from the tridiagonal Gershgorin radii,

$$\alpha = 1.05 \max_i \left(S_{ii} + |S_{i,i-1}| + |S_{i,i+1}|\right) + \epsilon, \tag{72}$$

where missing endpoint entries are zero. The bound is detached from the autograd graph to avoid gradients through the scale estimate. Starting from $\mathbf{Q}_0$, we perform J orthogonal subspace iterations

$$\mathbf{Q}_{j+1} = \mathrm{qf}\left(\mathbf{Q}_j - \frac{\mathbf{S}\mathbf{Q}_j}{\alpha}\right), \qquad j = 0, \ldots, J-1, \tag{73}$$

where qf denotes the reduced QR orthonormal factor. Applying $\mathbf{I} - \mathbf{S}/\alpha$ makes the lowest eigenvalues the dominant eigenvalues of the iteration operator.

The final Rayleigh–Ritz matrix is

$$\mathbf{R}_{\mathrm{small}} = \frac{1}{2}\left(\mathbf{Q}_J^\top \mathbf{S}\mathbf{Q}_J + \mathbf{Q}_J^\top \mathbf{S}^\top \mathbf{Q}_J\right). \tag{74}$$

After computing $\mathbf{R}_{\mathrm{small}}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}$, we retain the K_T smallest Ritz pairs and set

$$\mathbf{U} = \mathbf{Q}_J \mathbf{V}_{:,1:K_T}, \qquad \mathbf{\Phi} = \mathbf{M}^{-1/2}\mathbf{U}. \tag{75}$$

QR factorization, the reduced eigendecomposition, and all matrix products are differentiable with respect to the coefficient-network parameters. During training, the eigensystem is therefore rebuilt on every forward pass so that a fresh autograd graph is available.

The reference implementation performs the eigensolve in float64 by default and casts $\mathbf{\Lambda}$, $\mathbf{\Phi}$, and $\mathbf{M}$ back to the activation dtype. This is useful because the tridiagonal entries grow with the grid resolution and low-mode Ritz values can otherwise suffer from cancellation at large T. A lower-precision option is available when throughput is preferred over the most accurate low-mode solution.

C.1 Numerical diagnostics

We monitor weighted orthogonality using

$$e_{\mathrm{orth}} = \left\|\mathbf{\Phi}^\top \mathbf{M}\mathbf{\Phi} - \mathbf{I}\right\|_{\max}. \tag{76}$$

With $\mathbf{U} = \mathbf{M}^{1/2}\mathbf{\Phi}$, the relative eigen-equation residual is

$$e_{\mathrm{eig}} = \frac{\|\mathbf{S}\mathbf{U} - \mathbf{U}\mathbf{\Lambda}\|_F}{\|\mathbf{S}\mathbf{U}\|_F + \epsilon}. \tag{77}$$

These metrics diagnose the numerical eigensystem; they are not forecasting or classification performance metrics.

D Basis Equivariance of the Spectral Filter

Eigenvectors are not unique within a repeated eigenspace. Let $\mathbf{Q} \in \mathbb{R}^{K_T \times K_T}$ be orthogonal and commute with $\mathbf{\Lambda}$, so that it only rotates modes sharing the same eigenvalue. The alternative eigenbasis $\mathbf{\Phi}' = \mathbf{\Phi}\mathbf{Q}$ produces coefficients

$$\mathbf{C}' = \mathbf{\Phi}'^\top \mathbf{M}\mathbf{V} = \mathbf{Q}^\top \mathbf{C}. \tag{78}$$

Because the response gates depend only on the corresponding eigenvalues, each $\mathbf{G}_r$ commutes with $\mathbf{Q}$. Channel maps act only on the last dimension and therefore also commute with a mode-axis rotation. In addition, the dropout mask is shared across modes. For a fixed shared mask (and exactly during evaluation, when dropout is disabled), the pre-normalization spectral update consequently satisfies

$$\mathcal{F}_0(\mathbf{Q}^\top \mathbf{C}, \mathbf{\Lambda}) = \mathbf{Q}^\top \mathcal{F}_0(\mathbf{C}, \mathbf{\Lambda}). \tag{79}$$

The spectral normalization is

$$\mathcal{N}_{\mathrm{spec}}(\mathbf{U}) = \frac{\mathbf{U}}{\sqrt{(K_T D)^{-1} \|\mathbf{U}\|_F^2 + \epsilon}} \odot \mathbf{a}, \tag{80}$$

where $\mathbf{a} \in \mathbb{R}^D$ is a learned channel scale. Orthogonal rotations preserve the Frobenius norm, and the channel scale commutes with mode-axis rotations. Hence the full filter obeys

$$\mathcal{F}(\mathbf{Q}^\top \mathbf{C}, \mathbf{\Lambda}) = \mathbf{Q}^\top \mathcal{F}(\mathbf{C}, \mathbf{\Lambda}). \tag{81}$$

Reconstruction is therefore invariant to the admissible basis choice:

$$\mathbf{\Phi}' \mathcal{F}(\mathbf{C}', \mathbf{\Lambda}) = \mathbf{\Phi}\mathbf{Q}\mathbf{Q}^\top \mathcal{F}(\mathbf{C}, \mathbf{\Lambda}) = \mathbf{\Phi}\mathcal{F}(\mathbf{C}, \mathbf{\Lambda}). \tag{82}$$

This argument applies to exact rotations within repeated eigenspaces. For distinct eigenvalues, the ordered eigenvectors are fixed up to sign; for nearly repeated eigenvalues, eigenvalue-conditioned gates vary smoothly but do not imply exact rotation invariance.

E Variable-Length and Mixed-Length Evaluation

For a dense batch whose samples all have length T, SLiK constructs one eigensystem $\mathcal{E}_\theta(T)$ and reuses it across the batch and all blocks. For a prefix-padded batch, let $\boldsymbol{\ell} = (T_1, \ldots, T_B)$ contain the true lengths and let $\mathcal{U}$ be the set of distinct lengths. For every $u \in \mathcal{U}$, we perform the following operations:

1. select the indices $\mathcal{I}_u = \{b : T_b = u\}$;
2. crop the selected samples to $\mathbf{X}_{\mathcal{I}_u,:u,:}$;
3. construct $\mathcal{E}_\theta(u)$ and process the cropped group;
4. pad the returned features with zeros to the batch width and restore their original sample indices.

This grouped computation is identical, up to floating-point error, to processing each valid prefix independently. Changing values exclusively in a padded suffix cannot affect a valid output or a length-aware pooled prediction. For classification, length-aware mean pooling is

$$\overline{\mathbf{z}}_b = \frac{1}{T_b} \sum_{t=1}^{T_b} \mathbf{z}_{b,t}. \tag{83}$$

The current procedure supports prefix padding, not arbitrary masks with missing internal tokens. It also constructs one eigensystem per distinct length; a batch containing many unique lengths can therefore incur additional basis-building overhead.

E.1 Evaluation cache

During no-gradient evaluation, eigensystems are cached by sequence length, device, activation dtype, and the internal version counters of the operator parameters. The cache is invalidated when the module changes between training and evaluation modes or moves across devices or dtypes. A least-recently-used policy limits the number of cached entries. Training outputs are never cached because reusing them would reuse a stale autograd graph. Accordingly, reported latency should distinguish first-call latency, which includes basis construction, from steady-state latency for a cached length.

Table 6: Leading forward complexity of the principal SLiK operations. Fixed architectural constants are suppressed.

Operation	Time complexity
Coefficient-field sampling	$\mathcal{O}(T(FH_c + H_c^2))$
Tridiagonal assembly	$\mathcal{O}(T)$
J subspace iterations	$\mathcal{O}(JTK_s^2)$
Reduced eigendecomposition	$\mathcal{O}(K_s^3)$
Projection and reconstruction per block	$\mathcal{O}(BTK_TD)$
Spectral response bank per block	$\mathcal{O}(BK_TRD^2)$
Local, pointwise, and FFN paths per block	$\mathcal{O}(BTD^2)$

F Detailed Computational Complexity

We distinguish basis construction, which is performed once per distinct length, from the operations performed inside each SLiK block. Let F be the number of coordinate frequencies, H_c the coefficient-network width, $K_T = \min(K, T)$ the number of retained modes, r the oversampling size, $K_s = \min(T, K_T + r)$ the solver subspace size, J the number of subspace iterations, R the number of spectral responses, D the hidden dimension, L the number of blocks, and B the batch size.

Table 6 summarizes the leading forward costs. The basis is shared across the batch and across all L blocks for a dense equal-length call.

F.1 Basis construction

The coordinate network is evaluated on T nodes and $T - 1$ edges. Its exact leading cost is

$$\mathcal{O}\big(T\left(FH_c + H_c^2\right)\big), \tag{84}$$

which becomes $\mathcal{O}(TH_c^2)$ when F is fixed and no larger than the hidden width. Constructing the tridiagonal entries costs $\mathcal{O}(T)$.

A multiplication between the tridiagonal matrix $\mathbf{S}$ and a $T \times K_s$ subspace costs $\mathcal{O}(TK_s)$. The reduced QR factorization in each iteration costs $\mathcal{O}(TK_s^2)$. After J iterations, forming the Rayleigh–Ritz matrix costs $\mathcal{O}(TK_s^2)$ and diagonalizing it costs $\mathcal{O}(K_s^3)$. The total basis cost is therefore

$$\mathcal{O}\big(T(FH_c + H_c^2) + JTK_s^2 + K_s^3\big). \tag{85}$$

The $\mathcal{O}(TK_s)$ tridiagonal products are dominated by QR when $K_s > 1$ and are omitted from Eq. equation 85.

F.2 SLiK blocks

Weighted projection and reconstruction each cost $\mathcal{O}(BTK_TD)$. The R channel maps in the spectral filter cost $\mathcal{O}(BK_TRD^2)$, while the small gate network contributes $\mathcal{O}(K_TH_gR + K_TH_g)$ for gate width H_g. Value projection, output projection, pointwise local fusion, and the feed-forward network jointly cost $\mathcal{O}(BTD^2)$ when the number of local branches and the feed-forward ratio are fixed. The depthwise local convolutions additionally cost

$$\mathcal{O}\left(BTD \sum_{\kappa \in \mathcal{K}} \kappa\right). \tag{86}$$

Thus, suppressing fixed architectural constants, L blocks require

$$\mathcal{O}\big(LB\left(TK_TD + K_TRD^2 + TD^2\right)\big). \tag{87}$$

Combining Eqs. equation 85 and Eq. equation 87 gives

$$\mathcal{O}\big(T(FH_c + H_c^2) + JTK_s^2 + K_s^3 + LB\left(TK_TD + K_TRD^2 + TD^2\right)\big). \tag{88}$$

Algorithm 1 Length-adaptive Sturm–Liouville basis construction

Require: Length $T \geq 2$; maximum modes K; oversampling r; coordinate network g_θ; raw bases $\bar{p}, \bar{q}, \bar{w}$; scale s; stabilizer ϵ; iterations J
Ensure: Eigenvalues $\boldsymbol{\lambda}$, weighted basis $\boldsymbol{\Phi}$, mass diagonal $\mathbf{m}$
$t_i \leftarrow i/(T-1)$ for $i = 0, \ldots, T-1$
$t_{i+\frac{1}{2}} \leftarrow (t_i + t_{i+1})/2$ for $i = 0, \ldots, T-2$
Evaluate node and edge features $\boldsymbol{\psi}(t)$ and bounded perturbations $\boldsymbol{\delta}_\theta(t)$
$p_{i+\frac{1}{2}} \leftarrow \mathrm{softplus}(\bar{p} + s\delta_{\theta,p}(t_{i+\frac{1}{2}})) + \epsilon$
$q_i \leftarrow \mathrm{softplus}(\bar{q} + s\delta_{\theta,q}(t_i)) + \epsilon$
$\widetilde{w}_i \leftarrow \mathrm{softplus}(\bar{w} + s\delta_{\theta,w}(t_i)) + \epsilon$
$w_i \leftarrow \widetilde{w}_i / \max(T^{-1}\sum_j \widetilde{w}_j, \epsilon)$
$h \leftarrow 1/(T-1)$ and $m_i \leftarrow h w_i$
Assemble the diagonal $\mathbf{a}$ and off-diagonal $\mathbf{b}$ of $\mathbf{A}$ using Eqs. equation 65 and equation 66
$d_i \leftarrow a_i/m_i$ and $e_i \leftarrow b_i/\sqrt{m_i m_{i+1}}$ ▷ $\mathbf{S} = \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$
$K_T \leftarrow \min(K, T)$ and $K_s \leftarrow \min(T, K_T + r)$
Initialize $\mathbf{Q} \in \mathbb{R}^{T \times K_s}$ with the first K_s orthonormal DCT-II modes
$\varrho_i \leftarrow \sum_{j \neq i} |S_{ij}|$
$\alpha \leftarrow \mathrm{stopgrad}(1.05 \max_i(d_i + \varrho_i)) + \epsilon$
for $j = 1, \ldots, J$ **do**
    $\mathbf{Q} \leftarrow \mathrm{qf}(\mathbf{Q} - \mathrm{TriDiagMV}(\mathbf{d}, \mathbf{e}, \mathbf{Q})/\alpha)$
end for
$\mathbf{H} \leftarrow \mathrm{sym}\big(\mathbf{Q}^\top \mathrm{TriDiagMV}(\mathbf{d}, \mathbf{e}, \mathbf{Q})\big)$
$(\boldsymbol{\eta}, \mathbf{R}) \leftarrow \mathrm{eigh}(\mathbf{H})$ in ascending order
$\boldsymbol{\lambda} \leftarrow \boldsymbol{\eta}_{1:K_T}$
$\boldsymbol{\Phi} \leftarrow \mathrm{diag}(\mathbf{m})^{-1/2}\mathbf{Q}\mathbf{R}_{:,1:K_T}$
return $(\boldsymbol{\lambda}, \boldsymbol{\Phi}, \mathbf{m})$

For fixed F, H_c, K_s, K_T, J, R, D, L, and B, the forward time is linear in T. This is a conditional long-sequence statement rather than an unconditional one. If the requested number of modes or oversampled subspace grows with T, the QR and reduced eigendecomposition terms grow accordingly. In particular, when $K_s = T$, basis construction can be cubic in T.

F.3 Working memory

Excluding parameters and reverse-mode activation storage, the principal working tensors require

$$\mathcal{O}\big(TK_s + K_s^2 + BTD + BK_TD\big) \tag{89}$$

memory for the solver subspace, reduced problem, token features, and spectral coefficients. With fixed K_s and D, this memory is linear in T. Training retains additional block activations and differentiable eigensolver intermediates; its peak memory is therefore higher than Eq. equation 89. For mixed-length batches, the basis term in Eq. equation 85 is summed over the distinct length groups. For a cached evaluation length, the basis-construction term is amortized across repeated calls.

G Pseudocode for SLiK

Algorithms 1–3 specify the backbone computation. The pseudocode uses the same convention as the implementation: the eigensystem is built once per distinct valid length and reused across all SLiK blocks in that group. External task heads, objectives, and optimizers are not part of the backbone algorithms.

Scope of the pseudocode. Algorithm 1 describes the differentiable training-time construction. The reference implementation may execute the solve in float64 and cast the returned eigensystem to the activation dtype. During no-gradient evaluation, a valid cached eigensystem may replace the call to $\mathrm{SLBasis}(T)$. Algorithm 3 assumes prefix validity and does not represent arbitrary internal masks.

Algorithm 2 Eigenvalue-conditioned Sturm–Liouville spectral memory

Require: Features $\mathbf{Z} \in \mathbb{R}^{B\times T\times D}$; eigenvalues $\boldsymbol{\lambda}$; basis $\boldsymbol{\Phi}$; mass diagonal $\mathbf{m}$; response count R
Ensure: Global update $\mathbf{S}_{\text{out}} \in \mathbb{R}^{B\times T\times D}$

$\mathbf{V} \leftarrow \mathbf{Z}\mathbf{W}_v$
$\mathbf{C} \leftarrow \boldsymbol{\Phi}^\top \operatorname{diag}(\mathbf{m})\mathbf{V}$ ▷ Weighted projection
for $k = 1, \dots, K_T$ **do**
  $\xi_k \leftarrow \log(1 + \max(\lambda_k, 0))/\max(\log(1 + \max(\lambda_{K_T}, 0)), 10^{-6})$
  $\mathbf{g}_k \leftarrow \sigma(f_{\text{gate}}(\xi_k))$
end for
$\Delta\mathbf{C} \leftarrow \mathbf{0}$
for $r = 1, \dots, R$ **do**
  $\Delta\mathbf{C} \leftarrow \Delta\mathbf{C} + \mathbf{G}_r\mathbf{C}\mathbf{W}_r$ ▷ $\mathbf{G}_r = \operatorname{diag}(g_{1r}, \dots, g_{K_T,r})$
end for
$\Delta\mathbf{C} \leftarrow \Delta\mathbf{C}/\sqrt{R}$
$\widehat{\mathbf{C}} \leftarrow \text{SpecNorm}(\mathbf{C} + \text{ModeSharedDropout}(\Delta\mathbf{C}))$
$\mathbf{S}_{\text{out}} \leftarrow \text{Dropout}\left(\boldsymbol{\Phi}\widehat{\mathbf{C}}\mathbf{W}_o\right)$
return $\mathbf{S}_{\text{out}}$

Algorithm 3 SLiK forward pass with optional prefix lengths

Require: Input $\mathbf{X} \in \mathbb{R}^{B\times T_{\max}\times C_{\text{in}}}$; optional valid lengths $\{T_b\}_{b=1}^{B}$; number of blocks L
Ensure: Length-preserving features $\mathbf{Z}_{\text{out}} \in \mathbb{R}^{B\times T_{\max}\times D}$

if valid lengths are not provided **then**
  Set $T_b \leftarrow T_{\max}$ for every sample
end if
Initialize $\mathbf{Z}_{\text{out}} \leftarrow \mathbf{0}$
for each distinct T in $\{T_b\}_{b=1}^{B}$ **do**
  $\mathcal{I} \leftarrow \{b : T_b = T\}$
  $\mathbf{X}_T \leftarrow \mathbf{X}_{\mathcal{I},1:T,:}$
  $(\boldsymbol{\lambda}, \boldsymbol{\Phi}, \mathbf{m}) \leftarrow \text{SLBasis}(T)$ using Algorithm 1
  $\mathbf{Z} \leftarrow \mathbf{X}_T\mathbf{W}_{\text{in}} + \mathbf{b}_{\text{in}}$
  for $\ell = 1, \dots, L$ **do**
    $\mathbf{Z}_{\text{local}} \leftarrow \text{MultiScaleLocalConv}_\ell(\text{LN}_{\text{local},\ell}(\mathbf{Z}))$
    $\mathbf{Z}_{\text{global}} \leftarrow \text{SLMemory}_\ell(\text{LN}_{\text{spec},\ell}(\mathbf{Z}); \boldsymbol{\lambda}, \boldsymbol{\Phi}, \mathbf{m})$
    $\mathbf{H} \leftarrow \mathbf{Z} + \boldsymbol{\gamma}_{\text{local},\ell} \odot \mathbf{Z}_{\text{local}} + \boldsymbol{\gamma}_{\text{spec},\ell} \odot \mathbf{Z}_{\text{global}}$
    $\mathbf{Z} \leftarrow \mathbf{H} + \boldsymbol{\gamma}_{\text{ffn},\ell} \odot \text{FFN}_\ell(\text{LN}_{\text{ffn},\ell}(\mathbf{H}))$
  end for
  $\mathbf{Z} \leftarrow \text{LN}_{\text{out}}(\mathbf{Z})$
  $\mathbf{Z}_{\text{out}}[\mathcal{I}, 1:T, :] \leftarrow \mathbf{Z}$
end for
return $\mathbf{Z}_{\text{out}}$

H Reference Implementation Details

The reference coefficient network has two hidden layers of width $H_c = 32$ and uses coordinate frequencies $\{1, 2, 4, 8\}$. Unless changed for an experiment, the coefficient modulation scale is $s = 0.1$, the positivity constant is $\epsilon = 10^{-4}$, and the raw coefficient bases are initialized as $\bar{p} = 0$, $\bar{q} = -4$, and $\bar{w} = 0$. The last coefficient-network layer is initialized from $\mathcal{N}(0, (10^{-3})^2)$, with zero bias. The default low-mode solver uses $J = 8$ subspace iterations and an oversampling size of $r = 16$.

Each local branch uses depthwise kernels $\{3, 5, 7\}$ followed by a pointwise fusion convolution. The spectral filter uses $R = 4$ response maps and a gate network of width $H_g = 32$. LayerScale vectors are initialized to 10^{-3}. The hidden dimension D, number of blocks L, number of retained modes K, dropout rate, optimizer, training schedule, patch configuration, and task head are experiment-specific and should be reported with each benchmark rather than inferred from these defaults.

Table 7: Complete time-series ablation results on ETTm2 with a prediction horizon of 96. The best MSE and MAE are highlighted in **bold**, and the second-best values are underlined; ties at the reported precision receive the same emphasis. Inference and basis-build times are reported separately. Basis orthogonality is measured as $\|\mathbf{\Phi}^\top \mathbf{M} \mathbf{\Phi} - \mathbf{I}\|_\infty$, with $\mathbf{M} = \mathbf{I}$ for the generic and DCT bases; "–" denotes an unavailable diagnostic.

Model variant	MSE↓	MAE↓	Params (M)	FLOPs (G)	Inf. (s)	Basis build (s)	Basis orth. err.	SL eig. resid.
Full	**0.152**	**0.257**	0.404846	0.062416	0.004129	0.004894	1.78×10^{-15}	1.24×10^{-7}
w/o sl operator	0.169	0.270	0.353568	0.042865	0.001933	0.000000	–	–
generic learned orthogonal	0.157	0.261	0.404040	0.062416	0.003879	0.000365	8.88×10^{-16}	–
fixed dct	0.162	0.266	0.403368	0.062416	0.004170	0.000317	1.03×10^{-15}	–
frozen sl operator	0.159	0.263	0.404846	0.062416	0.004232	0.005858	1.55×10^{-15}	9.06×10^{-12}
constant coefficients	0.164	0.267	0.404846	0.062416	0.004173	0.004624	2.44×10^{-15}	2.58×10^{-15}
classical sl	0.163	0.266	0.404846	0.062416	0.004122	0.005760	1.55×10^{-15}	2.42×10^{-15}
unit p	0.162	0.265	0.404846	0.062416	0.004059	0.005762	1.55×10^{-15}	2.74×10^{-7}
zero q	0.163	0.267	0.404846	0.062416	0.004089	0.005827	1.11×10^{-15}	1.84×10^{-6}
unit_w	0.163	0.267	0.404846	0.062416	0.004244	0.005787	2.22×10^{-15}	2.78×10^{-7}
w/o eigenvalue conditioning	0.157	0.261	0.404846	0.062416	0.004432	0.005943	1.67×10^{-15}	2.84×10^{-7}
identity spectral filter	0.160	0.262	0.371558	0.054908	0.002602	0.005712	1.22×10^{-15}	3.24×10^{-8}
continuous learned non-SL	0.158	0.262	0.404842	0.062416	0.003890	0.000803	6.66×10^{-16}	–

During training, eigensystems are recomputed on every forward pass. During evaluation, caching is enabled only under no-gradient execution. For numerical comparisons, we report both basis-build time and end-to-end inference time and state whether the latter uses a warm cache. Parameter counts exclude cached eigensystem tensors, which are buffers rather than learned parameters.

I Scope and Limitations of the Current Formulation

The learned coefficient fields are shared across samples. They adapt the spectral geometry to the task and normalized coordinate, but they are not conditioned on the content of an individual input sequence. The truncated eigenspace emphasizes low-order global structure and can omit high-frequency components, motivating the explicit local branch. The implementation handles variable lengths across calls and prefix-padded mixed-length batches, but does not directly support arbitrary internal masks. Finally, the linear-in-T complexity claim assumes a fixed truncated subspace; increasing the number of modes with sequence length changes both the computational and memory scaling.

J More Details of Time Series Forecasting

J.1 More Details of ablation study

The main paper reports a compact comparison of the learned Sturm–Liouville operator, a generic learned orthogonal basis, and fixed spectral controls. Table 7 extends this comparison to the individual coefficient fields, eigenvalue conditioning, and the learned spectral filter. All variants were evaluated on ETTm2 with a prediction horizon of 96. Forecasting errors are reported with parameter count, FLOPs, inference time, basis-build time, basis-orthogonality error, and generalized eigenvalue residual. FLOPs include profiler-supported operations after basis-cache warm-up but exclude QR and eigendecomposition, whose uncached costs are reported separately as basis-build time.

Learned operator and basis. The full model achieved an MSE/MAE of $0.152/0.257$. Removing the entire Sturm–Liouville operator increased MSE by 11.2% and MAE by 5.1%, the largest degradation among the tested variants. The generic control directly optimized a DCT-initialized $T \times K$ matrix and orthonormalized it by reduced QR, while retaining DCT spectral coordinates and unit mass. This control reduced the fixed-DCT errors from $0.162/0.266$ to $0.157/0.261$. The full model further reduced the errors to $0.152/0.257$ with a comparable parameter count. Freezing the Sturm–Liouville operator produced $0.159/0.263$. These comparisons distinguish generic basis adaptation from the additional gain associated with the learned SL parameterization.

Coefficient fields and spectral response. Restricting the operator to constant coefficients or to the classical Sturm–Liouville form yielded MSE values of 0.164 and 0.163, respectively. Fixing

Table 8: Look-back robustness on ETTh2 and ETTm2 with prediction length $S = 96$. Results are reported as mean $\pm$ standard deviation over three paired random seeds.

Dataset	L	Params (M)	MSE	MAE
ETTh2	96	0.220526	0.256952 ± 0.004227	0.341759 ± 0.003847
ETTh2	192	0.294254	0.251441 ± 0.005745	0.341736 ± 0.005099
ETTh2	336	0.404846	$\mathbf{0.243639 \pm 0.004491}$	$\mathbf{0.334395 \pm 0.002593}$
ETTm2	96	0.220526	0.157988 ± 0.000970	0.262540 ± 0.001813
ETTm2	192	0.294254	0.153798 ± 0.000510	0.257117 ± 0.001282
ETTm2	336	0.404846	$\mathbf{0.151867 \pm 0.003192}$	$\mathbf{0.256636 \pm 0.001631}$

individual fields through the `unit_p`, `zero_q`, and `unit_w` variants likewise increased MSE to 0.162–0.163 and MAE to 0.265–0.267. These variants retain the parameter count and FLOPs of the full model, indicating that the gain arises from the learned coefficient parameterization rather than additional model capacity. Removing eigenvalue conditioning produced the strongest ablated result, $0.157/0.261$, whereas an identity spectral filter yielded $0.160/0.262$. Thus, eigenvalue-conditioned responses provide a measurable improvement, and the learned spectral filter adds further accuracy beyond basis construction.

Continuous learned non-SL control. To separate the contribution of the Sturm–Liouville structure from the general benefit of learning a continuous basis, we replaced the SL eigensystem with a parameter-matched coordinate network that directly generates an orthogonal basis. The resulting continuous learned non-SL control achieved an MSE of 0.158 and an MAE of 0.262, improving over the fixed DCT basis ($0.162/0.266$). It also closely matched the directly learned fixed-grid orthogonal basis ($0.157/0.261$), indicating that continuous coordinate parameterization preserves most of the flexibility of a position-specific learned basis while remaining independent of sequence length.

Nevertheless, the complete SL model achieved lower errors of 0.152 MSE and 0.257 MAE. Relative to the continuous non-SL control, this corresponds to reductions of 3.80% and 1.91%, respectively. The two models have nearly identical parameter counts (0.404846M versus 0.404842M) and the same reported forward FLOPs. These results therefore support the contribution of the operator-constrained SL geometry beyond basis continuity, orthogonality, and learnability alone.

Efficiency and numerical diagnostics. The accuracy improvement entails a computational trade-off. Removing the Sturm–Liouville operator reduced the parameter count by 12.7%, FLOPs by 31.3%, and inference time from 0.004129 to 0.001933 seconds. The fixed DCT alternative reduced basis-build time from 0.004894 to 0.000317 seconds but remained less accurate than the learned basis. For every variant with an applicable Sturm–Liouville basis, the weighted orthogonality error remained between 1.03×10^{-15} and 2.44×10^{-15}. Eigenvalue residuals varied more widely and were not monotonic with forecasting error; for example, constant and classical operators had smaller residuals than the full model but worse MSE and MAE. The predictive gains therefore reflect the expressiveness of the learned spectral geometry rather than a numerical loss of orthogonality or residual accuracy.

Robustness to the look-back window. We evaluate SLiK on ETTh2 and ETTm2 with look-back windows $L \in \{96, 192, 336\}$ and a fixed prediction length $S = 96$. As shown in Table 8, SLiK remains stable across different context lengths. Longer context consistently improves performance on ETTh2 and ETTm2. These results indicate that SLiK is not highly sensitive to the look-back window and support the use of the standardized $L = 336$ setting in our main experiments.

J.2 Interpretability

Beyond aggregate forecasting accuracy, we examine whether SLiK behaves consistently with its intended design: learning a dataset-adaptive Sturm–Liouville (SL) operator whose eigenmodes serve as a compact global memory. We organize the analysis as an evidence chain from observable predictions to internal representations and functional interventions. Specifically, we first inspect representative forecasts to characterize the temporal structures captured by the model. We then visual-

ize the learned SL coefficient fields to determine whether the operator adapts to different datasets, and examine spectral occupancy to establish whether the resulting eigenmodes are actively used by the hidden representations. Finally, we perform post-training branch removal to assess whether the spectral pathway makes a functionally important contribution to forecasting. Unless otherwise specified, all analyses are conducted on ETTh2 and ETTm2 with a lookback length of $T = 336$ and a forecasting horizon of $S = 96$.

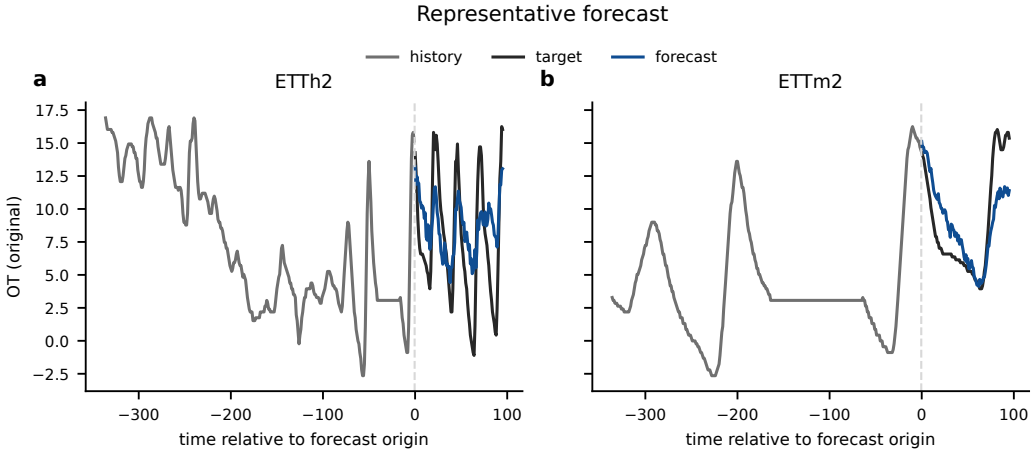

Figure 2: **Representative forecasts on ETTh2 and ETTm2.** Forecasts for a pre-specified test window using a lookback length of $T = 336$ and a prediction length of $S = 96$. Gray curves denote the historical OT observations, black curves denote the ground-truth future values, and blue curves denote the SLiK forecasts. The vertical dashed line marks the forecast origin. **(a)** On ETTh2, SLiK recovers the dominant oscillatory pattern while smoothing some high-amplitude extrema. **(b)** On ETTm2, SLiK captures the long-term decline, trough, and subsequent recovery, although the final rebound is underestimated. Values are shown in the original data scale.

Qualitative forecasting behavior. Figure 2 presents representative forecasts on ETTh2 and ETTm2. On ETTh2, SLiK recovers the dominant high-frequency periodic structure of the target sequence, although the amplitudes of several extrema are attenuated. On ETTm2, the model correctly follows the decline, trough, and subsequent recovery, but underestimates the magnitude of the final rebound. These examples suggest that SLiK captures both oscillatory and slowly varying temporal structures, while sharp changes and extreme amplitudes remain more challenging. As this visualization contains one preselected window per dataset, it is intended as qualitative evidence rather than an aggregate accuracy comparison.

Dataset-adaptive operator geometry. Figure 3 visualizes the learned coefficient fields $p_\theta(t)$, $q_\theta(t)$, and $w_\theta(t)$ that define the SL operator. On ETTh2, p_θ and w_θ exhibit pronounced and approximately complementary changes around the middle of the normalized domain, with coefficients of variation 0.058 and 0.058, respectively, whereas q_θ remains nearly constant (0.004). In contrast, the ETTm2 operator contains smoother but nontrivial modulation across all three fields, with coefficients of variation 0.017, 0.021, and 0.027. Thus, the operator adapts its geometry to the dataset rather than collapsing to one universal constant-coefficient basis. This observation distinguishes the learned SL representation from a fixed Fourier or DCT construction, although the performance benefit of such adaptation is established separately by the operator ablations.

Allocation of spectral memory. The learned operator becomes useful only if the resulting modes are actually occupied by the hidden representations. Figure 4 shows the average energy allocated to each SL mode. ETTh2 is strongly low-mode dominated: the first three modes account for approximately 61% of the total spectral energy, consistent with a compact representation of its dominant periodic structure. ETTm2 exhibits a more distributed allocation, combining low-frequency modes with a pronounced intermediate-mode component around modes 7–9. The two datasets therefore use the same spectral-memory mechanism differently: ETTh2 compresses most information into a

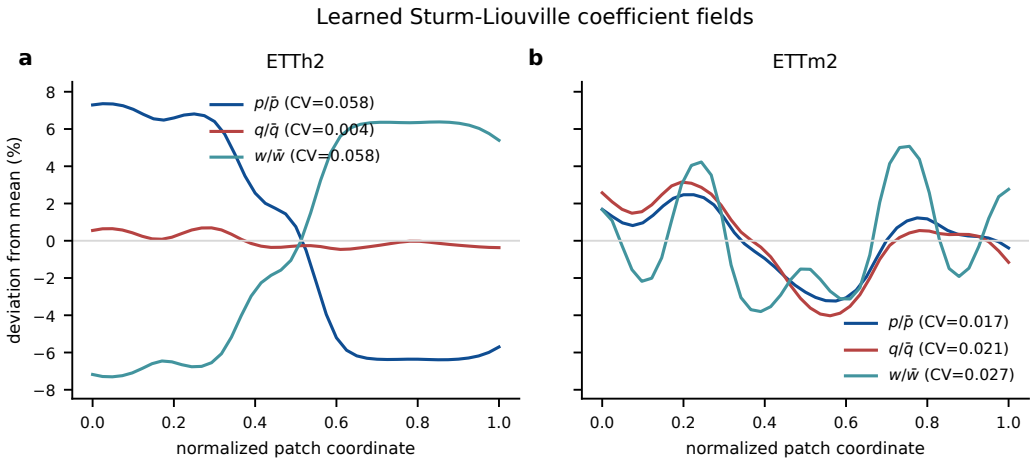

Figure 3: **Dataset-adaptive Sturm–Liouville coefficient fields.** Learned stiffness p, potential q, and mass-density w are plotted as percentage deviations from their respective means over the normalized patch coordinate. The legend reports the coefficient of variation (CV) of each field. **(a)** On ETTh2, p and w develop strong complementary variations around the patch midpoint, whereas q remains nearly constant. **(b)** On ETTm2, all three fields exhibit smoother, multi-scale spatial modulation. The distinct coefficient profiles show that SLiK learns dataset-specific differential operators rather than using a fixed Fourier or cosine basis.

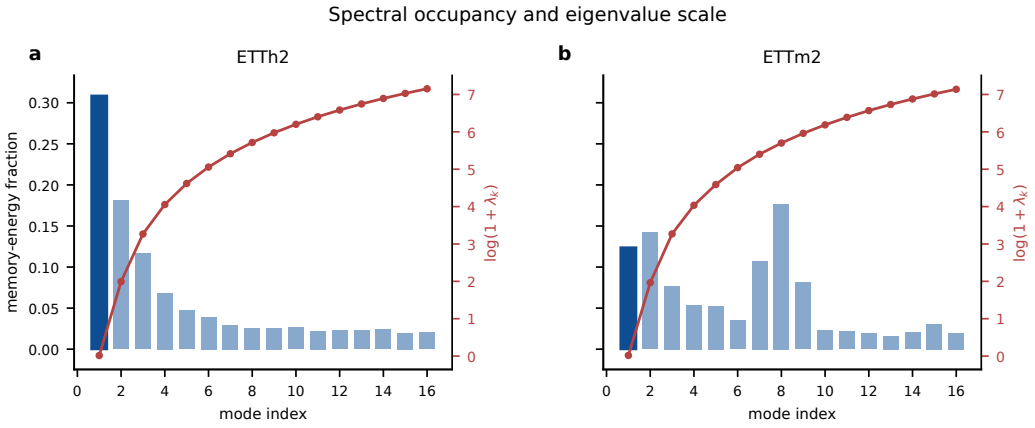

Figure 4: **Spectral-memory occupancy and learned eigenvalue scale.** Bars show the spectral-memory energy of each Sturm–Liouville mode, averaged across SLiK blocks and normalized to sum to one. Red curves show the corresponding eigenvalue coordinate $\log(1 + \lambda_k)$. **(a)** ETTh2 concentrates most memory in the first few modes, consistent with a predominantly low-frequency representation. **(b)** ETTm2 exhibits complementary low- and mid-order occupancy, including a pronounced energy peak around modes 7–9. Thus, the same SLiK architecture allocates its compact spectral memory according to dataset-specific temporal structure.

small low-frequency subspace, whereas ETTm2 retains additional intermediate-scale variation. This dataset-dependent occupancy provides representation-level evidence that the learned eigenspace is actively used rather than ignored.

Functional importance of the spectral branch. We directly evaluate branch-level importance by zeroing the corresponding LayerScale coefficients after training. As shown in Figure 5, removing the spectral branch causes by far the largest increase in standardized test MSE on both datasets. The increase is 0.047 on ETTh2, compared with 0.00835 for the feed-forward branch and an approxi-

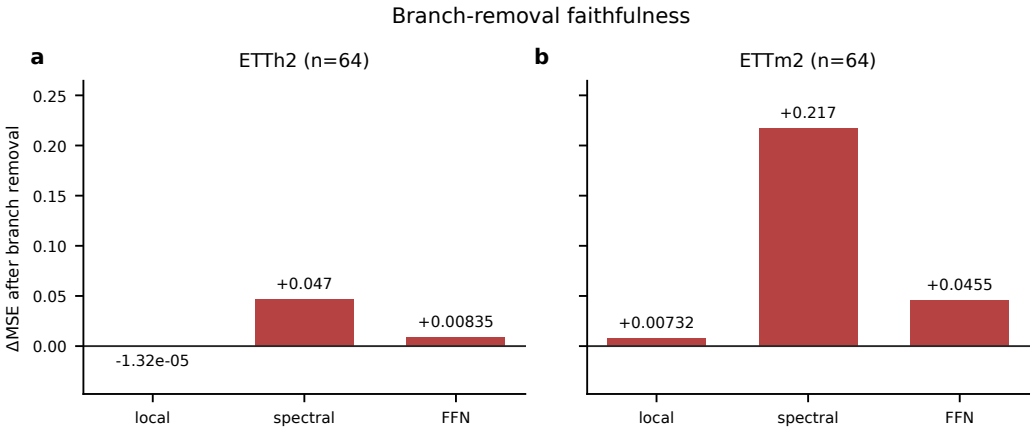

Figure 5: **Forecast degradation under post-training branch removal.** Each branch is disabled after training by setting its layer-scale coefficient to zero, while all remaining parameters are kept fixed. Bars report the change in standardized test MSE relative to the full model, evaluated on the same 64 test windows; positive values indicate performance degradation. Removing the Sturm–Liouville spectral branch produces the largest increase on both **(a)** ETTh2 ($+0.047$) and **(b)** ETTm2 ($+0.217$), substantially exceeding the effects of removing the local or feed-forward branch. This intervention identifies spectral memory as the principal global-information pathway in SLiK.

mately zero change for the local branch. On ETTm2, spectral-branch removal increases MSE by 0.217, compared with 0.0455 and 0.00732 for the feed-forward and local branches, respectively. The spectral degradation is therefore approximately $5.6\times$ the feed-forward degradation on ETTh2 and $4.8\times$ on ETTm2. Combined with the spectral occupancy and residual-update analyses, this intervention identifies the SL spectral branch as the principal global information pathway in the trained model. Because the remaining branches are not retrained after removal, the result measures reliance of the trained network rather than the standalone capacity of each branch.

Summary. Together, the four analyses provide a coherent account of how the learned SL spectral memory operates. The representative forecasts show that SLiK captures the dominant oscillatory and slowly varying structures of both datasets, although sharp extrema can remain attenuated. The learned coefficient fields further show that the SL operator does not collapse to a universal constant-coefficient basis, but instead develops distinct geometries for ETTh2 and ETTm2. Consistent with this adaptation, the spectral-occupancy analysis reveals dataset-dependent allocation of memory: ETTh2 concentrates most of its energy in a compact low-mode subspace, whereas ETTm2 retains a broader combination of low- and intermediate-frequency modes. Most importantly, removing the spectral branch produces the largest degradation in forecasting performance on both datasets, providing direct intervention-based evidence that the learned spectral representation is functionally important to the trained network. These complementary results support the intended role of the SL eigenspace as a compact, adaptive, and task-relevant global memory.

J.3 Sensitivity to the Number of Spectral Modes

To complement the fixed-DCT ablation in the main paper, we examine how the number of retained spectral modes K affects the learned Sturm–Liouville (SL) basis and a fixed DCT basis. All configurations are trained independently using the same random seed and optimization protocol; only the basis type and K are changed. We conduct this diagnostic experiment on ETTm2 with lookback length $T = 336$ and prediction length $S = 96$. The results are reported in the train-standardized space.

As shown in Table 9, the learned SL basis achieves lower MSE and MAE than the fixed DCT basis at every evaluated value of K. The absolute MSE improvement ranges from 0.009 to 0.011, corresponding to a relative reduction of approximately 5.6%–6.8% with respect to the DCT results.

Table 9: Single-seed sensitivity to the number of retained spectral modes K on ETTm2. Bold values indicate the better basis under the same K. Differences are computed as DCT minus learned SL, such that a positive value favors the learned basis. Lower MSE and MAE are better.

	Learned SL		Fixed DCT		Difference	
K	MSE	MAE	MSE	MAE	ΔMSE	ΔMAE
4	**0.156**	**0.260**	0.166	0.269	+0.010	+0.009
8	**0.156**	**0.259**	0.166	0.261	+0.010	+0.002
16	**0.152**	**0.257**	0.161	0.265	+0.009	+0.008
32	**0.151**	**0.256**	0.162	0.267	+0.011	+0.011

Table 10: Needle-in-a-Haystack (NIAH) retrieval accuracy (%) at context lengths of 1,024 and 2,048 tokens. All models contain approximately 180M parameters. The average is computed over all six evaluation settings. Higher is better. Best and second-best results are shown in **bold** and underlined, respectively.

Model	**NIAH-Single-1**		**NIAH-Single-2**		**NIAH-Single-3**		**Avg. ↑**
	1,024	**2,048**	**1,024**	**2,048**	**1,024**	**2,048**	
Transformer-180M	94.6	79.0	73.2	66.6	56.6	29.6	66.6
GDN-180M	94.8	82.8	93.6	84.6	49.2	**31.0**	72.7
Mamba-2-180M	95.4	76.8	87.8	45.0	46.8	21.8	62.3
Mamba-3-SISO-180M	**100.0**	94.2	**100.0**	90.0	53.6	25.4	77.2
Mamba-3-MIMO-180M	98.6	35.4	99.8	46.4	56.2	12.4	58.1
SLiK-180M	**100.0**	**100.0**	99.8	**95.6**	**62.2**	26.0	**80.6**

The MAE improvement ranges from 0.002 to 0.011. Thus, within this single-seed experiment, the advantage of the learned basis is not restricted to one particular spectral rank.

The low-rank results additionally provide evidence for compact spectral representation. With only four learned modes, the model obtains an MSE of 0.156, which is lower than the best DCT MSE of 0.161 among all evaluated values of K. Similarly, the learned four-mode model achieves an MAE of 0.260, compared with the best DCT MAE of 0.261. This observation is consistent with the learned operator concentrating forecasting-relevant temporal structure into a smaller spectral subspace.

Increasing the learned spectral rank from $K = 4$ to $K = 32$ reduces MSE from 0.156 to 0.151 and MAE from 0.260 to 0.256. Most of this improvement is obtained by $K = 16$, while increasing the rank from 16 to 32 provides only an additional 0.001 reduction in both metrics. The DCT results, by contrast, are non-monotonic: its best MSE is obtained at $K = 16$, whereas increasing K to 32 slightly degrades both MSE and MAE. These results motivate our default choice of $K = 16$ as a favorable trade-off between forecasting accuracy and spectral computation.

K More detail for Language model

K.1 Retrieval Capabilities

Beyond standard language modeling, retrieval ability serves as an important criterion for evaluating linear sequence models (Arora et al., 2025; 2024). To evaluate SLiK, we compared our model with baselines on the needle-in-a-haystack (NIAH) task (Hsieh et al., 2024) with our pretrained 180M model from section 4.2

Table 10 evaluates single-needle retrieval at context lengths of 1,024 and 2,048 tokens. SLiK achieves the highest average accuracy of 80.6, outperforming Mamba-3-SISO, the strongest competing model on average, by 3.4 percentage points. On NIAH-Single-1, SLiK maintains perfect retrieval accuracy at both context lengths. It also achieves 99.8 and 95.6 on NIAH-Single-2 and obtains the best NIAH-Single-3 result at the 1,024-token length, reaching 62.2. These results show that SLiK largely preserves retrieval performance when the context length increases from 1,024 to 2,048 tokens on the first two tasks.

Table 11: Effect of the number of SL spectral modes K on NIAH-Single-3. Values denote exact-match accuracy (%). Higher is better. Best and second-best results are shown in **bold** and underlined, respectively.

Spectral modes	1,024 tokens ↑	2,048 tokens ↑
$K = 16$	52.40	16.60
$K = 32$	<u>62.20</u>	<u>26.00</u>
$K = 64$	**68.80**	**32.80**

Table 12: NIAH retrieval accuracy (%) of SLiK-180M at a context length of 4,096 tokens. Each condition contains 500 examples. Higher values are better.

Model	Context length	Single-1	Single-2	Single-3
SLiK-180M	4,096	99.80	95.20	16.60

Performance remains limited in the more challenging NIAH-Single-3 setting. At a context length of 2,048, SLiK accuracy decreases to 26.0, below GDN (31.0) and Transformer (29.6). Thus, although SLiK provides the strongest aggregate retrieval performance, robust retrieval in the most difficult long-context setting remains an important limitation. Motivated by the scaling behavior observed in Mamba-3 (Lahoti et al., 2026), we hypothesize that increasing model capacity and training scale may help alleviate this limitation and improve retrieval robustness in more challenging long-context regimes.

Diagnosing the NIAH-Single-3 degradation. NIAH-Single-3 differs from the first two conditions by requiring the retrieval and exact reproduction of a multi-token UUID rather than a short numeric value. Its exact-match accuracy therefore depends on both locating the relevant span and generating every token of the retrieved value correctly. To test whether the degradation was partly caused by limited spectral capacity, we varied the number of SL spectral modes over $K \in \{16, 32, 64\}$ while keeping the remaining training and evaluation settings fixed. All variants were trained from scratch using the same data order, token budget, optimizer, learning-rate schedule, and random seed. Apart from K, all architectural and evaluation settings were held fixed.

As shown in Table 11, retrieval accuracy improves monotonically with K at both context lengths. Increasing K from 32 to 64 raises accuracy by 6.6 percentage points at 1,024 tokens and by 6.8 percentage points at 2,048 tokens. The disproportionately larger gain at 2,048 tokens is consistent with spectral capacity becoming a more important constraint as the context grows. In particular, increasing the number of retained modes from $K = 32$ to $K = 64$ improves the 2,048-token accuracy from 26.0% to 32.8%, corresponding to a 6.8 percentage-point (26.2% relative) improvement.

Retrieval Extrapolation to 4K Contexts We further evaluated the frozen SLiK-180M checkpoint on the needle-in-a-haystack (NIAH) benchmark (Hsieh et al., 2024) at a context length of 4,096 tokens. The model was pretrained with a maximum context length of 2,048 tokens and received no additional training, fine-tuning, or length-specific parameter updates. For this evaluation, we re-discretized the learned continuous Sturm–Liouville operator on a 4,096-position domain and recomputed the corresponding causal spectral basis. All learned parameters remained fixed. This procedure changes the discretized positional basis while preserving the learned continuous operator, and should therefore be interpreted as operator-domain extrapolation rather than evaluation on an unchanged 2K basis.

We used the official RULER data-generation protocol with the Llama-3.1 tokenizer. Each condition contained 500 examples, generation was limited to 128 tokens, and predictions were evaluated using the standard substring-matching criterion. No input truncation or positional-scaling heuristic was applied.

As shown in Table 12, SLiK retained strong retrieval performance on NIAH-Single-1 and NIAH-Single-2, achieving accuracies of 99.8% and 95.2%, respectively. These results indicate that the frozen continuous operator can preserve simple key–value retrieval after re-discretization from the

Table 13: Throughput scaling across different sequence lengths under a fixed token budget of 32,768 tokens per batch. Throughput is reported in K tokens/s. The best results are highlighted in **bold**, and the second-best results are underlined.Single-GPU steady-state backbone prefill efficiency on one NVIDIA GH200. All models contain approximately 187M parameters and use BF16 activations with FP32 parameters. Batch size is adjusted to maintain 32,768 tokens per step. For each sequence length T, the continuous SL operator is discretized on a length-specific T-point grid over the fixed normalized domain $[0, 1]$, and the resulting eigensystem is cached after its first construction.

Model (180M)	512 (B64)	1024 (B32)	2048 (B16)	4096 (B8)	8192 (B4)	Mean ↑	Rel. range ↓
Transformer	**888.7**	**884.8**	**850.9**	**780.0**	683.7	**817.6**	25.10%
GDN	767.8	746.5	786.0	754.6	**770.0**	765.0	5.17%
Mamba-3 (SISO)	729.8	719.9	689.1	695.6	631.6	693.2	14.20%
SLiK	744.2	741.7	745.2	734.5	726.0	738.3	**2.60%**

2K training domain to a 4K evaluation domain. In contrast, accuracy on NIAH-Single-3 remained substantially lower at 16.6%.

K.2 Throughput Scaling with Sequence Length

We evaluated the computational efficiency of all models across sequence lengths $T \in 512, 1024, 2048, 4096, 8192$. To maintain a constant token budget, the batch size was decreased inversely with sequence length, using batch sizes of $64, 32, 16, 8, 4$, respectively. Thus, each configuration processed 32,768 tokens per batch. This protocol evaluates aggregate token-processing efficiency under a fixed token budget rather than latency at a fixed batch size.

This experiment is an architecture-level scaling benchmark, not an evaluation of a single model with a fixed 2K eigensystem. For each sequence length T, we instantiated the same approximately 187M-parameter SLiK architecture and evaluated its continuous coefficient functions on T uniformly spaced points over the normalized domain $[0, 1]$. We then discretized the SL operator on this length-specific grid and computed the corresponding eigensystem. The number of spectral modes and all learned parameter dimensions remained unchanged across sequence lengths. The eigensystem was constructed during the first forward pass and cached for subsequent evaluations.

All models were evaluated on one NVIDIA GH200 using BF16 activations and FP32 parameters.

As shown in Table 13, the Transformer achieves the highest throughput at short sequence lengths, reaching 888.7K tokens/s at $T = 512$ and an overall mean throughput of 817.6K tokens/s. However, its throughput decreases progressively with sequence length, falling to 683.7K tokens/s at $T = 8192$, with a relative range of 25.10%. Mamba-3 exhibits a similar dependence on sequence length, decreasing from 729.8K to 631.6K tokens/s and yielding a relative range of 14.20%. GDN is substantially more stable, with throughput between 746.5K and 786.0K tokens/s, and achieves the highest absolute throughput at $T = 8192$ (770.0K tokens/s).

In comparison, SLiK exhibits the most stable throughput across the evaluated sequence lengths. Its throughput remains between 726.0K and 745.2K tokens/s, with a mean of 738.3K tokens/s and the smallest relative range of only 2.60%. At $T = 8192$, SLiK reaches 726.0K tokens/s, approximately 6.2% higher than the Transformer. Although SLiK does not maximize absolute throughput at every operating point, its substantially lower throughput variation indicates that its computational efficiency is considerably less sensitive to increasing sequence length under a fixed token budget.

K.3 Context-Length Extrapolation under Frozen Parameters

We evaluated whether models pretrained with a 2K context could generalize to longer sequences without additional optimization. All models were pretrained on 100B FineWeb-Edu tokens using a maximum context length of 2,048. During evaluation, all learned parameters were frozen, and no gradient updates, fine-tuning, or long-context training were performed. We evaluated context lengths $T \in \{1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}\}$ on the same ordered held-out FineWeb-Edu token stream. For every context length, evaluation started from the same stream offset and scored the same total number of target tokens using an identical tokenizer, document-packing policy, and sequence-reset policy. Only the segmentation of the stream into context windows changed with T.

Length-specific SL discretization. Freezing SLiK does not imply reusing the discrete 2K eigensystem at every evaluation length. Instead, we freeze the learned continuous coefficient functions $p_\theta(x)$, $q_\theta(x)$, and $w_\theta(x)$ together with all remaining model parameters. For each target context length T, we define

$$x_i^{(T)} = \frac{i}{T-1}, \qquad i = 0, \ldots, T-1,$$

evaluate the frozen coefficient functions on this normalized grid, and re-discretize the corresponding continuous Sturm–Liouville operator using T nodes. We retain the same number of spectral modes used during pretraining and compute the corresponding eigensystem once before evaluation, after which it is cached for all sequences at that length.

This procedure is a deterministic, parameter-free re-discretization of the same frozen continuous operator. It introduces no new learned parameters and involves no gradient-based or data-dependent adaptation. Because $x_i^{(T)}$ depends on T, the positional basis assigned to a fixed absolute token prefix is not identical across evaluation lengths. The experiment therefore evaluates frozen-parameter generalization under length-specific discretization, rather than applying an unchanged 2K discrete eigensystem to a 32K sequence. Within each evaluation run, T is fixed before inference and the eigensystem is never resized within an autoregressive sequence.

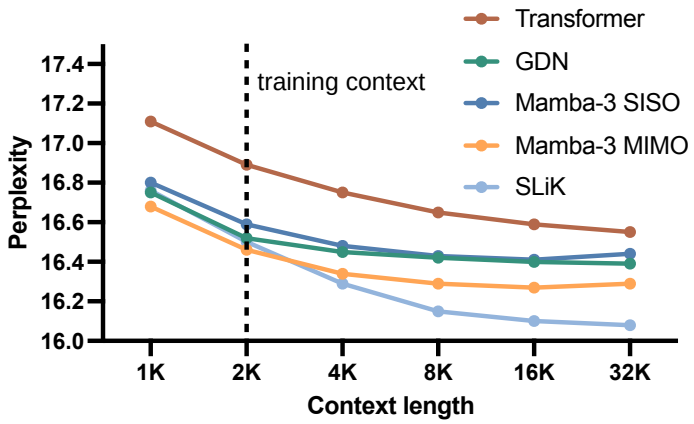

Figure 6: Pretrained 180M models' performance on the held-out FineWeb-Edu test set at varying context lengths. SLiK exhibits strong length extrapolation.

As shown in Figure 6, SLiK remained competitive at 1K and 2K and achieved the lowest perplexity among the evaluated models at every extrapolated context length from 4K to 32K. Its perplexity decreased from 16.76 at 1K to 16.08 at 32K. At 32K, SLiK achieved a perplexity 0.21 points lower than the best-performing baseline at that length, Mamba-3 MIMO, corresponding to a relative reduction of 1.29%. These results indicate that SLiK remains stable when evaluated far beyond its 2K training context while the same frozen continuous operator is re-discretized at the target resolution.

Longer evaluation windows provide additional left context and reduce the number of artificial segmentation boundaries in the token stream. Accordingly, the monotonic decrease in perplexity should not by itself be interpreted as evidence of improved long-range retrieval. Rather, this experiment evaluates context-length generalization of the frozen model under increasing sequence lengths.

L More detail of Long-read long RNA sequence model

To balance the number of transcript sequences across different splicing labels, the average count across all labels, 5,252, was set for selection. In total, the input sequence length for the dataset containing seven splicing events was set to 2,600 bp, comprising a total of 36,764 sequences. 20% of the sequences were utilised for test, and the remaining 80% of the sequences were applied for model training using random seed 42. The splicing types of transcript sequences are classified into seven categories: REF (reference from GENCODE transcript source), ASS (novel alternative splice

Table 14: Ablation study of SLiK on the seven-class long-read RNA splicing task. Params denotes millions of trainable parameters, and runtime is measured in minutes. Bold and underlined values indicate the best and second-best results, respectively, where applicable.

Model variant	Accuracy ↑	Macro-F1 ↑	Macro-AUROC ↑	Macro-AUPRC ↑	Params (M) ↓	Runtime (min) ↓
SLiK (full)	**0.9063**	**0.9022**	**0.9845**	**0.9386**	2.27	2.96
w/o SL operator	0.8232	0.8179	0.9544	0.8477	**1.48**	**0.53**
Fixed DCT	0.8504	0.8480	0.9592	0.8605	2.27	<u>0.89</u>
Frozen SL operator	<u>0.8652</u>	<u>0.8628</u>	<u>0.9654</u>	<u>0.8735</u>	2.27	1.12
Constant coefficients	0.8233	0.8241	0.9539	0.8388	2.27	1.24
Classical SL	0.8190	0.8218	0.9531	0.8403	2.27	1.01

site), ES (novel exon skipping), NE (novel exon), IR (novel intron retention), NC (new combination of existing exons), and MN (multiple novel events).

L.1 Ablation Study

All variants were evaluated using the same data split, optimization settings, and hardware. As shown in Table 14, the full SLiK model achieved the highest performance across all four predictive metrics. Removing the SL operator reduced Accuracy and Macro-F1 by 8.31 and 8.43 percentage points, respectively, while decreasing the parameter count by 0.79M. Freezing the SL operator produced the strongest ablated variant, but still reduced Accuracy and Macro-F1 by 4.11 and 3.94 percentage points. This result suggests that end-to-end optimization of the SL operator contributes substantially to model performance. The fixed-DCT variant reduced Accuracy and Macro-F1 by 5.59 and 5.42 percentage points, while the constant-coefficient and classical-SL variants produced larger accuracy reductions of 8.30 and 8.73 points. Macro-AUPRC decreased by 6.51–9.98 percentage points across all ablations. However, the full model required 2.96 minutes, compared with 0.53–1.24 minutes for the ablated variants, indicating a clear trade-off between predictive performance and computational cost.

L.2 Biological Relevance of Learned Representations in Long-Read Modelling

To understand how the SLiK architecture learns long-read transcript sequence representations for the prediction of specific splicing types, we performed an interpretability analysis by extracting random exon-exon splice junctions with a 60 nt sequence window. By probing the model's feature attribution, causal sensitivity, and internal spectral dynamics, we validate that the network learns biologically meaningful regulatory grammar rather than relying on spurious correlations.

We utilized Integrated Gradients (IG) to extract position-specific contribution scores relative to the splice junction (position 0). The resulting bar plots demonstrate that the model effectively localizes attention to functionally critical regions. When translated into Sequence Logos, these attribution scores reveal the specific nucleotide motifs driving the predictions. Bases with positive amplitudes represent learned Exonic Splicing Enhancer motifs, while negative amplitudes denote Exonic Splicing Silencers. The model successfully captures the context-dependent nature of RNA-binding protein (RBP) motifs, identifying diverse sequence patterns across different transcripts rather than memorizing a single global consensus (Figure 7 and Figure 8).

To map the causal landscape of the sequence space, we performed In Silico Mutagenesis (ISM) by systematically mutating every base of the splice junction to all possible alternatives (A, C, G, T, N). The resulting heatmaps illustrate the shift in target logits. The model exhibits a highly non-linear response to point mutations, which mimics the biological mutational constraints (Figure 9). We observed that some regulatory loci show extreme fragility, with deep blue or red indicating severely disrupted splicing type prediction. Conversely, adjacent regions often demonstrate robust neutrality, aligning with the biological presence of non-conserved linker regions within mature transcripts.

The SLiK Spectral Modes plots validate the theoretical foundations of the continuous operator. The monotonically increasing eigenvalues (λ_k) confirm that the model learns a well-posed, stable Sturm-Liouville basis (Figure 10). By transforming discrete nucleotide sequences into a continuous spectral domain, the model dynamically distributes mode activations across frequencies. Low-frequency

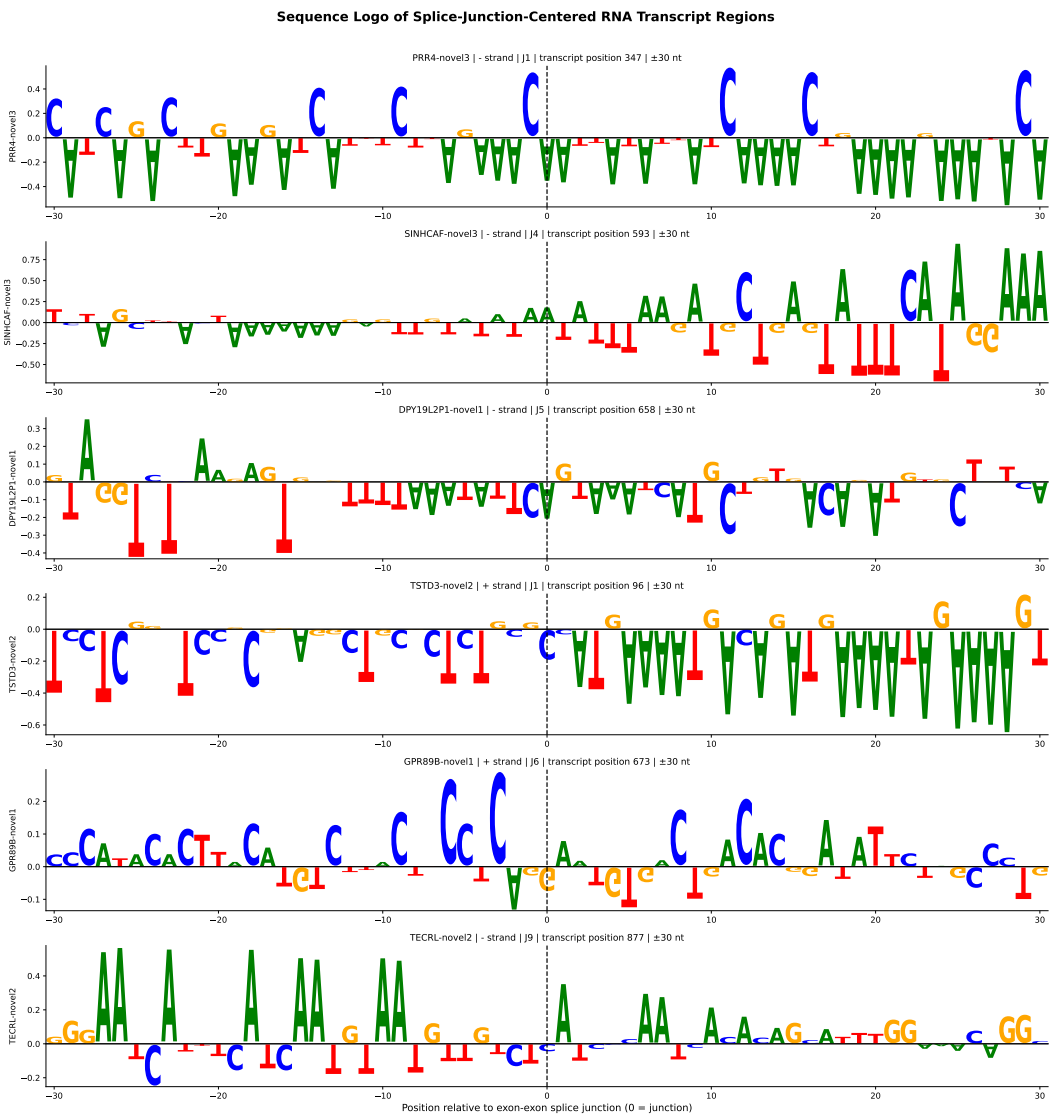

Figure 7: Nucleotide-level attribution represented as a sequence logo, where nucleotide heights indicate their Integrated Gradients contributions to the target prediction.

modes capture global contextual features (such as regional GC content or structural stability), while higher-frequency modes fine-tune the high-resolution, local motif interactions near the splice junction.

M Hyperparameter Tuning and Analysis

M.1 Common Experimental Settings

All models were implemented in PyTorch and evaluated under identical data splits, preprocessing procedures, and hardware conditions within each benchmark. The RNA and time-series models were trained on four NVIDIA GH200 GPUs using AdamW with a learning rate of 10^{-4} and weight decay of 10^{-4}. We applied cosine annealing with a 10% linear warm-up, a global batch size of 64, and gradient clipping at a global norm of 1.0. Training was performed for up to 200 epochs, and the checkpoint with the best validation performance was used for testing. We fixed the random seed

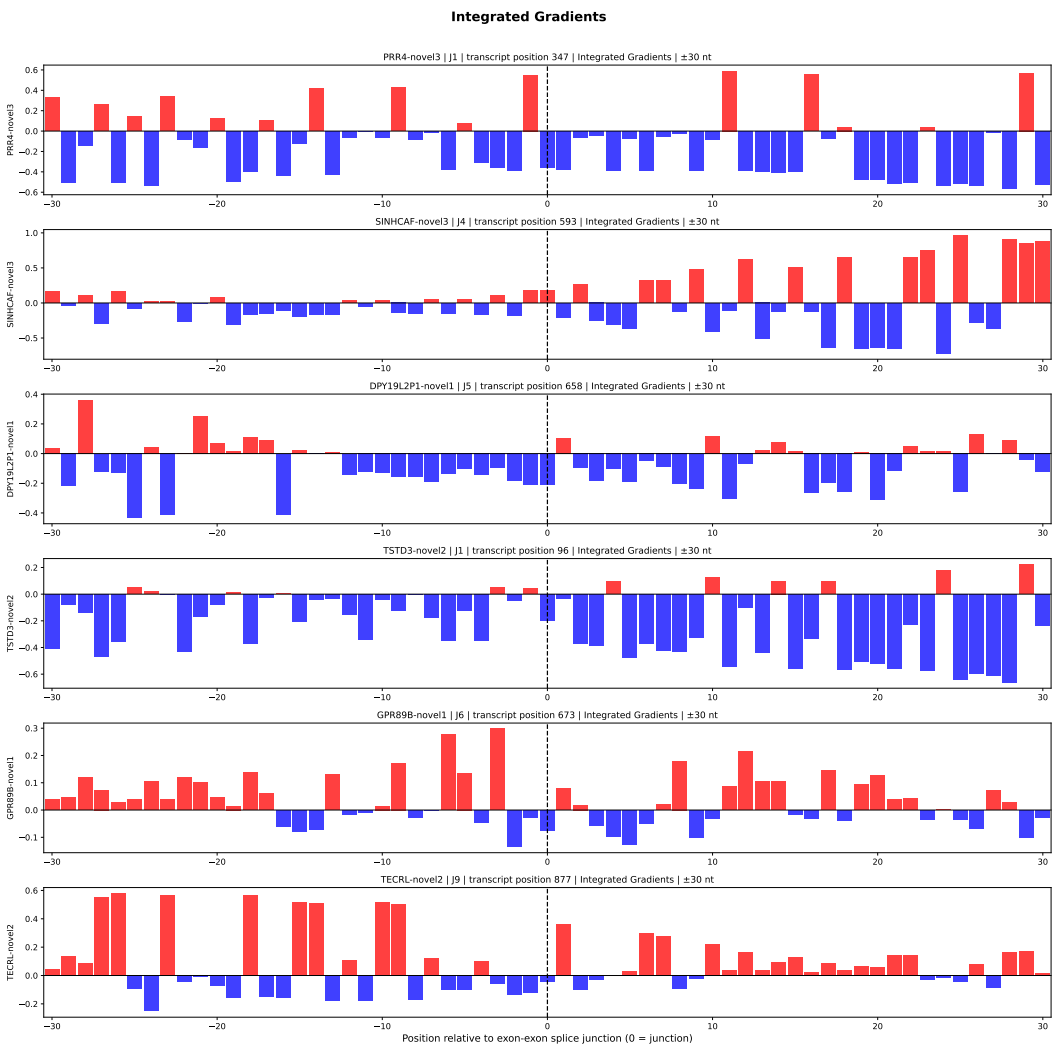

Figure 8: Positional Integrated Gradients profile showing the overall contribution of each transcript position

to 42. Model-specific architectures, patch configurations, objectives, and early-stopping criteria are reported in the corresponding benchmark subsections.

For language modeling, all models were trained for 100B tokens with a context length of 2,048 and a global batch size of 524,288 tokens. We used AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$), weight decay of 0.1, and global gradient clipping at 1.0. The learning rate was linearly warmed up to 6×10^{-4} over 375M tokens and then cosine-decayed to 10^{-5}. Training used bfloat16 mixed precision on four NVIDIA GH200 GPUs with seed 42.

M.2 Task-Specific SLiK Architecture Configurations

SLiK uses the same learnable Sturm–Liouville operator and local–spectral design across all task families. We divide its hyperparameters into shared operator settings and task-specific capacity settings. The shared settings control how the SL basis is constructed and filtered, whereas the task-specific settings determine the width, depth, and spectral rank of each model. Table 15 summarizes the configurations used in the RNA, time-series, and language-modeling experiments.

Shared SL operator settings. We keep the principal SL operator settings unchanged across the three task families. All models use local convolution widths $\{3, 5, 7\}$, four spectral responses, and

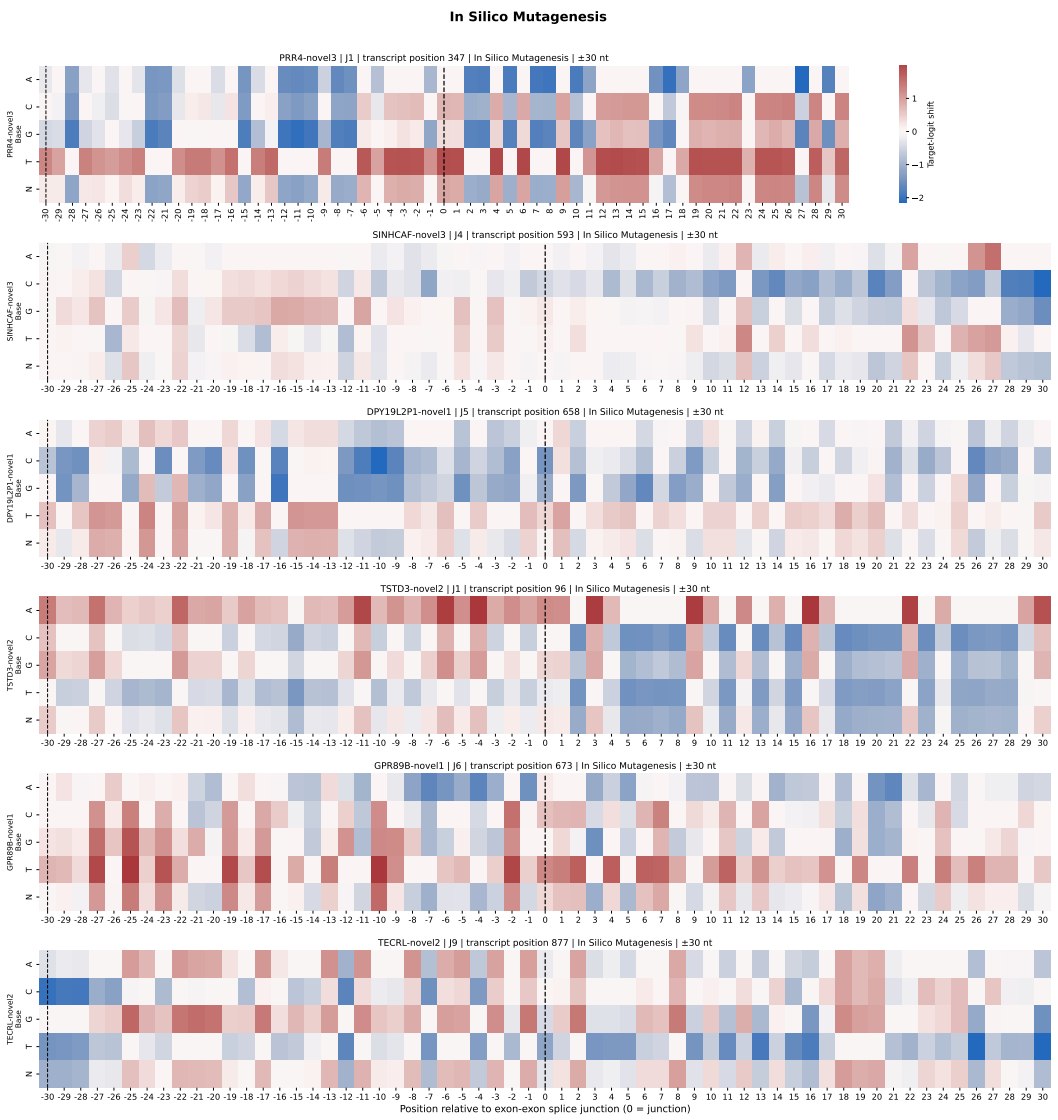

Figure 9: In silico mutagenesis heatmap showing changes in the target-class logit induced by single-nucleotide substitutions at each transcript position.

a response-gate hidden width of 32. The coefficient network also has a hidden width of 32 and uses coordinate frequencies $\{1, 2, 4, 8\}$. This common parameterization ensures that each experiment uses the same class of continuous variable-coefficient SL operators rather than a task-specific basis construction.

We use eight subspace iterations and an oversampling width of 16 when approximating the low-frequency eigenspace. The eigensolver performs its reduced numerical operations in the accurate-solve precision specified in Table 15. These settings provide a consistent numerical approximation of the retained low-frequency modes across different sequence lengths. All residual branches use a LayerScale initialization of 10^{-3}, which places each block close to the identity map at initialization while preserving gradients through the local, spectral, and feed-forward branches.

RNA and time-series configurations. The RNA model uses hidden dimension $D = 128$, eight SLiK blocks, and 32 retained spectral modes. This configuration assigns greater depth and representation capacity to the RNA experiments while retaining a compact spectral representation. The time-series model uses a smaller configuration with $D = 64$, two blocks, and 16 spectral modes.

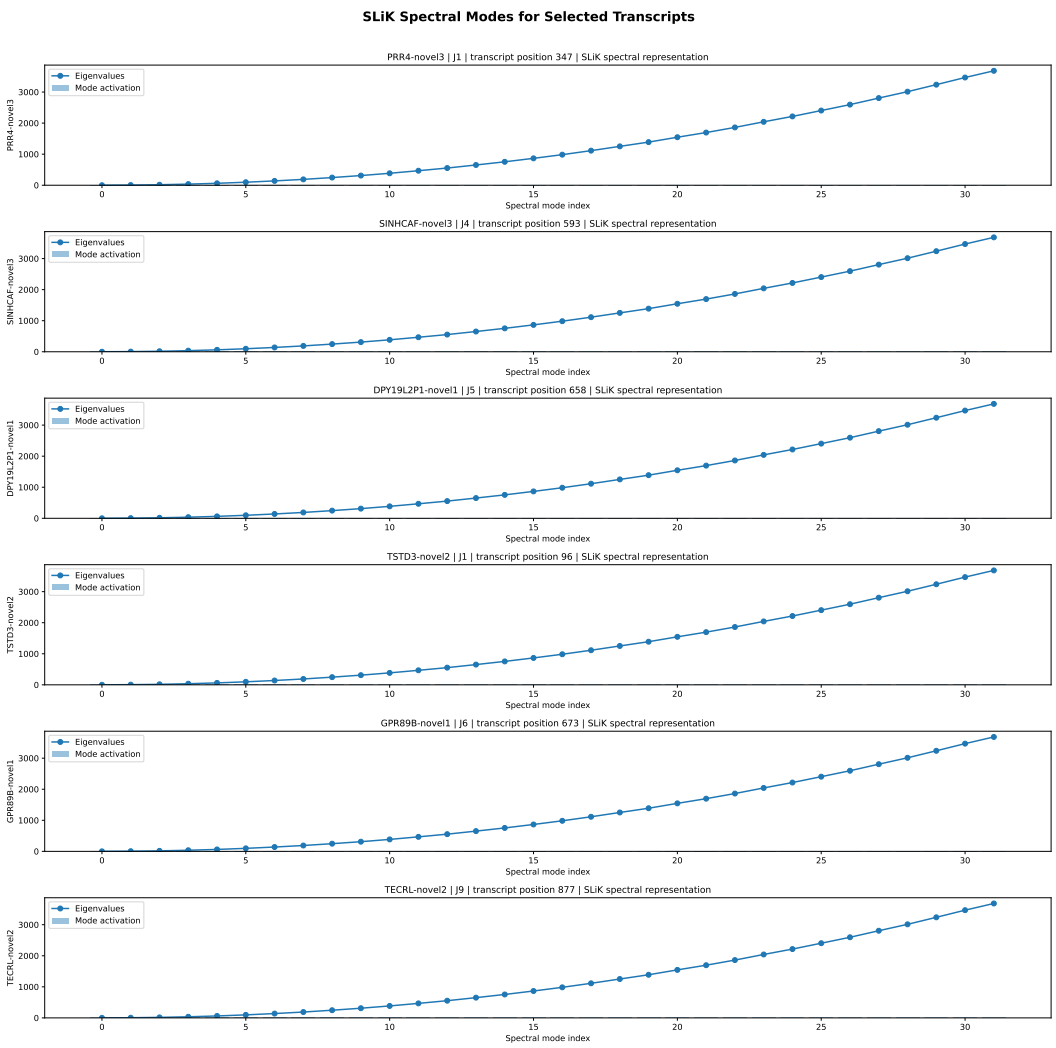

Figure 10: Spectral analysis of the SLiK representation, showing the activation magnitude of individual Sturm-Liouville modes together with their corresponding eigenvalues.

This setting provides a lower-cost backbone for the patch-based forecasting experiments, whose effective sequence length is the number of input patches rather than the number of raw time points. Both configurations use a two-layer GELU feed-forward network with hidden width $4D$, LayerNorm, and dropout probability 0.1.

For both task families, the complete input window is available before the prediction is produced. Their local convolutions therefore use symmetric same-length padding, and their spectral coefficients are computed from the full observed input window. This full-window processing does not expose the model to forecasting targets because only the historical input window enters the SLiK encoder.

Language-modeling configuration. The language-modeling model is designed for the parameter-matched 180M-scale comparison. It uses hidden dimension $D = 768$, 12 causal SLiK blocks, 32 spectral modes, and four spectral responses. The model retains the same coefficient-network and eigensolver settings used in the other task families. Its larger width and depth follow the capacity regime of the compared 180M-parameter language models.

The language-modeling scaffold uses pre-RMSNorm blocks and a SwiGLU feed-forward network. We set its hidden dimension to $D_{\text{ff}} = 896$ so that the complete causal SLiK model, including

Table 15: Architecture hyperparameters for the task-specific SLiK instantiations used in the RNA, time-series, and language-modeling experiments.

Hyperparameter	RNA	Time series	Language modeling
Hidden dimension D	128	64	768
Number of SLiK blocks L	8	2	12
Number of spectral modes K	32	16	32
Local convolution kernels	$\{3, 5, 7\}$	$\{3, 5, 7\}$	$\{3, 5, 7\}$
Number of spectral responses R	4	4	4
Response-gate width H_g	32	32	32
FFN type	GELU MLP	GELU MLP	SwiGLU
FFN hidden dimension D_{ff}	512	256	896
Normalization	LayerNorm	LayerNorm	RMSNorm
Dropout	0.1	0.1	0.0
LayerScale initialization	10^{-3}	10^{-3}	10^{-3}
Coefficient-network width H_c	32	32	32
Coordinate frequencies	$\{1, 2, 4, 8\}$	$\{1, 2, 4, 8\}$	$\{1, 2, 4, 8\}$
Eigensolver iterations	8	8	8
Eigensolver oversampling S	16	16	16
Accurate eigensolve	yes	yes	yes
SL basis domain	current length T	current length T	fixed length 2,048
Sequence mixing	full observed window	full observed window	causal prefix

the full-rank spectral response maps and tied token embeddings, contains 187,209,206 trainable parameters. The model uses the 128,256-token Llama-3.1 vocabulary and ties the input and output embedding matrices.

Unlike the RNA and time-series encoders, the language model must prevent future-token access at every position. We therefore use causal left-padded convolutions and prefix-accumulated spectral memory. For a configured maximum context length $T_{\max}$, the SL eigensystem is constructed once on the normalized domain discretized with $T_{\max}$ positions. Shorter forward passes use the corresponding prefix of this fixed basis, ensuring that an identical token prefix is represented by the same spectral basis coordinates during full-sequence training and autoregressive decoding.

For standard language-model pretraining and evaluation, we set $T_{\max} = 2{,}048$. In the throughput-scaling experiments, $T_{\max}$ is instead set to the target sequence length before evaluation. The same continuous SL coefficient parameterization is re-discretized on the corresponding $T_{\max}$-point grid, after which the resulting eigensystem is computed and cached. All learned model parameters and the number of retained spectral modes remain unchanged across sequence lengths. The basis is never resized within an autoregressive sequence. The measurements beyond the 2,048-token training context are used only to characterize computational scaling and are not interpreted as evidence of predictive length extrapolation.

We set language-model dropout to zero, following the common pretraining protocol used in the 100-billion-token comparison. The RNA and time-series models retain dropout 0.1 because they are trained on substantially smaller supervised datasets. These task-dependent choices modify model capacity, regularization, and causal information flow, but do not change the underlying learnable SL operator.

N Preliminary Exploration of SLiK for Image Classification

To examine whether SLiK can extend beyond one-dimensional sequential data, we conducted a preliminary image-classification experiment on CIFAR-10 (Krizhevsky, 2009). Each 32×32 image was divided into non-overlapping 4×4 patches and projected into visual tokens using a convolutional patch embedding. Fixed two-dimensional sinusoidal positional embeddings were added to preserve spatial information. The resulting tokens were arranged in a snake-like scanning order and processed by the original bidirectional SLiK backbone, followed by mean pooling and a linear classification head. Importantly, the internal SLiK blocks were not modified specifically for vision.

Using the unified CIFAR-10 training protocol adopted by NesT, the 6.11-million-parameter SLiK model achieved 89.57% top-1 and 99.22% top-5 test accuracy. Its top-1 accuracy exceeds that of DeiT-T (88.39%) (Touvron et al., 2021) and is close to that of PVT-T (90.51%) (Wang et al., 2021). These results are obtained using a simple patch-to-sequence adaptation rather than a vision-specific architecture, providing preliminary evidence that SLiK can effectively process visual inputs and has potential as a general-purpose backbone beyond conventional sequence tasks.

We emphasize that this experiment is intended only as a proof of concept. The current model does not explicitly exploit the two-dimensional and multiscale structure of images. Future work will develop a vision-specific SLiK architecture with native two-dimensional spectral operators, hierarchical representations, and multiscale spatial interactions, and will evaluate it systematically on larger-scale vision benchmarks.

O Limitations and Future Work

Our language-modeling experiments were constrained by the available computational budget and wall-clock time. The current study therefore focuses on models with approximately 180M parameters, trained on 100B tokens with a context length of 2,048. Although this setting permits controlled comparisons with strong baselines, it does not establish how SLiK scales to billion-parameter models, larger training corpora, or substantially longer contexts. Evaluating these scaling regimes under matched data and computational budgets is an important direction for future work.

The current Sturm–Liouville operator is learned across the training set and shared across samples. It therefore captures task-level spectral geometry but is not conditioned on the content of each individual sequence. Input-conditioned operators may provide greater flexibility for heterogeneous data, although they would introduce additional computational and numerical challenges. Moreover, SLiK retains a fixed number of low-order modes. This truncation controls computational cost but may discard information that cannot be represented adequately in the retained subspace. Increasing or dynamically selecting the number of modes could improve expressivity, but would alter the associated time and memory costs.

P LLM Usage Statement.

In this work, large language models (LLMs) are used solely as general-purpose assistive tools for grammar correction and language polishing. All technical content, research ideas, experiments, and analyses were conceived, designed, and executed solely by the authors. The LLM did not contribute to the scientific content, experimental design, or interpretation of the results.

Q Reproducibility Statement

We have taken deliberate steps to ensure the reproducibility of our work. Detailed descriptions of the model architecture, training process, and hyperparameter settings are presented in the main text and supplementary materials. An anonymous implementation of the core model is also made available in the supplementary code.