Aligning Inductive Bias for Data-Efficient Generalization in State Space Models

Qiyu Chen[1,2] **Guozhang Chen**[1,3*]
[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science
[2]School of Physics
[3]National Engineering Research Center of Visual Technology, School of Computer Science
Peking University, Beijing 100871, China
cqy@stu.pku.edu.cn guozhang.chen@pku.edu.cn

Abstract

The remarkable success of modern AI has been closely tied to scaling laws, yet the finite supply of high-quality data makes data efficiency—learning more from less—an increasingly important frontier. A model's inductive bias is a critical lever for data efficiency, but foundational sequence models such as State Space Models (SSMs) often rely on fixed, task-agnostic biases. When this fixed prior is misaligned with the underlying structure of a task, the model may require additional samples to overcome its own bias before learning the relevant signal. In this work, we introduce a principled framework for understanding and aligning the inductive bias of linear time-invariant SSMs. We first formalize this bias through an SSM-induced kernel and show theoretically and empirically that its spectrum is governed by the model's frequency response. This characterization motivates Task-Dependent Initialization (TDI), a fast power-spectrum matching method that aligns the initial SSM bias with the task's spectral characteristics before downstream training. Across controlled synthetic experiments, trainable one-layer SSMs, and deep SSMs on diverse real-world benchmarks, TDI can improve learning under fixed training budgets primarily when labeled data are limited, label-relevant spectral structure is estimable, and the default SSM initialization is spectrally mismatched. These benefits include better early learning, but do not necessarily imply higher accuracy after extended training. Our results provide both a theoretical lens and a practical tool for task-adaptive inductive bias, suggesting a path toward more data-efficient sequence modeling. Code is available at `https://github.com/Criticality-Cognitive-Computation-Lab/Aligning-Inductive-Bias-for-State-Space-Models`.

1 Introduction

The paradigm of scaling laws, wherein model performance predictably improves with more data, parameters, and compute, has driven unprecedented progress in AI [31, 28, 5]. However, high-quality training data is finite and often difficult to obtain in scientific, medical, industrial, and long-tail domains [41, 46, 14]. This makes data efficiency—learning more from fewer samples—an increasingly important goal [41, 1]. A central lever for data efficiency is *inductive bias*: the built-in assumptions that make some functions easier to learn than others [27, 6, 19]. When this bias is aligned with the task, it can reduce sample complexity; when it is misaligned, the model may require additional data to compensate for an unfavorable prior.

State Space Models (SSMs) have emerged as a powerful and efficient architecture for sequence modeling [23, 24, 43, 21, 16, 37]. Many widely used SSM layers, including S4 and S4D [23, 24],

*Corresponding author: guozhang.chen@pku.edu.cn

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

are built from linear time-invariant (LTI) filtering operators, which transform an input sequence through a structured convolution before the readout. Although recent selective models such as Mamba [21, 16] introduce input-dependent dynamics, the LTI case remains a fundamental and analytically tractable setting for understanding how SSM filtering induces inductive bias. This filtering operator is not neutral: it emphasizes some temporal and frequency modes more than others. Existing LTI-based initializations are typically task-agnostic and therefore impose a fixed spectral prior before downstream training [24, 49]. As illustrated in Figure 1, this prior can be beneficial when it overlaps with task-relevant frequencies, but can become a sample-efficiency bottleneck when the task-relevant components lie in low-response regions of the model.

In this work, we make this spectral inductive bias explicit and use it to design a task-adaptive initialization method. We view an LTI SSM as a convolutional feature map, define the corresponding SSM-induced kernel, and show that its spectrum is governed by the SSM frequency response. This characterization motivates *Task-Dependent Initialization (TDI)*, which estimates a task-relevant spectral statistic and aligns the initial SSM spectrum with it before downstream training. By placing more model response on task-ranked frequency modes, TDI increases the kernel mass assigned to components predicted to be task-relevant. TDI changes only the initialization, leaving the architecture, supervised objective, and inference cost unchanged. Unlike Yu et al. [49] who tune SSM frequency bias empirically, and Liu and Li [34] who optimize autocorrelation-based initialization, our method aligns the model to task-dependent spectrum, concentrating model's power on task-relevant frequencies (see Appendix A: Related Work for full comparison). TDI targets data-limited settings where label-relevant spectral structure can be estimated from the available training data and the default initialization is spectrally mismatched. Our subsampling experiments simulate limited data availability; they do not recommend discarding available training data.

Our contributions are as follows.

- **A kernel-theoretic characterization of SSM inductive bias.** We formalize LTI SSMs as convolutional feature maps and define an SSM-induced kernel whose spectrum reveals which modes are favored in the small-sample regime, and show that this spectrum is governed by the SSM frequency response.
- **Task-Dependent Initialization via spectral alignment.** Based on this characterization, we propose Task-Dependent Initialization (TDI), a lightweight initialization method that aligns the initial SSM spectrum with task-relevant spectral components without changing the downstream architecture, training objective, or inference cost.
- **Validation from controlled theory to practical models.** We validate the mechanism in kernel regression, trainable one-layer SSMs, and deep SSMs, showing that gains concentrate in finite-sample regimes where task-relevant spectral structure is present and the default SSM bias is mismatched with the task.

2 Preliminaries

2.1 Problem Statement

In finite-sample regimes, generalization depends not only on model capacity, but also on whether the model's inductive bias makes the task easy to learn. The generalization error, E_g, quantifies this ability by averaging the model's performance over all possible training datasets $\mathcal{D}$ of a given size P.

Models rely on an inductive bias—a set of built-in assumptions—to generalize effectively. For SSMs, however, the relationship between their frequency response, induced inductive bias, and data-efficient generalization remains insufficiently characterized. The goal of this work is not merely to analyze the expressive power of SSMs, but to characterize which target components are favored by an SSM in the small-sample regime. To this end, we study the kernel induced by the SSM convolution operator, because kernel spectra provide a direct description of mode-wise learnability.

2.2 State Space Models as Toeplitz Convolution Operators

In general, a continuous-time linear time-invariant (LTI) state space model (SSM) is defined as

$$\dot{\mathbf{x}}(t) = \boldsymbol{A}\mathbf{x}(t) + \boldsymbol{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \boldsymbol{C}\mathbf{x}(t) + \boldsymbol{D}\mathbf{u}(t), \tag{1}$$

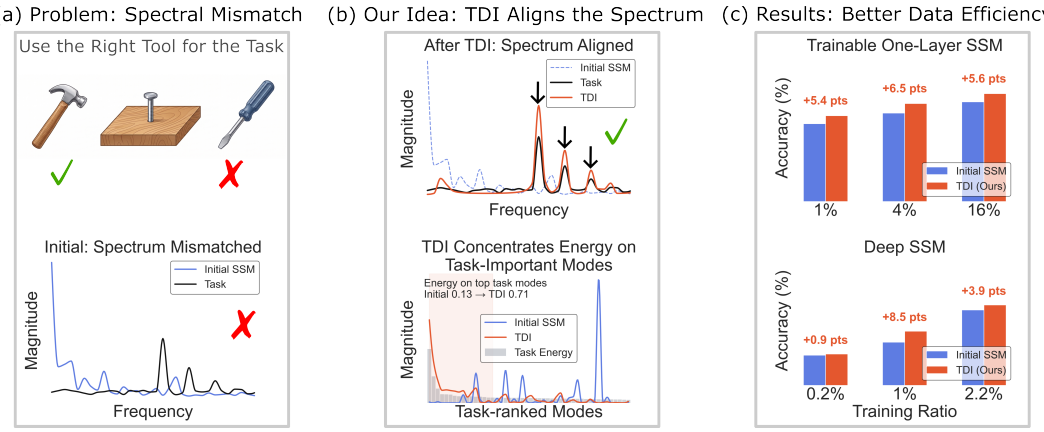


Figure 1: **Graphic overview.** (a) A fixed, task-agnostic SSM initialization can place spectral energy in frequency regions that do not match the task-relevant spectrum, analogous to using an unsuitable tool for a job. (b) TDI initializes the SSM so that its frequency response aligns with task-relevant spectral components and assigns more energy to task-important modes. (c) Across trainable one-layer SSMs and deep SSMs, TDI achieves higher mean final-epoch test accuracy under fixed training budgets. A training ratio r denotes training on an r-fraction of the original training split, while keeping the validation and test splits fixed.

where $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the hidden state, $\mathbf{u}(t) \in \mathbb{R}^m$ is the input, and $\mathbf{y}(t) \in \mathbb{R}^p$ is the output. We discretize the continuous SSM using the Zero-Order Hold (ZOH) method with step size Δ, yielding the discrete SSM:

$$\mathbf{x}_k = \overline{\boldsymbol{A}}\mathbf{x}_{k-1} + \overline{\boldsymbol{B}}u_k, \quad y_k = \overline{\boldsymbol{C}}\mathbf{x}_k, \quad k = 0, 1, \cdots L-1 \quad (L \text{ is input length}), \tag{2}$$

where $\overline{\boldsymbol{A}} = \exp(\Delta\boldsymbol{A}), \overline{\boldsymbol{B}} = (\Delta\boldsymbol{A})^{-1}(\exp(\Delta\boldsymbol{A}) - \boldsymbol{I})\Delta\boldsymbol{B}, \overline{\boldsymbol{C}} = \boldsymbol{C}$. For simplicity, and in line with modern deep SSMs such as S4 [23] and S4D [24], we omit the direct feedthrough term $\boldsymbol{D}$ and treat u_k and y_k as scalars. For an initial state $\mathbf{x}_{-1} = \mathbf{0}$, the output sequence $\boldsymbol{y} \equiv [y_0, \cdots, y_{L-1}]^\top$ is the convolution of the input sequence $\boldsymbol{u} \equiv [u_0, \cdots, u_{L-1}]^\top$ with the SSM's impulse response kernel h, given by $y_k = \sum_{n=0}^{k} h_n u_{k-n}$, where the kernel is defined as $h_n = \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^n\overline{\boldsymbol{B}}$. The frequency response $H(\omega)$ of an SSM is the discrete-time Fourier transform of its impulse response h, $H(\omega) = \mathrm{DTFT}(h) = \sum_{n=0}^{\infty} h_n e^{-i\omega n}$, $\omega \in (0, 2\pi)$. This convolution process can be represented by a Toeplitz matrix $\boldsymbol{T}_h$ that maps the input sequence $\boldsymbol{u}$ to the output sequence $\boldsymbol{y}$ as $\boldsymbol{y} = \boldsymbol{T}_h\boldsymbol{u}$. Specifically, $\boldsymbol{T}_h \in \mathbb{R}^{L\times L}$ is the lower-triangular Toeplitz matrix defined entrywise by $(\boldsymbol{T}_h)_{k\ell} = h_{k-\ell}$ for $k \geq \ell$ and $(\boldsymbol{T}_h)_{k\ell} = 0$ otherwise, with $k, \ell = 0, \ldots, L-1$. Since $\boldsymbol{T}_h$ is the operator through which an SSM filters the input before the readout, we treat $\boldsymbol{T}_h\boldsymbol{u}$ as the SSM feature representation of $\boldsymbol{u}$.

2.3 A Kernel-Theoretic Framework for Analyzing Inductive Bias

To analyze the inductive bias of SSMs, we employ the framework of kernel methods. *Kernel Ridge Regression (KRR)* provides a powerful lens for this by finding a function f in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_K$ that minimizes the regularized empirical risk: $f^*_{\mathcal{D}} = \arg\min_{f\in\mathcal{H}_K} \frac{1}{P}\sum_{\mu=1}^{P} \|f(\boldsymbol{u}^\mu) - \boldsymbol{Y}^\mu\|_2^2 + \lambda\|f\|^2_{\mathcal{H}_K}$, where $\lambda > 0$ is a regularization parameter, P is the number of training samples, and $\mathcal{D} = \{\boldsymbol{u}^\mu, \boldsymbol{Y}^\mu\}_{\mu=1}^P$ denotes the training set. The inputs $\boldsymbol{u}^\mu$ are sampled i.i.d. from a distribution $p(\boldsymbol{u})$, and labels $\boldsymbol{Y}^\mu \in \mathbb{R}^d$ (d is the dimension of labels) are generated by $\boldsymbol{Y}^\mu = \overline{f}(\boldsymbol{u}^\mu) + \epsilon^\mu$, where ϵ^μ are zero-mean noise with covariance $\langle\epsilon^\mu\epsilon^\nu\rangle = \sigma^2\delta_{\mu\nu}$. The RKHS is uniquely defined by a kernel function $K(\boldsymbol{u}, \boldsymbol{u}')$ that satisfies the reproducing property [4, 40].

The spectral properties of the kernel, revealed by Mercer's Theorem, are crucial for understanding generalization. Mercer's Theorem [39] states that, a continuous, symmetric, and positive semi-definite kernel $K(\boldsymbol{u}, \boldsymbol{u}')$ can be decomposed as a series expansion in terms of its eigenfunctions $\{\phi_\rho\}_{\rho=1}^\infty$ and non-negative eigenvalues $\{\eta_\rho\}_{\rho=1}^\infty$: $K(\boldsymbol{u}, \boldsymbol{u}') = \sum_{\rho=1}^{\infty} \eta_\rho\phi_\rho(\boldsymbol{u})\phi_\rho(\boldsymbol{u}')$, where the eigenfunctions are normalized with respect to a given distribution $p(\boldsymbol{u})$.

The inductive bias of the kernel is encoded in its spectrum. As shown by Canatar et al. [11], Bordelon and Pehlevan [10], the generalization error can be decomposed into a sum of mode-wise errors, revealing a *spectral bias*: modes ρ corresponding to larger eigenvalues η_ρ are learned more rapidly as the number of data samples increases. Therefore, if the target function has most of its power on large-eigenvalue modes, it is well *aligned* with the model bias and can be learned with fewer samples. *Cumulative power* $C(\rho)$ is introduced to quantify the alignment between task and model. After sorting the modes ρ by decreasing eigenvalues $\eta_1 \geq \eta_2 \geq \cdots$ and expanding the target function $\overline{f}(\boldsymbol{u}) = \sum_\rho \overline{w}_\rho \phi_\rho(\boldsymbol{u})$ in the kernel's eigenbasis, $C(\rho) = \frac{\sum_{l=1}^{\rho} \overline{w}_l^2}{\sum_{l=1}^{\infty} \overline{w}_l^2}$. Target functions whose power is concentrated in the top kernel modes (i.e., those with a fast-rising $C(\rho)$) can be learned with fewer samples. For a d-dimensional target function, we set $\overline{w}_\rho^2 = \sum_{j=1}^{d} \overline{w}_{\rho,j}^2$, where each $\overline{w}_{\rho,j}$ is defined by the expansion $\overline{f}_j(\boldsymbol{u}) = \sum_\rho \overline{w}_{\rho,j} \phi_\rho(\boldsymbol{u})$.

3 Theory and Method

3.1 SSM-Induced Kernel

In Section 2.2, we showed that a discrete-time LTI SSM maps an input sequence $\boldsymbol{u} \in \mathbb{R}^L$ to the filtered sequence $\boldsymbol{T}_h \boldsymbol{u}$, where $\boldsymbol{T}_h$ is the Toeplitz convolution matrix generated by the SSM impulse response h. We treat the SSM convolution as a feature map, $\Phi_{\mathrm{SSM}}(\boldsymbol{u}) = \frac{1}{\sqrt{L}} \boldsymbol{T}_h \boldsymbol{u}$. The inner product between these SSM features defines a kernel over input sequences.

Definition 1 (SSM-Induced Kernel)**.** *Let $\boldsymbol{u}, \boldsymbol{u}' \in \mathbb{R}^L$ be two input sequences of length L. The SSM-induced kernel is defined as the inner product of their corresponding Toeplitz matrix features: $K_{\mathrm{SSM}}(\boldsymbol{u}, \boldsymbol{u}') = \frac{1}{L} (\boldsymbol{T}_h \boldsymbol{u})^\top (\boldsymbol{T}_h \boldsymbol{u}')$, where $\boldsymbol{T}_h$ denotes the Toeplitz matrix defined by the SSM's impulse response kernel.*

Since K_{SSM} is an inner product in the feature space Φ_{SSM}, it is positive semi-definite (see Appendix B.1: Properties of the SSM-Induced Kernel). The structure of the kernel is determined by the SSM parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, and Δ, which govern the temporal dynamics of the system. Unlike the standard linear kernel on raw inputs, this kernel first filters the sequence through the SSM convolution operator. Thus, its geometry reflects the temporal filtering bias encoded by the Toeplitz operator.

3.2 Characterizing the SSM-Induced Kernel

We characterize the SSM-induced kernel directly through the singular value decomposition of its finite-dimensional feature operator. Write $\boldsymbol{T}_h = \sqrt{L} \boldsymbol{U} \boldsymbol{S} \boldsymbol{V}^\top$, where $\boldsymbol{S} = \mathrm{diag}(s_1, \ldots, s_L)$ contains the scaled singular values and $\boldsymbol{v}_\rho$ denotes the ρ-th column of $\boldsymbol{V}$. Then $\frac{1}{L} \boldsymbol{T}_h^\top \boldsymbol{T}_h = \boldsymbol{V} \boldsymbol{S}^2 \boldsymbol{V}^\top$. Thus, the kernel decomposition follows from the ordinary spectral decomposition of this positive semi-definite matrix. To identify these eigenvectors with orthonormal kernel eigenfunctions, we first assume whitened inputs, isolating the contribution of the SSM operator. For centered inputs with covariance $\boldsymbol{\Sigma} \succ 0$, writing $\boldsymbol{u} = \boldsymbol{\Sigma}^{1/2} \boldsymbol{x}$ gives whitened coordinates $\boldsymbol{x}$ and the effective feature operator $\boldsymbol{T}_h \boldsymbol{\Sigma}^{1/2}$. Its spectrum therefore reflects both the SSM and the input covariance, and this effective operator need not retain the Toeplitz structure (see Appendix B.2: Frequency-Domain Characterization of the SSM Spectrum).

Theorem 3.1 (Eigen-Decomposition of SSM-Induced Kernel)**.** *Assume $\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}\boldsymbol{u}^\top] = \boldsymbol{I}_L$. Then the SSM-induced kernel admits the finite expansion $K_{\mathrm{SSM}}(\boldsymbol{u}, \boldsymbol{u}') = \boldsymbol{u}^\top \boldsymbol{V} \boldsymbol{S}^2 \boldsymbol{V}^\top \boldsymbol{u}' = \sum_{\rho=1}^{L} \eta_\rho \phi_\rho(\boldsymbol{u}) \phi_\rho(\boldsymbol{u}')$, where $\eta_\rho = s_\rho^2$ and $\phi_\rho(\boldsymbol{u}) = \boldsymbol{v}_\rho^\top \boldsymbol{u}$. These eigenfunctions satisfy $\mathbb{E}_{\boldsymbol{u}}[\phi_\rho(\boldsymbol{u}) \phi_\sigma(\boldsymbol{u})] = \delta_{\rho\sigma}$. Consequently, the kernel has rank at most L.*

This result shows that the SSM-induced kernel favors the right singular directions of the SSM convolution operator. As discussed in Section 2.3, modes associated with larger s_ρ become high-eigenvalue kernel modes and are therefore learned more rapidly in the small-sample regime. Thus, the inductive bias of an LTI SSM is encoded in the singular spectrum of its Toeplitz matrix. The next subsection connects this singular spectrum to the SSM frequency response.

3.3 Frequency Response Reveals the Spectral Bias of SSMs

Theorem 3.1 shows that the eigenvalues of the SSM-induced kernel are determined by the singular values of the Toeplitz convolution matrix $\boldsymbol{T}_h$. We now connect these singular values to the SSM frequency response. This step turns the algebraic characterization above into a frequency-domain interpretation of the model's inductive bias.

Recall that the impulse response $h = \{h_n\}_{n\geq 0}$ of an LTI SSM has frequency response $H(\omega) = \sum_{n=0}^{\infty} h_n e^{-i\omega n}, \omega \in (0, 2\pi)$. Intuitively, $|H(\omega)|$ measures how strongly the SSM amplifies an input component at frequency ω. Therefore, if the singular values of $\boldsymbol{T}_h$ are governed by $|H(\omega)|$, then the kernel eigenvalues in Theorem 3.1 encode the frequency preference of the SSM.

For a finite sequence length L, let $\omega_j = 2\pi j/L, j = 0, \cdots, L-1$, denote the discrete Fourier frequencies. We use $\tilde{s}_j$ to denote the frequency-indexed singular values associated with ω_j. Unlike the sorted singular values s_ρ in Theorem 3.1, the quantities $\tilde{s}_j$ are indexed by frequency and are not sorted.

Theorem 3.2 (Frequency response governs the SSM-induced spectrum)**.** *For an SSM with absolutely summable impulse response h, i.e., $\sum_{n=0}^{\infty} |h_n| < \infty$, the singular values of the Toeplitz convolution operator $\boldsymbol{T}_h$ are asymptotically governed by the SSM frequency response. In particular, as $L \to \infty$, $\tilde{s}_j \sim |H(\omega_j)|/\sqrt{L}, \omega_j = \frac{2\pi j}{L}$. Equivalently, the eigenvalues of the SSM-induced kernel are governed by $\tilde{\eta}_j \sim |H(\omega_j)|^2$ up to a scale factor determined by the kernel feature map normalization.*

The proof is provided in Appendix B.2: Frequency-Domain Characterization of the SSM Spectrum. The whitened-coordinate analysis above isolates the bias contributed by the SSM operator itself. We also discuss generalized results for non-whitened inputs with non-trivial covariance.

Combining Theorem 3.1 with Theorem 3.2, we see that the frequency response determines which portions of the kernel spectrum receive large eigenvalues. Frequency regions with large $|H(\omega)|^2$ are therefore favored by the SSM-induced kernel, while regions with small response require more samples to learn reliably. This provides a concrete notion of spectral inductive bias for LTI SSMs. The remaining question is how to estimate which frequency regions are relevant to a task and how to align the model response with them.

3.4 Task-Dependent Initialization via Spectral Alignment

The previous sections show that the spectral inductive bias of an LTI SSM is governed by its frequency response. Specifically, frequency regions with larger model spectrum $S_{\text{model}}(\omega) = |H(\omega)|^2$ correspond to larger portions of the SSM-induced kernel spectrum and are therefore favored in the low-sample regime. This suggests a task-adaptive strategy: before downstream training, initialize the SSM so that its high-response frequency regions are aligned with frequency regions that are statistically predictive of the task. We refer to this principle as task-model spectral alignment.

To make this principle concrete and to quantify the task spectrum, we first introduce a task-relevant spectral statistic based on input-output cross-covariance. Given training samples $\{(\boldsymbol{u}^\mu, \boldsymbol{Y}^\mu)\}_{\mu=1}^P$, where $\boldsymbol{u}^\mu \in \mathbb{R}^L$ and $\boldsymbol{Y}^\mu \in \mathbb{R}^d$, we define $\mathbf{C}_{\boldsymbol{uY}} = \frac{1}{P}\sum_{\mu=1}^P (\boldsymbol{u}^\mu - \bar{\boldsymbol{u}})(\boldsymbol{Y}^\mu - \bar{\boldsymbol{Y}})^\top \in \mathbb{R}^{L\times d}$. For classification tasks, $\boldsymbol{Y}^\mu$ is the one-hot label vector. We then compute the Fourier transform along the sequence dimension and aggregate across output dimensions: $S_{\text{cross}}(\omega_j) = \frac{1}{d}\sum_{r=1}^d \left|\mathcal{F}\left[\mathbf{C}_{\boldsymbol{uY}}^{(:,r)}\right](\omega_j)\right|^2$, where $\omega_j = \frac{2\pi j}{L}$. The labels distinguish input variation associated with the prediction target from variation that merely carries large input power. Specifically, S_{cross} measures the spectral content of input variations that are linearly correlated with the output. It therefore provides an analyzable proxy for task-relevant spectrum, rather than a complete characterization of the task. We then define the spectral matching loss as the L_2 distance between the model spectrum and the task spectrum.

Definition 2 (Spectral Matching Loss)**.** *For nonzero model and task spectra S_{model} and S_{task}, we define the spectral matching loss as*

$$\mathcal{L}_{\text{spec}} = \left\| \frac{S_{\text{model}}}{\|S_{\text{model}}\|_2} - \frac{S_{\text{task}}}{\|S_{\text{task}}\|_2} \right\|_2^2. \tag{3}$$

The normalization makes the objective focus on the relative allocation of spectral power rather than the overall gain of the filter. In the theoretical analysis below, we take $S_{\text{task}} = S_{\text{cross}}$ because

the cross-spectrum admits a direct connection to model alignment. If $\boldsymbol{Y}$ is independent of $\boldsymbol{u}$, the population centered cross-covariance vanishes, and hence the population cross-spectrum is identically zero. In this case, the normalized loss is undefined, and the task statistic provides no preferred spectral alignment. Appendix F.5: Label Permutation during Task-Spectrum Estimation examines the role of labels empirically by permuting them only during task-spectrum estimation.

We propose *Task-Dependent Initialization (TDI)*, which aligns the initial SSM spectrum with the estimated task spectrum through peak initialization (see Implementation), optionally followed by gradient refinement of $\mathcal{L}_{\mathrm{spec}}$. After this initialization step, the model is trained using the standard supervised task loss, with the same architecture, parameter count, and downstream training protocol as the baseline. Thus, TDI should be interpreted not as an additional model component, but as a theory-motivated way to choose an initial spectral inductive bias. We provide a two-part theoretical justification of this method in Appendix B.3: Theoretical Analysis of Task-Dependent Initialization. Then, we obtain our final theorem.

Theorem 3.3 (Spectral Matching Promotes Task-Model Spectral Alignment and Generalization)**.** *Minimizing spectral matching loss $\mathcal{L}_{\mathrm{spec}}$ is equivalent to maximizing the normalized alignment objective $J/(\|S_{\mathrm{model}}\|_2\|S_{\mathrm{task}}\|_2)$ (Lemma B.7), which concentrates task-relevant projection power toward higher-eigenvalue kernel modes and thereby yields a faster-rising cumulative-power curve $C(\rho)$ (Lemma B.6). Consequently, the resulting spectral alignment improves finite-sample generalization.*

Implementation. TDI first estimates and normalizes a task-relevant spectrum from the available training data. For classification, Fisher log-power provides a practical alternative to the cross-spectrum used in our theory: we compute the log-power features $z(\omega) = \log(|\mathrm{rFFT}(\boldsymbol{u})(\omega)|^2 + \epsilon)$ and score each frequency by the ratio of class-prior-weighted between-class variance to within-class variance plus a small regularizer. This emphasizes frequencies whose spectral energy distinguishes labels consistently (Appendix C: Task Spectrum Estimation by Fisher Log-Power).

We then select dominant frequencies and map them to frequency-selective SSM modes (peak initialization). For a complex diagonal SSM, a selected frequency f_j, measured in cycles per input step, is encoded by a stable pole $A_j = -\gamma_j - i\,2\pi f_j/\Delta$, where $\gamma_j > 0$ controls damping and $\Delta > 0$ is the discretization step. Real-valued SSMs use corresponding stable 2×2 resonant blocks. Kernel regression and trainable one-layer SSM experiments further refine this peak initialization by minimizing $\mathcal{L}_{\mathrm{spec}}$. Deep SSM experiments instead use the constructed initialization directly, without gradient refinement, and apply TDI only to the first SSM layer. Downstream training otherwise follows the baseline protocol, and TDI introduces no inference-time overhead. Full algorithms and experimental settings are provided in Appendix D: Task-Dependent Initialization Algorithms and Appendix E: Additional Experimental Details.

We provide a bootstrap diagnostic for assessing the stability of the estimated task spectrum without model training (Appendix C: Task Spectrum Estimation by Fisher Log-Power). The observed stability transitions inform an empirical screen for considering TDI, but do not guarantee positive downstream gains.

4 Experiments

4.1 Empirical Validation of the Kernel's Eigen-Decomposition

Before evaluating TDI, we first validate the kernel-theoretic characterization underlying our method. Theorem 3.1 predicts that, in whitened input coordinates, the SSM-induced kernel has eigenfunctions given by the right singular directions of the Toeplitz convolution operator, with eigenvalues $\eta_\rho = s_\rho^2$.

On MNIST [18], the empirical spectrum and eigenfunctions closely match this prediction (Figure 2(a,b)). The induced expansion also accurately captures the cumulative power $C(\rho)$ (Figure 2(c)). Finally, eigenspectrum-based generalization predictions align with empirical kernel ridge regression across sample sizes, up to a small residual from the empirical tail (Figure 2(d)). These results provide a sanity check that the SSM-induced kernel analysis captures the relevant spectral structure used in the later TDI experiments. We include additional diagnostics, including unwhitened-input results and the small empirical tail beyond the theoretical rank, in Appendix E.1: Empirical Validation of Kernel Eigen-Decomposition.

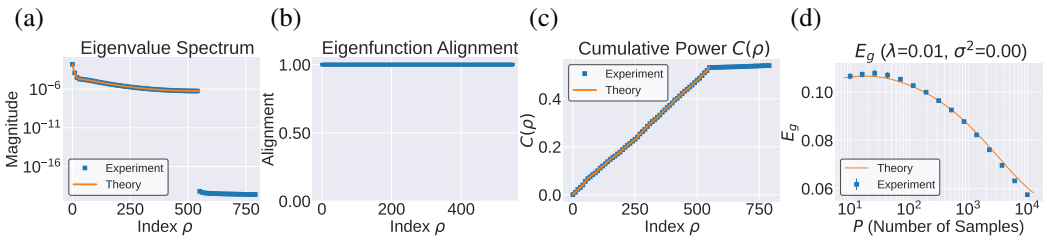


Figure 2: **Sanity check of the SSM-induced kernel eigendecomposition.** (a) The empirical eigenvalue spectrum matches the theoretical prediction $\eta_\rho = s_\rho^2$ obtained from the singular values of the Toeplitz convolution operator. (b) The empirical and predicted eigenfunctions are well aligned across most modes. (c) The predicted eigenbasis accurately captures the cumulative task power $C(\rho)$. (d) Kernel regression generalization predicted from the eigenspectrum matches empirical KRR performance across sample sizes. Error bars indicate standard deviations over 50 independent runs.

4.2 Kernel Regression: TDI Corrects Spectral Mismatch

We next test the mechanism predicted by our theory in a controlled kernel regression setting. This experiment asks whether modifying the SSM spectrum according to the task spectrum changes the generalization behavior of the model.

We fix the initial model to be a low-frequency SSM and compare two synthetic tasks with different spectral structure (see Appendix E.2: Kernel Regression Experimental Details for details). In the matched case, the task is also low-frequency, so the initial model spectrum already substantially overlaps with the task-relevant frequencies. In the mismatched case, the task is high-frequency, so the task-relevant frequencies are weakly represented by the initial model. This comparison isolates the effect of spectral mismatch while keeping the model class, initialization method, and kernel regression protocol fixed.

For each task, we apply TDI by estimating the task spectrum and aligning the SSM frequency response with it (Appendix D: Task-Dependent Initialization Algorithms). We compare the original and TDI-aligned kernels using the model spectrum, cumulative power $C(\rho)$, and kernel regression generalization curve. Figure 3 shows that, in the matched case, TDI makes only minor changes because the initial spectrum is already close to the task spectrum; the cumulative-power and generalization curves therefore remain similar. In the mismatched case, however, the high-frequency task initially places much of its power in late, small-eigenvalue modes. TDI shifts the model spectrum toward these task-relevant frequencies, moves the target power earlier in the sorted kernel eigenbasis, and lowers the generalization curve across the finite-sample range. Additional parameter settings in Appendix E.2: Kernel Regression Experimental Details show the same qualitative trend. These results support the mechanism predicted by our analysis: TDI improves data efficiency by correcting spectral mismatch, effectively converting task-relevant directions from late-learned modes into early-learned modes of the SSM-induced kernel.

4.3 Trainable One-Layer SSMs

We next test whether the spectral alignment mechanism predicted by our theory continues to govern generalization when the recurrent dynamics are trainable. We use a simple one-layer linear SSM followed by a nonlinear readout. The SSM baseline and TDI model have the same architecture and number of trainable parameters; TDI only changes the initialization using the estimated task spectrum. We evaluate on seven sequence-classification tasks: a synthetic binary frequency-classification task, sequential MNIST [18], and five univariate time-series datasets from the UCR archive [17]: ECG5000, FordA, FordB, TwoPatterns, and Wafer. For each dataset, we sweep the training ratio over $\{0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.0\}$. A training ratio r means that we train on an r fraction of the original training split, while keeping the validation and test splits fixed. Throughout the experiments, "low-data" and "high-data" refer to lower and higher within-dataset training ratios, respectively, rather than comparable absolute sample counts or task difficulty across datasets. Full model structure, dataset statistics, and hyperparameters are provided in Appendix E.3: Trainable One-Layer SSM Experimental Details. Unless otherwise stated, dataset-level summaries average over ratios within each regime after averaging seeds, while paired improvement tables use matched dataset-

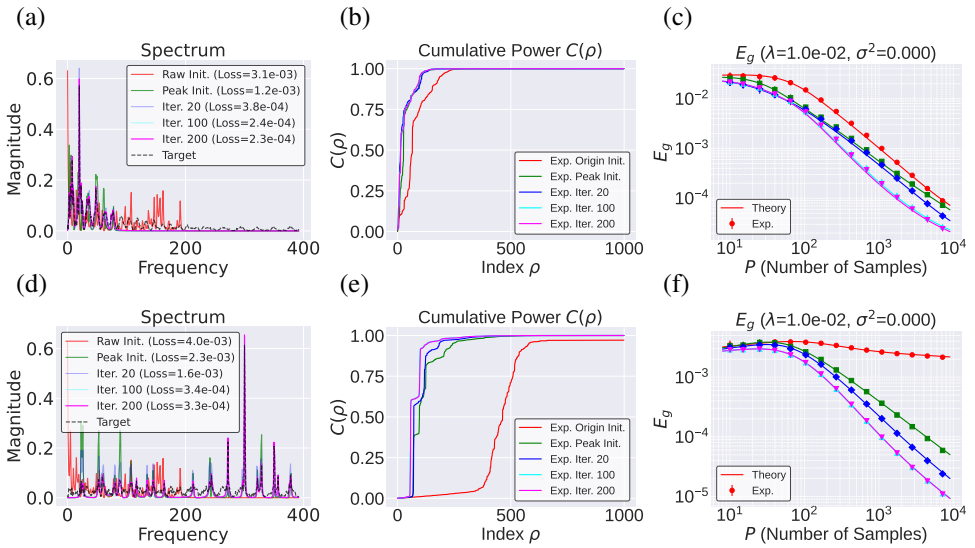


Figure 3: **Kernel regression mechanism experiment.** We fix the initial model to be a low-frequency SSM and compare a matched low-frequency task with a mismatched high-frequency task. (a)-(c) In the *matched* case, the initial model spectrum already overlaps with the task spectrum, so TDI produces little change in the model spectrum, cumulative task power, or generalization. (d)-(f) In the *mismatched* case, the task-relevant high-frequency components are weakly represented by the initial model. TDI shifts the model spectrum toward these frequencies, moves task power into earlier high-eigenvalue modes, and shifts the kernel regression generalization curve downward across the finite-sample range. This supports the view that TDI improves data efficiency by correcting spectral mismatch rather than by universally improving all initializations.

ratio-seed triples with cluster bootstrap confidence intervals. Full per-dataset, per-ratio comparisons among the SSM baseline, TDI, and the Standard RNN baseline are provided in Appendix F.1: Complete Results for Trainable One-Layer SSMs.

Table 1 shows that TDI achieves higher mean final accuracy on all seven datasets in the low-data regime and reduces final loss on most of them. The largest gains appear on ECG5000, sequential MNIST, and Wafer. In the high-data regime, the trend is more mixed: TDI still helps on several datasets, but the gains are smaller and can reverse on some tasks. This supports our interpretation that TDI mainly improves finite-sample learning by correcting spectral mismatch, rather than acting as a universally better initialization. Table 2 further confirms this trend with paired comparisons over

Table 1: **Dataset-level low- and high-data summary for the one-layer SSM baseline and TDI.** Low-data averages ratios ≤ 0.16; high-data averages ratios ≥ 0.32. Each cell reports the mean across ratios within the corresponding regime, where each ratio is first averaged over 3 random seeds; the standard deviation is computed across these ratio-level means. Accuracy and loss are the final-epoch test metrics. The better value between baseline and TDI is bolded.

	Low-data regime				High-data regime			
Dataset	Base Acc.	TDI Acc.	Base Loss	TDI Loss	Base Acc.	TDI Acc.	Base Loss	TDI Loss
Binary Freq.	86.5±9.0%	**88.6±6.8%**	0.292±0.163	**0.256±0.135**	**97.5±0.5%**	97.3±0.6%	**0.070±0.012**	0.076±0.014
ECG5000	62.9±11.8%	**72.8±14.0%**	1.158±0.196	**0.981±0.615**	90.2±1.6%	**90.5±1.1%**	0.375±0.065	**0.370±0.053**
FordA	52.2±3.5%	**56.8±7.7%**	**0.689±0.016**	0.837±0.413	68.9±5.6%	**70.4±5.1%**	0.569±0.065	**0.548±0.052**
FordB	53.5±3.5%	**55.0±3.9%**	**0.729±0.054**	0.763±0.138	60.6±2.7%	**61.7±2.8%**	0.669±0.039	**0.640±0.014**
sMNIST	61.1±13.3%	**78.4±11.5%**	1.144±0.378	**0.696±0.353**	79.3±26.5%	**88.2±6.2%**	0.375±0.144	**0.364±0.175**
TwoPatterns	28.7±4.3%	**30.8±6.3%**	1.609±0.778	**1.479±0.136**	**67.5±17.5%**	52.6±13.5%	**0.788±0.328**	1.021±0.258
Wafer	79.8±9.5%	**87.4±6.2%**	0.486±0.111	**0.323±0.115**	94.9±2.9%	**95.5±3.3%**	0.129±0.071	**0.125±0.096**

matched dataset-ratio-seed triples. Final accuracy gains are positive from 1% to 32% training data, but weaken at 64% and 100%. Mean dataset-level low-data gains are positive across all datasets, although the confidence intervals include zero for Binary Frequency Classification and FordB; early-epoch improvements are more dataset-dependent. Thus, TDI is not merely a fixed-kernel artifact: task-aligned spectral initialization can also improve data efficiency in trainable one-layer SSMs without adding parameters or changing the architecture.

Table 2: **Paired accuracy improvements of TDI over the one-layer SSM baseline.** Each paired point matches the same dataset, ratio, and seed. Final uses the final test accuracy; Early-5 and Early-10 use the mean test accuracy over the first 5 and 10 epochs. All entries are percentage-point improvements with cluster bootstrap 95% confidence intervals. (a) Ratio-level improvements with datasets as bootstrap clusters. (b) Dataset-level low-data improvements with training ratios as bootstrap clusters. In each cluster, values are first averaged over 3 random seeds.

(a) Ratio-level improvement

Ratio	Final ΔAcc.	Early-5 ΔAcc.	Early-10 ΔAcc.
0.01	+5.4 [+1.8, +10.3]	+3.5 [-0.7, +9.7]	+3.2 [+0.8, +6.1]
0.02	+7.5 [+0.9, +15.1]	-0.2 [-2.7, +3.1]	+2.1 [-0.4, +5.1]
0.04	+6.5 [+2.3, +11.2]	+3.2 [-1.5, +9.0]	+3.6 [+1.2, +6.8]
0.08	+7.3 [+4.0, +10.8]	+2.7 [+0.4, +5.3]	+4.7 [+3.2, +6.1]
0.16	+5.6 [+1.5, +10.0]	+1.7 [-1.2, +4.6]	+2.7 [-0.1, +5.7]
0.32	+2.3 [+0.7, +4.0]	+1.5 [+0.0, +3.1]	+1.3 [-0.1, +2.9]
0.64	-1.2 [-6.7, +2.3]	-1.2 [-4.3, +1.6]	-0.6 [-2.2, +1.0]
1.00	-2.3 [-14.1, +7.3]	-0.6 [-4.4, +3.0]	-1.6 [-7.8, +3.3]

(b) Dataset-level low-data improvement

Dataset	Final ΔAcc.	Early-5 ΔAcc.	Early-10 ΔAcc.
Binary Freq.	+2.2 [-0.3, +4.5]	+3.3 [+0.3, +6.4]	+2.4 [-0.0, +4.8]
ECG5000	+9.9 [+4.0, +16.9]	+6.9 [-2.1, +16.0]	+6.1 [+1.2, +10.7]
FordA	+4.7 [+0.7, +9.4]	+1.0 [-0.8, +3.7]	+2.5 [+0.1, +6.1]
FordB	+1.5 [-1.1, +4.7]	+0.2 [-0.8, +1.0]	+0.4 [-0.7, +1.6]
sMNIST	+17.3 [+15.7, +19.4]	-4.1 [-5.6, -2.5]	+2.4 [+0.3, +4.8]
TwoPatterns	+2.0 [+0.1, +4.0]	+3.3 [+1.8, +5.0]	+3.0 [+1.4, +4.8]
Wafer	+7.6 [+4.8, +10.2]	+4.7 [+2.7, +6.7]	+6.2 [+5.0, +7.3]

4.4 Deep SSMs

We finally evaluate whether the spectral alignment mechanism underlying TDI continues to hold in deep SSMs. We compare the S4D [24] baseline with TDI using the same architecture, downstream training protocol, and number of trainable parameters. TDI is applied only to the first S4 layer, while all later layers are initialized and trained as in the baseline (Appendix D: Task-Dependent Initialization Algorithms). We evaluate on eight sequence benchmarks: synthetic Frequency Classification, sequential and permuted MNIST [18], PathFinder [47], CIFAR-10 [32], ListOps [35, 47], Speech Commands, and SC10 [48]. The training ratios are logarithmically spaced between 10^{-3} and 1. Other experimental details are provided in Appendix E.4: Deep SSM Experimental Details. Full per-dataset and per-ratio results, including the TDI-Frozen variant where the TDI-initialized first S4 layer is kept fixed during supervised training, are provided in Appendix F.2: Complete Results for Deep SSMs.

Table 3 summarizes final-epoch accuracy and loss in low- and high-data regimes. In the low-data regime, TDI achieves higher mean final accuracy on most datasets, although the paired confidence intervals include zero for several datasets (Table 4). The trend is more mixed in the high-data regime, where some datasets still benefit but others match or underperform the baseline. This is consistent with our interpretation that TDI primarily improves data-efficient learning by correcting spectral mismatch, rather than acting as a universally better initialization. Table 4 provides paired comparisons over matched dataset-ratio-seed triples. Final accuracy gains are positive across much of the low-data range, but are small or uncertain at the extremely small ratios, where the task spectrum is difficult to estimate reliably. The gains also weaken at larger training ratios. Early-5 and Early-10 improvements are positive on most datasets with clearer spectral structure, suggesting that TDI often improves the initial learning trajectory. However, gains are limited on ListOps and PathFinder, whose labels may be less directly characterized by a one-dimensional input frequency spectrum. This negative evidence is consistent with the limitation of using a one-dimensional spectral statistic for tasks whose label structure is more compositional or spatial. Overall, the deep SSM results support the same conditional claim as the previous experiments: TDI is most useful in finite-sample regimes where task-relevant spectral structure is present and the default SSM initialization is not already well aligned with it.

Early learning versus extended-training accuracy. We extend training from 30 to 100 epochs on sMNIST and from 10 to 50 epochs on Frequency Classification, using five matched seeds. On sMNIST with 2.15% training data, TDI reduces the mean number of epochs required to reach 80% test accuracy from 34.2 to 20.0, while the final accuracy difference is only $+0.12 \pm 0.47$ percentage points (mean $\pm$ SD). Frequency Classification similarly shows an early-learning advantage in the low-data setting, with small final accuracy differences after extended training. These results distinguish faster early learning from systematically higher accuracy after longer training (Appendix F.3: Early Learning versus Extended Training). As complementary evidence on early learning across the original training budgets, TDI achieves higher Early-10 accuracy in 45/70 deep SSM dataset–ratio conditions, and in 39/50 conditions among the six tasks (excluding PathFinder and ListOps) better matched to the current one-dimensional spectral statistic.

Table 3: **Dataset-level low- and high-data summary for the deep SSM baseline and TDI.** Low-data averages ratios ≤ 0.100; high-data averages ratios ≥ 0.215. Each cell reports the mean across ratios within the corresponding regime, where each ratio is first averaged over 5 random seeds; the standard deviation is computed across these ratio-level means. Accuracy and loss are final-epoch test metrics. The better value between baseline and TDI is bolded.

	Low-data regime				High-data regime			
Dataset	Base Acc.	TDI Acc.	Base Loss	TDI Loss	Base Acc.	TDI Acc.	Base Loss	TDI Loss
CIFAR-10	33.9±15.7%	**35.4±14.5%**	2.026±0.190	**2.012±0.184**	**73.7±7.2%**	69.4±8.5%	**0.948±0.344**	1.094±0.406
Freq. Classification	88.1±18.9%	**91.1±11.5%**	0.385±0.590	**0.331±0.439**	**99.2±0.4%**	98.9±0.5%	**0.024±0.014**	0.026±0.012
ListOps	**16.6±7.1%**	16.2±6.6%	**5.419±0.754**	5.597±0.941	**44.7±6.9%**	43.2±6.2%	1.928±0.786	**1.816±0.584**
sMNIST	56.3±42.2%	**67.4±38.1%**	1.145±1.084	**0.897±0.991**	**99.1±0.2%**	99.0±0.2%	**0.034±0.011**	0.038±0.011
Pathfinder	51.4±2.4%	**51.7±2.6%**	3.482±0.853	**3.343±0.864**	66.6±10.4%	**68.4±8.1%**	0.935±0.567	**0.891±0.407**
pMNIST	52.2±38.6%	**52.6±39.1%**	1.294±0.945	**1.292±0.971**	**96.9±0.7%**	96.7±1.0%	**0.121±0.037**	0.130±0.051
Speech Commands	51.8±31.5%	**55.7±29.7%**	3.777±3.810	**3.716±4.483**	87.3±1.6%	**87.9±2.2%**	0.432±0.051	**0.415±0.079**
SC10	49.2±31.8%	**52.8±31.9%**	**6.003±7.895**	6.492±10.229	92.4±1.1%	**92.7±0.7%**	0.246±0.041	**0.242±0.035**

Table 4: **Paired accuracy improvements of TDI over the deep SSM baseline.** Each paired point matches the same dataset, training ratio, and seed. Final uses final-epoch test accuracy; Early-5 and Early-10 use mean test accuracy over the first 5 and 10 epochs. All entries are percentage-point improvements with cluster bootstrap 95% confidence intervals. (a) Ratio-level improvements with datasets as bootstrap clusters. (b) Dataset-level low-data improvements with training ratios as bootstrap clusters. In each cluster, values are first averaged over 5 random seeds.

(a) Ratio-level improvement

Ratio	Final ΔAcc	Early-5 ΔAcc.	Early-10 ΔAcc.
0.001	+0.0 [-0.3, +0.5]	+0.2 [+0.0, +0.5]	+0.8 [+0.0, +1.5]
0.002	+0.9 [-0.5, +3.2]	+0.8 [+0.0, +1.6]	+1.0 [+0.0, +2.3]
0.005	+2.0 [-0.0, +4.7]	+0.9 [-0.0, +1.9]	+1.5 [+0.1, +3.2]
0.010	+8.5 [+2.6, +15.6]	+5.2 [+0.4, +13.2]	+5.3 [+0.7, +12.3]
0.022	+3.9 [-0.7, +10.9]	+6.0 [+1.3, +11.7]	+4.4 [+1.2, +8.0]
0.046	+0.6 [+0.2, +1.0]	+3.8 [+0.5, +7.4]	+3.6 [+0.4, +7.5]
0.100	-0.3 [-1.2, +0.5]	+5.5 [-0.4, +13.4]	+7.0 [+0.3, +15.7]
0.215	-0.3 [-1.9, +0.9]	+2.4 [-1.8, +5.8]	+1.1 [-1.8, +3.3]
0.464	-0.7 [-2.2, +0.5]	+0.8 [-2.3, +4.0]	-0.4 [-2.6, +1.8]
1.000	-0.4 [-1.2, +0.4]	-1.8 [-4.1, -0.1]	-1.6 [-3.8, +0.1]

(b) Dataset-level low-data improvement

Dataset	Final ΔAcc	Early-5 ΔAcc.	Early-10 ΔAcc.
CIFAR-10	+1.5 [-0.8, +4.1]	+0.9 [+0.3, +2.0]	+1.4 [+0.5, +2.7]
Freq. Classification	+3.0 [-2.6, +11.6]	+14.7 [+3.4, +26.0]	+9.6 [+1.5, +21.9]
ListOps	-0.4 [-1.1, +0.5]	-0.6 [-2.7, +0.9]	-0.4 [-1.8, +0.8]
sMNIST	+11.1 [+0.1, +22.2]	+6.3 [-0.0, +18.2]	+10.0 [+0.3, +23.3]
Pathfinder	+0.2 [-0.0, +0.5]	-0.1 [-0.1, -0.1]	-0.1 [-0.2, -0.0]
pMNIST	+0.3 [-0.2, +0.7]	+1.4 [-0.4, +4.7]	+3.0 [-0.9, +9.6]
Speech Commands	+2.6 [+0.5, +5.2]	+6.2 [+2.5, +10.5]	+5.5 [+3.1, +8.4]
SC10	+3.7 [+0.8, +8.2]	+4.8 [+1.9, +7.7]	+4.8 [+2.2, +7.6]

5 Conclusion and Limitations

We introduced a kernel-theoretic framework for understanding the spectral inductive bias of LTI State Space Models. By viewing an SSM as a convolutional feature map, we showed that the spectrum of the induced kernel is governed by the SSM frequency response, connecting frequency-domain bias to mode-wise learnability in the small-sample regime. This analysis motivated Task-Dependent Initialization (TDI), a lightweight method that aligns the initial SSM spectrum with task-relevant spectral structure before training. Across controlled kernel regression, trainable one-layer SSMs, and deep SSMs, TDI is most beneficial when data are limited, label-relevant spectral structure is estimable, and the default initialization is spectrally mismatched. Improvements in early learning and accuracy under fixed training budgets should be distinguished from higher accuracy after extended training: on sMNIST and Frequency Classification, the baseline largely closes the gap with longer training.

This work also has several limitations. Our theory focuses on LTI SSMs, leaving extensions to selective architectures such as Mamba as an important direction. Under small, smooth input-dependent modulation around an LTI reference, selective dynamics may admit a local approximation consisting of the LTI operator and a first-order input-dependent correction, suggesting a possible direction for extending the analysis. We primarily analyze one-dimensional scalar sequences and apply TDI to the first SSM layer; an additional ablation shows that applying the same input-derived spectrum to all layers can help but is not uniformly preferable (Appendix F.4: First-Layer versus All-Layer TDI). For multivariate inputs, extending the convolutional representation to a block-Toeplitz operator with a matrix-valued frequency response is straightforward, while a full treatment of cross-channel task structure and principled layer-dependent initialization remains future work. Additionally, TDI also depends on reliable estimation of task-relevant spectra, which can be noisy at extremely small training ratios. Finally, TDI is not intended as a universal improvement: its benefits are most pronounced when task-relevant structure is spectral and mismatched with the default SSM bias.

Acknowledgments and Disclosures

Funding. This work was supported by the National Key R&D Program of China, Project Number 2025YFA1016700. This work was also supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62576011. This work was supported in part by the Beijing Major Science and Technology Project under Contract no. Z251100008125055. This work was also supported by Beijing Academy of Artificial Intelligence (BAAI).

Competing interests. The authors declare no competing interests.

References

[1] Amina Adadi. A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1): 24, 2021.

[2] Naman Agarwal, Daniel Suo, Xinyi Chen, and Elad Hazan. Spectral state space models. In *First Workshop on Long-Context Foundation Models@ ICML 2024*, 2024.

[3] Ido Amos, Jonathan Berant, and Ankit Gupta. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PdaPky8MUn.

[4] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[5] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

[6] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[7] Mikhail Belkin, Daniel J Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. *Advances in neural information processing systems*, 31, 2018.

[8] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International conference on machine learning*, pages 541–549. PMLR, 2018.

[9] Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. Does data interpolation contradict statistical optimality? In *The 22nd international conference on artificial intelligence and statistics*, pages 1611–1619. PMLR, 2019.

[10] Blake Bordelon and Cengiz Pehlevan. Population codes enable learning from few examples by shaping inductive bias. *eLife*, 11:e78606, dec 2022. ISSN 2050-084X. doi: 10.7554/eLife.78606. URL https://doi.org/10.7554/eLife.78606.

[11] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1):1–12, 2021.

[12] Abdulkadir Canatar, Evan Peters, Cengiz Pehlevan, Stefan M. Wild, and Ruslan Shaydulin. Bandwidth enables generalization in quantum kernel models. *Trans. Mach. Learn. Res.*, 2023, 2022. URL https://api.semanticscholar.org/CorpusID:249642463.

[13] Yunxiang Chen, Tao Wang, Wei Wang, and Feng Liu. Emergent attractor-like dynamics support stable and generalizable path integration in recurrent neural networks. *Cognitive neurodynamics*, 20(1):179, 2026. ISSN 1871-4080. doi: 10.1007/s11571-026-10555-8.

[14] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. *Advances in Neural Information Processing Systems*, 34:11933–11944, 2021.

[15] Cucker and Smale. Best choices for regularization parameters in learning theory: On the bias—variance problem. *Foundations of computational Mathematics*, 2(4):413–428, 2002.

[16] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

[17] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. doi: 10.1109/JAS.2019.1911747.

[18] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

[19] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022.

[20] Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006.

[21] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=tEYskw1VY2`.

[22] 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.

[23] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.

[24] Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35, 2022.

[25] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in neural information processing systems*, 35:22982–22994, 2022.

[26] László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer, 2002.

[27] Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

[28] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[29] Cecilia Jarne. Different eigenvalue distributions encode the same temporal tasks in recurrent neural networks. *Cognitive neurodynamics*, 17(1):257–275, 2023. ISSN 1871-4080. doi: 10.1007/s11571-022-09802-5.

[30] Cecilia Jarne and Mariano Caruso. Effect in the spectra of eigenvalues and dynamics of rnns trained with excitatory-inhibitory constraint. *Cognitive neurodynamics*, 18(3):1323–1335, 2024. ISSN 1871-4080. doi: 10.1007/s11571-023-09956-w.

[31] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[32] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[33] Fusheng Liu and Qianxiao Li. From generalization analysis to optimization designs for state space models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[34] Fusheng Liu and Qianxiao Li. Autocorrelation matters: Understanding the role of initialization schemes for state space models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[35] Nikita Nangia and Samuel R Bowman. Listops: A diagnostic dataset for latent tree learning. *arXiv preprint arXiv:1804.06028*, 2018.

[36] Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 2022.

[37] Badri Narayana Patro and Vijay Srinivas Agneeswaran. Mamba-360: Survey of state space models as transformer alternative for long sequence modelling: Methods, applications, and challenges. *Engineering Applications of Artificial Intelligence*, 159:111279, 2025.

[38] Sidney Pontes-Filho, Pedro Lind, Anis Yazidi, Jianhua Zhang, Hugo Hammer, Gustavo B. M. Mello, Ioanna Sandvig, Gunnar Tufte, and Stefano Nichele. A neuro-inspired general framework for the evolution of stochastic dynamical systems: Cellular automata, random boolean networks and echo state networks towards criticality. *Cognitive Neurodynamics*, 14(5):657–674, 2020. doi: 10.1007/s11571-020-09600-x. URL https://doi.org/10.1007/s11571-020-09600-x.

[39] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

[40] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[41] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[42] James B Simon, Madeline Dickens, Dhruva Karkada, and Michael R DeWeese. The eigenlearning framework: A conservation law perspective on kernel ridge regression and wide neural networks. *Transactions on Machine Learning Research*, 2023.

[43] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. In *ICLR*, 2023.

[44] Ruben Solozabal, Velibor Bojkovic, Hilal AlQuabeh, Kentaro Inui, and Martin Takáč. Uncovering the spectral bias in diagonal state space models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4X2UwlYsXt.

[45] Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, 2020.

[46] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[47] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qVyeW-grC2k.

[48] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[49] Annan Yu, Dongwei Lyu, Soon Hoe Lim, Michael W. Mahoney, and N. Benjamin Erichson. Tuning frequency bias of state space models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wkHcXDv7cv.

A Related Work

Inductive bias and kernel generalization. Inductive bias shapes generalization by steering models toward solutions aligned with specific assumptions [27]. In classical kernel regression, the kernel itself encodes the inductive bias [40], and a large body of work has investigated how different kernels shape generalization [15, 26, 8, 7, 9, 45]. In particular, Canatar et al. [11], Bordelon and Pehlevan [10] derived analytical characterizations of the generalization error for finite datasets and arbitrary kernels, revealing how spectral bias and code–task alignment govern kernel generalization. Our work builds on this view by deriving the kernel induced by an LTI SSM convolution operator and using its spectrum to characterize the model's frequency-dependent inductive bias.

State space models and spectral bias. State space models (SSMs) have recently emerged as effective architectures for modeling long sequences [37, 22, 23, 24, 25, 36, 43]. Recent selective SSMs such as Mamba [21, 16] introduce input-dependent dynamics and achieve strong empirical performance, while LTI SSMs such as S4 and S4D remain an important analytically tractable setting for understanding the filtering behavior underlying this model family. The frequency bias of LTI SSMs has recently attracted attention: Yu et al. [49], Solozabal et al. [44] analyze and tune this bias, and Agarwal et al. [2] propose spectral initializations to better control frequency response. In contrast, we connect the SSM frequency response to an induced kernel spectrum and explicitly relate this spectrum to task-model alignment and sample efficiency.

Recurrent dynamics and spectral structure. Beyond SSMs, spectral and dynamical properties of recurrent networks have been studied as a way to characterize temporal computation. Prior work has related the eigenspectrum of recurrent connectivity to task-dependent dynamics in trained RNNs, and examined how biologically motivated excitatory–inhibitory constraints reshape these spectra and the resulting dynamics [29, 30]. Related neuro-inspired work has investigated how recurrent dynamical substrates and criticality support computation [38], while recent task-trained RNN models show that structured attractor-like dynamics can emerge through learning and support robust generalization [13]. In contrast, our work focuses on LTI SSMs and establishes an explicit link from frequency response to an induced kernel spectrum, mode-wise learnability, and finite-sample generalization.

Initialization and data-dependent adaptation. Initialization plays an important role in SSM performance, as the initial state dynamics determine which temporal modes are emphasized before training [24, 34]. More broadly, data-dependent initialization and training schemes aim to adapt model biases to the structure of the data [33, 34]. Prior work studies SSM frequency bias tuning [49] and data-dependent initialization [34], but does not connect it to kernel spectra or mode-wise learnability. Another line of research uses data-driven self-pretraining as initialization before downstream training [3]. Instead, we provide a kernel-theoretic link from frequency response to sample efficiency. Additionally, our method focuses on a lightweight, task-dependent spectral initialization: TDI aligns the initial SSM response to task-specific spectral statistics estimated from input-output pairs, unlike input-agnostic or generic frequency-bias methods.

B Proofs and Theoretical Analysis

This section provides complete proofs of all the theorems and lemmas stated in the main text.

B.1 Properties of the SSM-Induced Kernel

We first present proofs characterizing the SSM-induced kernel. We begin by recalling our definition of SSM-induced kernel in Section 3.1.

Lemma B.1 (Positive Semi-Definiteness)**.** *The SSM-induced kernel $K_{\mathrm{SSM}}(\boldsymbol{u}, \boldsymbol{u}')$ is a valid kernel.*

We first show our kernel is valid by Lemma B.1.

Proof of Lemma B.1. By definition, for any set of input sequences $\{\boldsymbol{u}^1, \cdots, \boldsymbol{u}^P\}$, the Gram matrix $\mathbf{K}_{\mathrm{SSM}}$ has entries $\mathbf{K}_{\mu\nu} = K_{\mathrm{SSM}}(\boldsymbol{u}^\mu, \boldsymbol{u}^\nu) = \frac{1}{L}(\boldsymbol{T}_h \boldsymbol{u}^\mu)^\top (\boldsymbol{T}_h \boldsymbol{u}^\nu)$. Let $\mathbf{U} = [\boldsymbol{u}^1, \cdots, \boldsymbol{u}^P]$ be the matrix of input sequences. The Gram matrix can be written in matrix form as $\mathbf{K} = \frac{1}{L}(\boldsymbol{T}_h \mathbf{U})^\top (\boldsymbol{T}_h \mathbf{U})$. For any non-zero vector $z \in \mathbb{R}^P$, we have:

$$\boldsymbol{z}^\top \mathbf{K} \boldsymbol{z} = \frac{1}{L} \boldsymbol{z}^\top (\boldsymbol{T}_h \mathbf{U})^\top (\boldsymbol{T}_h \mathbf{U}) \boldsymbol{z} = \frac{1}{L} \left\| (\boldsymbol{T}_h \mathbf{U}) \boldsymbol{z} \right\|_2^2 \geq 0.$$

Therefore, since the Gram matrix $\mathbf{K}$ is positive semi-definite (and symmetric obviously), the kernel $K_{\mathrm{SSM}}(\boldsymbol{u}, \boldsymbol{u}')$ is valid. □

Inserting the singular value decomposition (SVD) of Toeplitz matrix $\boldsymbol{T}_h = \sqrt{L}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ into Definition 1, we obtain $K_{\mathrm{SSM}}(\boldsymbol{u}, \boldsymbol{u}') = (\boldsymbol{V}^\top \boldsymbol{u})^\top \boldsymbol{S}^2 (\boldsymbol{V}^\top \boldsymbol{u}')$. Then, we prove rows of $\boldsymbol{V}^\top \boldsymbol{u}$ form a set of eigenfunctions of the kernel.

Lemma B.2 (Orthonormal Eigenfunctions)**.** *For whitened input coordinates, rows of $\boldsymbol{V}^\top \boldsymbol{u}$ form a set of orthonormal eigenfunctions for the SSM-induced kernel.*

Proof of Lemma B.2. Let $\boldsymbol{v}_i^\top$ denote the ith row of the right singular matrix $\boldsymbol{V}^\top$. For any integer $i, j \in [1, L]$,

$$\int (\boldsymbol{v}_i^\top \boldsymbol{u}) \cdot (\boldsymbol{v}_j^\top \boldsymbol{u}) p(\boldsymbol{u}) d\boldsymbol{u} = \mathbb{E}_{\boldsymbol{u}} \left[\left(\boldsymbol{V}^\top \boldsymbol{u} \boldsymbol{u}^\top \boldsymbol{V} \right)_{ij} \right] = \left(\boldsymbol{V}^\top \mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}\boldsymbol{u}^\top] \boldsymbol{V} \right)_{ij} = \delta_{ij}.$$

□

The finite-dimensional decomposition above, together with Lemma B.2, establishes Theorem 3.1.

Theorem 3.1 (Eigen-Decomposition of SSM-Induced Kernel)**.** *Assume $\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}\boldsymbol{u}^\top] = \boldsymbol{I}_L$. Then the SSM-induced kernel admits the finite expansion $K_{\mathrm{SSM}}(\boldsymbol{u}, \boldsymbol{u}') = \boldsymbol{u}^\top \boldsymbol{V} \boldsymbol{S}^2 \boldsymbol{V}^\top \boldsymbol{u}' = \sum_{\rho=1}^{L} \eta_\rho \phi_\rho(\boldsymbol{u}) \phi_\rho(\boldsymbol{u}')$, where $\eta_\rho = s_\rho^2$ and $\phi_\rho(\boldsymbol{u}) = \boldsymbol{v}_\rho^\top \boldsymbol{u}$. These eigenfunctions satisfy $\mathbb{E}_{\boldsymbol{u}}[\phi_\rho(\boldsymbol{u}) \phi_\sigma(\boldsymbol{u})] = \delta_{\rho\sigma}$. Consequently, the kernel has rank at most L.*

B.2 Frequency-Domain Characterization of the SSM Spectrum

In this part, we analyze the relation between singular values of the SSM's Toeplitz matrix (also eigenvalues of SSM-induced kernel) and the frequency response of SSM (Theorem 3.2).

Theorem 3.2 (Frequency response governs the SSM-induced spectrum)**.** *For an SSM with absolutely summable impulse response h, i.e., $\sum_{n=0}^{\infty} |h_n| < \infty$, the singular values of the Toeplitz convolution operator $\boldsymbol{T}_h$ are asymptotically governed by the SSM frequency response. In particular, as $L \to \infty$, $\tilde{s}_j \sim |H(\omega_j)|/\sqrt{L}, \omega_j = \frac{2\pi j}{L}$. Equivalently, the eigenvalues of the SSM-induced kernel are governed by $\tilde{\eta}_j \sim |H(\omega_j)|^2$ up to a scale factor determined by the kernel feature map normalization.*

Proof. We begin by introducing a lemma from Gray et al. [20].

Lemma B.3 (Lemma 4.6 of Gray et al. [20])**.** *Let $\boldsymbol{T}_n(f) = \{t_{k-j}\}_{j,k=0,\cdots,n-1}$ where*

$$\sum_{k=-\infty}^{\infty} |t_k| < \infty,$$

and

$$f(\lambda) = \sum_{k=-\infty}^{\infty} t_k e^{ik\lambda}, \quad \hat{f}_n(\lambda) = \sum_{k=-(n-1)}^{n-1} t_k e^{ik\lambda}.$$

Define the circulant matrices $\boldsymbol{C}_n(f)$ with top row $(c_0^{(n)}, \cdots, c_{n-1}^{(n)})$, and $\boldsymbol{C}_n(\hat{f})$ with top row $(\hat{c}_0^{(n)}, \cdots, \hat{c}_{n-1}^{(n)})$ where

$$c_k^{(n)} = \frac{1}{n}\sum_{j=0}^{n-1} f(2\pi j/n)e^{2\pi ijk/n}, \quad \hat{c}_k^{(n)} = \frac{1}{n}\sum_{j=0}^{n-1} \hat{f}_n(2\pi j/n)e^{2\pi ijk/n}.$$

Then $\boldsymbol{C}_n(f) \sim \boldsymbol{C}_n(\hat{f}_n) \sim \boldsymbol{T}_n$ as $n \to \infty$.

Lemma B.3 states that as $n \to \infty$, $\boldsymbol{T}_n$ is asymptotically equivalent to a circulant matrix $\boldsymbol{C}_n(f)$ or $\boldsymbol{C}_n(\hat{f})$. Based on the fact that the eigenvalues of a circulant matrix are given by the discrete Fourier transform of its top row $(c_0^{(n)}, \cdots, c_{n-1}^{(n)})$,

$$\lambda_k = \sum_{j=0}^{n-1} c_j^{(n)} e^{-2\pi ijk/n},$$

we obtain

$$\lambda_k(\boldsymbol{C}_n(f)) = \sum_{j=0}^{n-1}\frac{1}{n}\sum_{l=0}^{n-1} f(2\pi l/n)e^{2\pi ijl/n}e^{-2\pi ijk/n} = f(2\pi k/n) = \sum_{p=-\infty}^{\infty} t_p e^{ip\cdot 2\pi k/n}.$$

Since $\boldsymbol{C}_n$ is circulant, its singular values s_k are equal to the absolute value of its eigenvalues λ_k. Thus we arrive at

$$s_k(\boldsymbol{T}_n) \sim s_k(\boldsymbol{C}_n) = |\lambda_k(\boldsymbol{C}_n)| = \left|\sum_{p=-\infty}^{\infty} t_p e^{ip\cdot 2\pi k/n}\right|.$$

For the SSM's Toeplitz matrix, $t_p = h_{-p}$ (see Section 2.2). Then, $t_{p>0} = 0, t_{p\le 0} = h_{-p}$ is the impulse response kernel of SSM. Then,

$$s_k = \left|\sum_{p=-\infty}^{\infty} t_p e^{ip\cdot 2\pi k/n}\right| = \left|\sum_{p=0}^{\infty} h_p e^{-ip\omega_k}\right| = |H(\omega_k)|,$$

where $\omega_k = \frac{2\pi k}{n}(k = 0, \cdots, n-1)$ and $H(\omega)$ is exactly the frequency response (or transfer function) of the SSM (see Section 2.2). We summarize this result in Theorem B.4.

Theorem B.4 (Asymptotic Behavior of Singular Values)**.** *For large n, the k-th singular value s_k of the SSM's Toeplitz matrix $\boldsymbol{T}_h \in \mathbb{R}^{n\times n}$ is asymptotically approximated by the frequency response $|H(\omega_k)|$, where $\omega_k = \frac{2\pi k}{n}$ for $k = 0, \ldots, n-1$.*

With Theorem B.4, it is straightforward to prove Theorem 3.2. □

Similar to the derivation of singular values, we can analogously show that the k-th row of $\boldsymbol{V}^\top$ corresponds to the Fourier basis with frequency ω_k.

Theorem B.5 (Asymptotic Behavior of Right Singular Vectors)**.** *For large n, the k-th row of the right singular matrix $\boldsymbol{V}^\top$ of the SSM's Toeplitz matrix $\boldsymbol{T}_h \in \mathbb{R}^{n\times n}$ is asymptotically approximated by the Fourier basis with frequency ω_k for $k = 0, \ldots, n-1$.*

It is worth noting that the singular values and the right singular matrix are not sorted here; the index k refers to the corresponding frequency $\omega_k = \frac{2\pi k}{n}$.

While we assume a unit input covariance for simplicity, our main results can be generalized. For a general input covariance $\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}\boldsymbol{u}^\top] = \boldsymbol{\Sigma}$, we reparameterize the input as $\boldsymbol{x} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}$, which ensures $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{I}_L$. Then, the original Toeplitz map $\boldsymbol{T}_h\boldsymbol{u}$ is equivalent to $\tilde{\boldsymbol{T}}\boldsymbol{x}$ where $\tilde{\boldsymbol{T}} = \boldsymbol{T}_h\boldsymbol{\Sigma}^{\frac{1}{2}} = \sqrt{L}\tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{V}}^\top$. This modified SVD result determines the eigen-decomposition by $\eta_\rho = \tilde{s}_\rho^2, \phi_\rho(\boldsymbol{u}) = \left(\tilde{\boldsymbol{V}}^\top\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}\right)_\rho$, for $\rho = 1, \cdots, L$. While $\tilde{s}_\rho$ does not admit a closed-form expression, one can demonstrate that s_ρ is strongly associated with the degree of alignment between the SSM and the

data (see below). Our analysis indicates that greater alignment leads to larger values of s_ρ, thereby accelerating the learning dynamics of ϕ_ρ.

In the unwhitened data case, $\tilde{\boldsymbol{T}}$ defined above can be expressed as $\tilde{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^\top$, where $\boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^\top = \boldsymbol{\Sigma}^{\frac{1}{2}}$. Thus, the SVD of $\tilde{\boldsymbol{T}}$ is determined by the SVD of $\boldsymbol{M} = \boldsymbol{S}(\boldsymbol{V}^\top\boldsymbol{Q})\boldsymbol{D}$, where $\tilde{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{U}_{\boldsymbol{M}}\boldsymbol{\Sigma}_{\boldsymbol{M}}\boldsymbol{V}_{\boldsymbol{M}}{}^\top\boldsymbol{Q}^\top$ for $\boldsymbol{M} = \boldsymbol{U}_{\boldsymbol{M}}\boldsymbol{\Sigma}_{\boldsymbol{M}}\boldsymbol{V}_{\boldsymbol{M}}{}^\top$. Since both $\boldsymbol{V}^\top$ and $\boldsymbol{Q}$ are orthogonal, $\boldsymbol{V}^\top\boldsymbol{Q}$ is itself an orthogonal change of basis: it maps the Fourier basis $\boldsymbol{V}^\top$ into the eigenbasis of the data covariance (columns of $\boldsymbol{Q}$). Therefore, the singular values of $\boldsymbol{M}$, and hence of $\tilde{\boldsymbol{T}}$, are governed by how the frequency response encoded in $\boldsymbol{S}$ aligns with the spectral distribution of the high-variance directions encoded in $\boldsymbol{D}$. In particular, a large singular value arises when a dominant data mode (large $\boldsymbol{D}_j$) has most of its energy concentrated in frequencies where the LTI operator exhibits strong gain, resulting in faster learning at the corresponding basis. Actually, our spectral matching method in Section 3.4 is inspired by this finding.

B.3 Theoretical Analysis of Task-Dependent Initialization

Finally, we present proofs showing that minimizing the spectral matching loss promotes task-model spectral alignment by increasing a cross-spectrum alignment objective.

Let $S_{\text{model}} = |H(\omega)|^2$ denote the model spectrum derived from the SSM's intrinsic frequency response and $\mathbf{C}_{\boldsymbol{u}\boldsymbol{Y}} = \mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}\boldsymbol{Y}^\top] \approx \frac{1}{P}\sum_{\mu=1}^{P}\boldsymbol{u}^\mu\boldsymbol{Y}^{\mu\top}$ denote the input-output cross-covariance matrix, where $\boldsymbol{u}$ and $\boldsymbol{Y}$ are centered. The spectral matching loss is defined in Section 3.4.

Definition 2 (Spectral Matching Loss)**.** *For nonzero model and task spectra S_{model} and S_{task}, we define the spectral matching loss as*

$$\mathcal{L}_{\text{spec}} = \left\| \frac{S_{\text{model}}}{\|S_{\text{model}}\|_2} - \frac{S_{\text{task}}}{\|S_{\text{task}}\|_2} \right\|_2^2. \tag{3}$$

We provide a two-part theoretical justification. We define the alignment objective J as the Frobenius norm of the product of the Toeplitz matrix and the input-output cross-covariance matrix, $J \equiv \frac{1}{L}\|\boldsymbol{T}_h\mathbf{C}_{\boldsymbol{u}\boldsymbol{Y}}\|_F^2$. First, we show that maximizing the objective J promotes a rapid increase in $C(\rho)$ (Lemma B.6). Then we show that minimizing spectral matching loss $\mathcal{L}_{\text{spec}}$ is an effective strategy for maximizing the objective J (Lemma B.7).

Lemma B.6. *Assume $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_L \geq 0$. The total task-relevant projection energy $E = \sum_{\rho=1}^{L}\overline{w}_\rho^2$ is invariant. The alignment objective satisfies*

$$J \equiv \frac{1}{L}\|\boldsymbol{T}_h\mathbf{C}_{\boldsymbol{u}\boldsymbol{Y}}\|_F^2 = \sum_{\rho=1}^{L}\eta_\rho\overline{w}_\rho^2.$$

For fixed $\{\eta_\rho\}_{\rho=1}^{L}$, increasing J favors concentrating task-relevant projection power in higher-eigenvalue kernel modes, and therefore favors a faster-rising cumulative-power curve $C(\rho)$. More precisely, any redistribution of projection power from a lower-eigenvalue mode to a higher-eigenvalue mode increases J and promotes rapid increase in $C(\rho)$.

Proof of Lemma B.6. Since the eigenfunctions of the SSM-induced kernel are linear, we write $\phi_\rho(\boldsymbol{u}) = \boldsymbol{\alpha}_\rho\boldsymbol{u}$. Let $\overline{w}_{\rho,j}$ denote the projection coefficient of the j-th component of $\boldsymbol{Y}$ onto ϕ_ρ:

$$\overline{w}_{\rho,j} = \mathbb{E}_{\boldsymbol{u}}[\phi_\rho(\boldsymbol{u})\boldsymbol{Y}_j] = \boldsymbol{\alpha}_\rho\mathbb{E}_{\boldsymbol{u}}[\boldsymbol{u}\,\boldsymbol{Y}_j] = \boldsymbol{\alpha}_\rho\mathbf{C}_j = \phi_\rho(\mathbf{C}_j),$$

where $\mathbf{C}_j$ is the j-th column of $\mathbf{C}_{\boldsymbol{u}\boldsymbol{Y}}$. Therefore,

$$E = \sum_{\rho=1}^{L}\overline{w}_\rho^2 = \sum_{\rho=1}^{L}\sum_{j=1}^{d}\phi_\rho(\mathbf{C}_j)^2 = \sum_{j=1}^{d}\|\mathbf{C}_j\|_2^2 = \|\mathbf{C}_{\boldsymbol{u}\boldsymbol{Y}}\|_F^2 \text{ is invariant,}$$

$$J \equiv \frac{1}{L}\|\boldsymbol{T}_h\mathbf{C}_{\boldsymbol{u}\boldsymbol{Y}}\|_F^2 = \sum_{\rho=1}^{L}\eta_\rho\sum_{j=1}^{d}\phi_\rho(\mathbf{C}_j)^2 = \sum_{\rho=1}^{L}\sum_{j=1}^{d}\eta_\rho\overline{w}_{\rho,j}^2 = \sum_{\rho=1}^{L}\eta_\rho\overline{w}_\rho^2.$$

The last step follows from the definition in Section 2.3.

To characterize how J depends on the allocation of projection power, consider transferring an amount $\delta > 0$ from mode j to mode i, with $i < j$ and $\eta_i > \eta_j$, while keeping the total projection energy E fixed. Then

$$J' - J = \eta_i \delta - \eta_j \delta = \delta(\eta_i - \eta_j) > 0.$$

Meanwhile, since the added power enters the cumulative sum at mode i and is removed only at mode j,

$$C'(k) - C(k) = \begin{cases} 0, & k < i, \\ \delta/E, & i \le k < j, \\ 0, & k \ge j. \end{cases}$$

Hence $C'(k) \ge C(k)$ for all k, with strict inequality for $i \le k < j$.

Therefore, for a fixed eigenspectrum and fixed total projection energy, redistributing task-relevant projection power toward higher-eigenvalue modes increases J and makes the cumulative-power curve rise faster. This establishes the preference encoded by maximizing J.

Although the eigenspectrum itself may change during spectral matching, for any resulting eigenvalue multiset the rearrangement inequality shows that J is maximized, with respect to the pairing of modes and task-relevant projection powers, when larger η_ρ are paired with larger $\overline{w}_\rho^2$, which is consistent with the transfer argument above. □

Lemma B.7. *For the alignment objective J, there exists a constant $\kappa > 0$ such that $J = \kappa\langle S_{\mathrm{model}}, S_{\mathrm{task}}\rangle$. Therefore,*

$$\mathcal{L}_{\mathrm{spec}} = 2 - \frac{2J}{\kappa \|S_{\mathrm{model}}\|_2 \|S_{\mathrm{task}}\|_2}.$$

Hence minimizing $\mathcal{L}_{\mathrm{spec}}$ is exactly equivalent to maximizing the normalized alignment objective $\frac{J}{\|S_{\mathrm{model}}\|_2\|S_{\mathrm{task}}\|_2}$. If $\|S_{\mathrm{model}}\|_2$ is fixed, this is also equivalent to maximizing J.

Proof of Lemma B.7. Expanding the general alignment objective $J = \frac{1}{L}\|\boldsymbol{T}_h \mathbf{C}_{\boldsymbol{uY}}\|_F^2$ as the sum of squared Euclidean norms, $J = \frac{1}{L}\sum_{j=1}^{d}\|\boldsymbol{T}_h \mathbf{C}_j\|_2^2$ (recall that d is the dimension of labels). By Parseval's Theorem, the norm of a vector is proportional to the norm of its Fourier transform. Thus,

$$\|\boldsymbol{T}_h \mathbf{C}_j\|_2^2 \propto \sum_\omega |\mathcal{F}(\boldsymbol{T}_h \mathbf{C}_j)(\omega)|^2 = \sum_\omega |\mathcal{F}(h * \mathbf{C}_j)|^2 = \sum_\omega |H(\omega)|^2 \left|\mathcal{F}\left[\mathbf{C}_{\boldsymbol{uY}}^{(:,j)}\right](\omega)\right|^2,$$

where the last step follows from Convolution Theorem. Substituting this back into the expression for J, we arrive at the frequency-domain representation,

$$J \propto \sum_\omega |H(\omega)|^2 \sum_{j=1}^{d} \left|\mathcal{F}\left[\mathbf{C}_{\boldsymbol{uY}}^{(:,j)}\right](\omega)\right|^2 \propto \sum_\omega S_{\mathrm{model}}(\omega) S_{\mathrm{task}}(\omega) = \langle S_{\mathrm{model}}, S_{\mathrm{task}}\rangle.$$

This equation shows that J is proportional to the inner product between the model's power spectrum and the task's total power spectrum. Combining this relation with Definition 2,

$$\mathcal{L}_{\mathrm{spec}} = \left\| \frac{S_{\mathrm{model}}}{\|S_{\mathrm{model}}\|_2} - \frac{S_{\mathrm{task}}}{\|S_{\mathrm{task}}\|_2} \right\|_2^2 = 2 - 2\frac{\langle S_{\mathrm{model}}, S_{\mathrm{task}}\rangle}{\|S_{\mathrm{model}}\|_2\|S_{\mathrm{task}}\|_2} = 2 - \frac{2J}{\kappa\|S_{\mathrm{model}}\|_2\|S_{\mathrm{task}}\|_2}.$$

This proves the result. □

Based on Lemma B.6, Lemma B.7, and the kernel generalization theory discussed in Section 2.3, we derive Theorem 3.3.

Theorem 3.3 (Spectral Matching Promotes Task-Model Spectral Alignment and Generalization)**.** *Minimizing spectral matching loss $\mathcal{L}_{\mathrm{spec}}$ is equivalent to maximizing the normalized alignment objective $J/(\|S_{\mathrm{model}}\|_2\|S_{\mathrm{task}}\|_2)$ (Lemma B.7), which concentrates task-relevant projection power toward higher-eigenvalue kernel modes and thereby yields a faster-rising cumulative-power curve $C(\rho)$ (Lemma B.6). Consequently, the resulting spectral alignment improves finite-sample generalization.*

B.4 Parameter-Level Analysis of SSM Frequency Bias

In this section, we will analyze the relation between frequency bias of an SSM and its parameters.

First, we restate the definition of frequency bias in Yu et al. [49]: frequency bias of an SSM means that the frequency responses of the systems have more variation in the low-frequency area than the high-frequency area.

The frequency response $H(\omega)$ of an SSM is actually the discrete-time Fourier transform of its impulse response h,

$$H(\omega) = \mathrm{DTFT}(h) = \sum_{n=0}^{\infty} h_n e^{-i\omega n}, \quad \omega \in (0, 2\pi).$$

Substituting $h_n = \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^n\overline{\boldsymbol{B}}$,

$$H(\omega) = \sum_{n=0}^{\infty} \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^n\overline{\boldsymbol{B}} e^{-i\omega n} = \overline{\boldsymbol{C}}(\boldsymbol{I} - \overline{\boldsymbol{A}}e^{-i\omega})^{-1}\overline{\boldsymbol{B}}.$$

We substitute $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \Delta$ into $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}$, and set $\boldsymbol{B} = \boldsymbol{1} \in \mathbb{R}^{n\times 1}$ (fixed $\boldsymbol{B}$ in S4D training), we derive

$$H(\omega) = \sum_k \boldsymbol{C}_k \left(e^{\Delta \boldsymbol{A}_k} - 1\right) \frac{1}{\boldsymbol{A}_k} \frac{1}{1 - e^{-i\omega + \Delta \boldsymbol{A}_k}},$$

where $\boldsymbol{A}_k$ denotes the k-th diagonal element of $\boldsymbol{A}$ and $\boldsymbol{C}_k$ denotes the k-th element of $\boldsymbol{C}$. This result is similar to equation (3) in Yu et al. [49]. $|H(\omega)|$ attains an extremum when the denominator reaches its extremum, which is given by

$$\frac{d}{d\omega} \operatorname{Re} e^{-i\omega + \Delta(\operatorname{Re}\boldsymbol{A}_k + i \operatorname{Im}\boldsymbol{A}_k)} = 0.$$

Thus, $\omega = \Delta \operatorname{Im} \boldsymbol{A}_k$ gives the location of the local extremum of $|H(\omega)|$. Additionally, the corresponding $\boldsymbol{C}_k$ and $\operatorname{Re}\boldsymbol{A}_k$ determine the magnitude at $\omega_k = \Delta \operatorname{Im} \boldsymbol{A}_k$.

Theorem B.8. *Let $|H(\omega)|$ be the frequency response of an S4D-like SSM with parameters $\boldsymbol{A} \in \mathbb{C}^{n\times n}, \boldsymbol{C} \in \mathbb{C}^{1\times n}, \Delta \in \mathbb{R}$. Then, the location of its extremum is determined by $\omega_k = \Delta \operatorname{Im} \boldsymbol{A}_k$, whereas the magnitude of the extremum is determined by $\operatorname{Re}\boldsymbol{A}$ and $\boldsymbol{C}$.*

This result can successfully explain the frequency bias of the SSM. We now provide an estimate for S4D. Since the original S4D initialization gives

$$\boldsymbol{A}_k = -0.5 + i\pi \cdot k, \quad k = 1, \cdots, \lfloor n/2 \rfloor,$$

the extremum points are then determined by $\omega_k = \pi\Delta k$. Typically, the number of states, n, is 64, and Δ is sampled from a log-uniform distribution between 0.001 and 0.1. Thus, the proportion of the largest ω_k relative to π, $\omega_n/\pi = \Delta \cdot \lfloor n/2 \rfloor$ satisfies a log-uniform distribution between 0.032 and 3.2. Thus, $P(\omega_n/\pi < 0.1) = 0.25, P(\omega_n/\pi < 0.3) = 0.49, P(\omega_n/\pi < 0.6) = 0.64$. This result shows that with the initialization of S4D, the informative frequency response, i.e., frequency area with more variation, has a high probability to concentrate in a narrow, low-frequency area, causing a low frequency bias of SSM. The first approach to tuning the frequency bias in Yu et al. [49], involving the simultaneous multiplication of all frequencies by a coefficient, is equivalent to adjusting the range of Δ.

C Task Spectrum Estimation by Fisher Log-Power

For a sequence-to-class dataset $\mathcal{D} = \{(\boldsymbol{u}^\mu, \boldsymbol{y}^\mu)\}_{\mu=1}^P$, where $\boldsymbol{y}^\mu \in \{0, 1\}^d$ is the one-hot label and $\boldsymbol{u}^\mu$ has sequence length L, we define

$$\widehat{\boldsymbol{u}}^\mu(\omega_k) = \mathrm{rFFT}(\boldsymbol{u}^\mu)_k, \qquad k = 0, \ldots, \lfloor L/2 \rfloor.$$

The Fisher log-power method forms

$$z^\mu(\omega_k) = \log\left(\left|\widehat{\boldsymbol{u}}^\mu(\omega_k)\right|^2 + \epsilon\right).$$

For class j, let

$$\pi_j = \mathbb{E}[y_j], \qquad \mu_j(\omega_k) = \mathbb{E}[z(\omega_k) \mid y_j = 1]\,,$$

and

$$\sigma_j^2(\omega_k) = \mathrm{Var}(z(\omega_k) \mid y_j = 1)\,.$$

The class-prior-weighted mean is

$$\bar{\mu}(\omega_k) = \sum_{j=1}^{d} \pi_j \mu_j(\omega_k).$$

The Fisher log-power spectrum used in the experiments is

$$S_{\mathrm{FLP}}(\omega_k) = \frac{\sum_{j=1}^{d} \pi_j \left(\mu_j(\omega_k) - \bar{\mu}(\omega_k)\right)^2}{\sum_{j=1}^{d} \pi_j \sigma_j^2(\omega_k) + \lambda},$$

where $\epsilon > 0$ stabilizes the logarithm and $\lambda > 0$ is a small regularizer. In the implementation, $\epsilon = 10^{-6}$ and $\lambda = 10^{-4}$ by default.

Algorithm 1 Fisher log-power task spectrum

Require: Dataset $\mathcal{D} = \{(\boldsymbol{u}^\mu, \boldsymbol{y}^\mu)\}_{\mu=1}^{P}$, number of classes d, stabilizer ϵ, Fisher regularizer λ

Initialize class counts $N_j \leftarrow 0$
Initialize accumulators $A_j(\omega) \leftarrow 0$ and $B_j(\omega) \leftarrow 0$
for each minibatch $(\boldsymbol{u}, \boldsymbol{y})$ **do**
  Compute $\widehat{\boldsymbol{u}}(\omega) = \mathrm{rFFT}(\boldsymbol{u})$
  Compute $z(\omega) = \log(|\widehat{\boldsymbol{u}}(\omega)|^2 + \epsilon)$
  for each class $j = 1, \ldots, d$ **do**
    Update $N_j \leftarrow N_j + \sum_\mu y_j^\mu$
    Accumulate $A_j(\omega) \leftarrow A_j(\omega) + \sum_\mu y_j^\mu z^\mu(\omega)$
    Accumulate $B_j(\omega) \leftarrow B_j(\omega) + \sum_\mu y_j^\mu z^\mu(\omega)^2$
  end for
end for
$\pi_j \leftarrow N_j / \sum_{j'} N_{j'}$
$\mu_j(\omega) \leftarrow A_j(\omega)/N_j$
$\sigma_j^2(\omega) \leftarrow B_j(\omega)/N_j - \mu_j(\omega)^2$
$\bar{\mu}(\omega) \leftarrow \sum_j \pi_j \mu_j(\omega)$
$B_{\mathrm{between}}(\omega) \leftarrow \sum_j \pi_j (\mu_j(\omega) - \bar{\mu}(\omega))^2$
$B_{\mathrm{within}}(\omega) \leftarrow \sum_j \pi_j \sigma_j^2(\omega)$
return the Fisher log-power spectrum $S_{\mathrm{FLP}}(\omega) = \frac{B_{\mathrm{between}}(\omega)}{B_{\mathrm{within}}(\omega)+\lambda}$.

Connection to the cross spectrum. The theoretical analysis uses the cross spectrum based on the centered input-label cross-covariance. For each class j, define

$$\mathbf{C}_{\boldsymbol{uy},j} \equiv \mathbf{C}_{\boldsymbol{uY}}^{(:,j)} = \mathbb{E}\left[(\boldsymbol{u} - \bar{\boldsymbol{u}})\,(y_j - \pi_j)\right], \qquad \pi_j = \mathbb{P}(y_j = 1).$$

Since y_j is one-hot, this can be written as

$$\mathbf{C}_{\boldsymbol{uy},j} = \pi_j \left(\mathbb{E}[\boldsymbol{u} \mid y_j = 1] - \mathbb{E}[\boldsymbol{u}]\right).$$

Thus the cross-covariance measures class-dependent deviations of the input signal from its marginal mean. The corresponding cross spectrum is

$$S_{\mathrm{cross}}(\omega) = \frac{1}{d} \sum_{j=1}^{d} \left|\mathcal{F}[\mathbf{C}_{\boldsymbol{uy},j}](\omega)\right|^2.$$

Empirically, this is estimated by

$$\widehat{S}_{\mathrm{cross}}(\omega) = \frac{1}{d} \sum_{j=1}^{d} \left| \mathrm{rFFT}\left(\frac{1}{P} \sum_{\mu=1}^{P} (\boldsymbol{u}^\mu - \bar{\boldsymbol{u}})(y_j^\mu - \pi_j) \right)(\omega) \right|^2.$$

Therefore, S_{cross} measures frequencies at which the input signal has class-dependent mean structure.

Fisher log-power follows the same label-conditioned spectral principle, but uses the phase-invariant log-power statistic

$$\phi_\omega(\boldsymbol{u}) = \log\left(|\widehat{\boldsymbol{u}}(\omega)|^2 + \epsilon\right)$$

instead of the signed Fourier coefficient of $\boldsymbol{u}$. Its between-class numerator is

$$B_{\text{FLP}}(\omega) = \sum_{j=1}^{d} \pi_j \left(\mathbb{E}[\phi_\omega(\boldsymbol{u}) \mid y_j = 1] - \mathbb{E}[\phi_\omega(\boldsymbol{u})]\right)^2.$$

Thus, S_{cross} and Fisher log-power are both frequency-wise label-dependence measures. The cross spectrum measures phase-sensitive linear dependence between the input signal and the label, whereas Fisher log-power measures class separability of spectral-energy features.

They share several useful properties. Both are nonnegative spectra over frequencies. Both assign low importance to frequencies whose relevant spectral statistic has weak dependence on the label, even if the marginal spectral energy at those frequencies is large. Both are estimated from empirical moments involving the one-hot labels and can be normalized into an importance distribution over frequencies.

The additional Fisher normalization is

$$S_{\text{FLP}}(\omega) = \frac{B_{\text{FLP}}(\omega)}{W_{\text{FLP}}(\omega) + \lambda}, \qquad W_{\text{FLP}}(\omega) = \sum_{j=1}^{d} \pi_j \operatorname{Var}(\phi_\omega(\boldsymbol{u}) \mid y_j = 1).$$

Hence Fisher log-power can be viewed as a noise-normalized and phase-invariant surrogate of the cross-spectrum criterion. It preserves the key idea used in the theory, namely selecting frequencies whose statistics are predictive of the one-hot label, while discounting frequencies with large within-class variability. We do not claim that S_{FLP} is equivalent to S_{cross}; rather, it is a practical surrogate tailored to classification tasks where discriminative information is often carried by spectral energy rather than signed phase-sensitive coefficients.

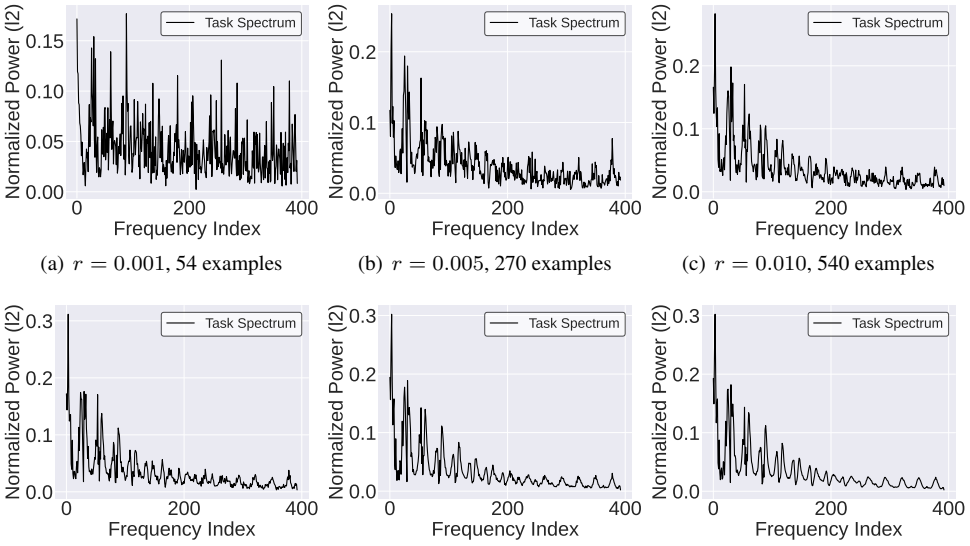
Stability of Fisher log-power under subsampling. Because TDI relies on estimating a task-relevant spectrum from finite training data, we examine whether the Fisher log-power statistic is stable under different subsampling ratios. Using sequential MNIST as a representative classification task, we compute the Fisher log-power spectrum from training subsets with ratios $r \in \{0.001, 0.005, 0.010, 0.020, 0.100, 1.000\}$ (Figure C.1), while keeping the preprocessing and frequency grid fixed. Empirically, the Fisher log-power spectrum becomes relatively stable once the number of examples reaches the order of 10^2: the dominant peaks remain consistent, while additional samples mainly reduce small fluctuations rather than changing the main task-relevant frequency bands. This suggests that Fisher log-power provides a reasonably stable estimate of task-relevant spectral structure even from limited labeled data.

We emphasize that this diagnostic does not imply perfect spectrum estimation in all low-data settings; rather, it supports the use of Fisher log-power as a practical classification-oriented surrogate for S_{task}.

Bootstrap stability diagnostic. We complement the qualitative spectrum comparison above with a quantitative diagnostic of Fisher log-power estimation stability. The diagnostic requires labeled training data but no model training.

We evaluate a grid of candidate sample sizes. At each size, we draw five independent class-stratified subsets. For each subset, we compute its Fisher log-power spectrum and generate 50 stratified bootstrap resamples, recomputing the spectrum for each resample with the same preprocessing and frequency grid. We then calculate the correlation between each bootstrap spectrum and the original subset spectrum. A subset is considered stable if the 2.5th percentile of these correlations is at least 0.95. The stability transition is the first evaluated sample size at which all five subsets satisfy this criterion and continue to satisfy it at every subsequent evaluated size. These transitions are coarse, grid-dependent estimates, rather than exact sample-size thresholds.

Table C.1 summarizes the observed transitions. The first stable training ratio varies substantially across datasets, so a common training-ratio threshold is inappropriate. PathFinder does not reach the



(a) $r = 0.001$, 54 examples (b) $r = 0.005$, 270 examples (c) $r = 0.010$, 540 examples

(d) $r = 0.020$, 1080 examples (e) $r = 0.100$, 5400 examples (f) $r = 1.000$, 54000 examples

Figure C.1: **Stability of the Fisher log-power task spectrum under subsampling.** We compute the Fisher log-power spectrum on sequential MNIST using six training-subset ratios while keeping the preprocessing and frequency grid fixed.The estimated spectra become relatively stable once the number of examples reaches the order of 10^2: the dominant spectral peaks remain largely consistent across larger subsets, while additional examples mainly reduce small fluctuations rather than changing the main task-relevant frequency bands.

stability criterion on the evaluated grid. For ListOps, numerical token IDs lack an intrinsic spectral interpretation, so we do not apply this diagnostic.

Table C.1: **Bootstrap stability of the Fisher log-power spectrum.** The first stable ratio satisfies the bootstrap criterion for all five subsets and all subsequent evaluated sizes. Ratios refer to fractions of each dataset's original training split. Stability does not imply a positive TDI accuracy gain.

Dataset	First stable ratio	Observation
Freq. Cls.	2.15%	Identifiable transition
sMNIST	2.15%	Identifiable transition
Speech Commands	4.64%	Identifiable transition
SC10	4.64%	Identifiable transition
CIFAR-10	21.5%	Negative TDI gains on both sides of the transition
PathFinder	Not reached	No transition identified on the evaluated grid
ListOps	Not applicable	Token IDs lack an intrinsic spectral interpretation

For each of the five datasets with an identifiable transition, we compare the matched TDI-minus-baseline accuracy gain at the grid point immediately before the transition with that at the grid point immediately after it. The gain decreases in all five cases, with the mean across datasets changing from $+9.00$ to -0.95 percentage points. This observation motivates a conservative empirical screen: beyond the dataset-specific stability transition, we would not apply TDI by default. Below or near the transition, TDI remains a candidate only if the estimated spectrum is informative and the default initialization is spectrally mismatched; instability alone is not a reason to apply TDI.

The diagnostic measures spectral-estimation stability, not initialization mismatch or the sign of the downstream gain. In particular, CIFAR-10 has negative TDI gains on both sides of its transition, demonstrating that stability is insufficient to ensure a benefit. For PathFinder and ListOps, we abstain from making a TDI applicability recommendation using this diagnostic. The observed transitions therefore should not be interpreted as guaranteed benefit–harm crossovers or universal measures of task difficulty.

Table D.2: **TDI variants used in different experiments.**

Experiment	SSM parameterization	TDI variant
Kernel regression	Complex diagonal SSM	Peak initialization + gradient refinement (Algorithm 3)
One-layer RNN	Real-valued matrix SSM	Peak initialization + gradient refinement (Algorithm 4)
Deep SSM	Complex diagonal SSM	Direct construction (Algorithm 3)

Algorithm 2 Unified Task-Dependent Initialization (TDI)

Require: Training data $\mathcal{D}_{\mathrm{train}}$, parameterization $p \in \{\textsc{ComplexDiag}, \textsc{RealMatrix}\}$, TDI variant v, task-spectrum estimator $\mathrm{TaskSpectrum}$

Ensure: Initialized SSM parameters

1: Estimate and normalize the task spectrum:

$$S_{\mathrm{task}} \leftarrow \mathcal{N}(\mathrm{TaskSpectrum}(\mathcal{D}_{\mathrm{train}}))\,.$$

2: **if** $p = \textsc{ComplexDiag}$ **then**

3: Initialize the complex diagonal SSM using Algorithm 3 with S_{task} and variant $v \in \{\textsc{PeakGD}, \textsc{Construct}\}$.

4: **else if** $p = \textsc{RealMatrix}$ **then**

5: Initialize the real-valued SSM/RNN using Algorithm 4 with S_{task} and variant $v = \textsc{PeakGD}$.

6: **end if**

7: Train the model with the standard supervised objective and the same downstream training protocol as the baseline.

Variable-length sequences. Our theoretical analysis assumes a fixed sequence length L. For variable-length numerical sequences with a common sampling interval, task spectra could be estimated on each sequence's native frequency grid and interpolated onto a shared frequency grid before aggregation, using consistent spectral normalization across lengths and excluding padded positions. This provides a possible extension of the spectrum estimator; the fixed-length kernel analysis does not directly establish guarantees for variable-length inputs.

D Task-Dependent Initialization Algorithms

TDI follows the same high-level principle across all experiments: estimate a task-relevant spectrum S_{task}, initialize the SSM so that its model spectrum S_{model} is aligned with S_{task}, and then train the model with the standard supervised objective.

The implementation differs only in the SSM parameterization and in whether the spectrum-aware initialization is further refined by gradient descent. A unified TDI algorithm is summarized in Algorithm 2. We use three TDI variants in our experiments, summarized in Table D.2. Kernel regression experiments use a complex diagonal SSM with peak initialization followed by gradient refinement. The one-layer RNN experiments use a real-valued matrix SSM, where peak initialization is implemented by real resonant blocks followed by gradient refinement. Deep SSM experiments use a complex diagonal SSM with direct construction, applied to the first SSM layer for efficiency. All variants share the same spectral matching objective; they differ only in how task-relevant frequencies are converted into valid SSM parameters. It is worth noting that TDI is a one-time initialization step and introduces no inference-time overhead; in deep SSM experiments we use the constructive peak initialization without gradient refinement.

Throughout the algorithms, selected frequencies f_j are represented as normalized rFFT frequencies in cycles per input step, i.e., $f_j \in [0, 1/2]$. If raw FFT bin indices b_j are selected, we convert them by $f_j = b_j/L$.

Algorithm 3 TDI for Complex Diagonal SSMs

Require: Task spectrum S_{task}, default parameters $(\Delta_0, \boldsymbol{A}_0, \boldsymbol{B}_0, \boldsymbol{C}_0)$, number of heads H, state size N, sequence length L, variant $v \in \{\text{PEAKGD}, \text{CONSTRUCT}\}$, refinement steps T
Ensure: Complex diagonal SSM parameters $\{(\Delta^{(h)}, \boldsymbol{A}^{(h)}, \boldsymbol{B}^{(h)}, \boldsymbol{C}^{(h)})\}_{h=1}^{H}$
Normalize $S_{\text{task}} \leftarrow \mathcal{N}(S_{\text{task}})$.
Select dominant frequencies $\{f_j\}$ and strengths $\{r_j\}$ from S_{task}.
for $h = 1$ to H **do**
  Initialize $(\Delta, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ from the default SSM initialization $(\Delta_0, \boldsymbol{A}_0, \boldsymbol{B}_0, \boldsymbol{C}_0)$.
  Peak initialization.
  Assign each state j a task frequency f_j and strength r_j.
  Set $\Delta \leftarrow \Delta_0$.
  Set damping $\gamma_j \leftarrow \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})(1 - r_j)$.
  Set complex diagonal modes $A_j \leftarrow -\gamma_j - i\, 2\pi f_j / \Delta$.
  Set gains $B_j, C_j \leftarrow g_{\min} + (g_{\max} - g_{\min}) r_j$.
  if $v = \text{PEAKGD}$ **then**
    Gradient refinement.
    Reparameterize $\Delta = \exp(\theta_\Delta)$ and $\boldsymbol{A} = -\exp(\theta_\Re) - i\theta_\Im$.
    Initialize Adam on $(\theta_\Re, \theta_\Im, \boldsymbol{C}, \theta_\Delta)$; keep $\boldsymbol{B}$ fixed.
    for $t = 1$ to T **do**
      Compute impulse response $\boldsymbol{k} \leftarrow \text{S4DImpulseResponse}(\Delta, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, L)$.
      Compute normalized model spectrum $S_{\text{model}} \leftarrow \mathcal{N}\big(|\,\text{FFT}(\boldsymbol{k})|^2\big)$.
      Compute spectral matching loss $\mathcal{L}_{\text{spec}} = \|S_{\text{model}} - S_{\text{task}}\|_2^2$.
      Update $(\theta_\Re, \theta_\Im, \boldsymbol{C}, \theta_\Delta)$ by one Adam step.
    end for
    Recover Δ, $\boldsymbol{A}$, and $\boldsymbol{C}$ from the optimized parameters.
  else if $v = \text{CONSTRUCT}$ **then**
    Use the peak-initialized parameters directly without gradient refinement.
  end if
  Store $(\Delta^{(h)}, \boldsymbol{A}^{(h)}, \boldsymbol{B}^{(h)}, \boldsymbol{C}^{(h)}) \leftarrow (\Delta, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$.
end for
Stack head-wise parameters into SSM tensors.
return Complex diagonal SSM parameters.

E Additional Experimental Details

E.1 Empirical Validation of Kernel Eigen-Decomposition

Dataset and preprocessing. We use 15,000 randomly selected MNIST images [18] to empirically validate the eigendecomposition of the SSM-induced kernel. Since Theorem 3.1 is stated in whitened input coordinates, we apply whitening before constructing the empirical kernel. After whitening, the effective dimensionality of the data is reduced, and we keep the top 550 dimensions in our main experiment. This preprocessing allows us to isolate the spectral bias induced by the SSM convolution operator from correlations in the input distribution.

Empirical kernel eigendecomposition. To obtain empirical eigenvalues and eigenfunctions, we diagonalize the Gram matrix computed on the full evaluation set. Specifically, for samples $\{\boldsymbol{u}^\mu\}_{\mu=1}^{M}$, we construct

$$\mathbf{K}_{\mu\nu} = K_{\text{SSM}}(\boldsymbol{u}^\mu, \boldsymbol{u}^\nu), \qquad \mu, \nu = 1, \ldots, M,$$

where M denotes the number of samples used to estimate the empirical kernel operator. We then diagonalize

$$\mathbf{K} = M \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^\top,$$

where the diagonal entries of $\boldsymbol{\Lambda}$ are the empirical kernel eigenvalues and the columns of $\boldsymbol{\Phi}$ are the corresponding empirical eigenvectors on the sampled data. The factor M ensures consistency with the empirical data distribution $p_M(\boldsymbol{u}) = \frac{1}{M} \sum_{\mu=1}^{M} \delta(\boldsymbol{u} - \boldsymbol{u}^\mu)$. Under this empirical distribution, the

Algorithm 4 TDI for Real-Valued SSMs/RNNs

Require: Task spectrum S_{task}, initial parameters $(\Delta_0, \boldsymbol{A}_0, \boldsymbol{B}_0)$, state size N, sequence length L, refinement steps K
Ensure: Initialized parameters $(\Delta, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$
Normalize $S_{\text{task}} \leftarrow \mathcal{N}(S_{\text{task}})$.
Set output vector $\boldsymbol{C} \leftarrow \mathbf{1}_{1\times N}$.
Peak initialization.
Initialize $\Delta \leftarrow \Delta_0$, $\boldsymbol{A} \leftarrow -0.5\mathbf{I}$, $\boldsymbol{B} \leftarrow \mathbf{0}$.
Select top-M frequency bins $\{f_j\}_{j=1}^{M}$ from S_{task}, where $M = \min(\lfloor N/2 \rfloor, |S_{\text{task}}|)$.
Save $\boldsymbol{B}_{\text{init}} \leftarrow \boldsymbol{B}_0$.
for $j = 1$ to M **do**
  Set angular frequency $\omega_j \leftarrow 2\pi f_j/\Delta$.
  Set damping $\alpha_j \leftarrow \begin{cases} 0.1, & j \le 6, \\ 0.5, & j > 6. \end{cases}$
  Insert the real resonant block $\begin{bmatrix} -\alpha_j & \omega_j \\ -\omega_j & -\alpha_j \end{bmatrix}$ into rows/columns $2j-1 : 2j$ of $\boldsymbol{A}$.
  Set $B_{2j-1} \leftarrow B_{\text{init},2j-1}$.
end for
Gradient refinement.
Initialize Adam on $(\Delta, \boldsymbol{A}, \boldsymbol{B})$.
for $i = 1$ to K **do**
  Discretize by ZOH: $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}) \leftarrow \mathrm{ZOH}(\Delta, \boldsymbol{A}, \boldsymbol{B})$.
  Compute impulse response $\boldsymbol{k} \leftarrow \mathrm{ImpulseResponse}(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \boldsymbol{C}, L)$.
  Compute normalized model spectrum $S_{\text{model}} \leftarrow \mathcal{N}\big(|\,\mathrm{FFT}(\boldsymbol{k})|^2\big)$.
  Compute spectral matching loss $\mathcal{L}_{\text{spec}} = \|S_{\text{model}} - S_{\text{task}}\|_2^2$.
  Update $(\Delta, \boldsymbol{A}, \boldsymbol{B})$ by one Adam step.
end for
return $(\Delta, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$.

empirical eigenfunctions are represented by $\sqrt{M}\boldsymbol{\Phi}$, since

$$\frac{1}{M}\sum_{\mu=1}^{M}\left(\sqrt{M}\boldsymbol{\Phi}_{\mu i}\right)\left(\sqrt{M}\boldsymbol{\Phi}_{\mu j}\right) = \delta_{ij}.$$

Thus, the Gram-matrix eigendecomposition provides empirical counterparts to the eigenvalues and eigenfunctions predicted by Theorem 3.1.

Additional results on unwhitened inputs. In addition to the whitened setting used in the main text, we also evaluate the same diagnostics on unwhitened MNIST inputs. The unwhitened setting no longer isolates the SSM operator alone: the effective feature operator becomes $\boldsymbol{T}_h\Sigma^{1/2}$, so the resulting spectrum reflects both the SSM convolution and the input covariance structure. Nevertheless, the empirical kernel eigendecomposition remains consistent with the corresponding effective operator.

Figure E.2 reports the eigenvalue spectrum, eigenfunction agreement, squared target weights, cumulative power, and kernel regression generalization on unwhitened inputs. Compared with the whitened setting, the cumulative power $C(\rho)$ often rises faster because the input covariance concentrates task-relevant variation into fewer effective directions. This can lead to better empirical generalization, but it also mixes the inductive bias of the SSM with the structure of the input distribution. Therefore, we use whitened inputs in the main text to cleanly validate Theorem 3.1.

Finite-sample rank and empirical tail. Theorem 3.1 states that the SSM-induced kernel has rank at most L, because the feature map $\Phi_{\text{SSM}}(\boldsymbol{u}) = \boldsymbol{T}_h\boldsymbol{u}$ lies in an L-dimensional feature space. In the empirical Gram-matrix eigendecomposition, however, eigenvalues beyond the theoretical rank are not exactly zero. This small tail arises from finite-sample estimation, numerical precision, and preprocessing operations such as whitening and dimensionality reduction.

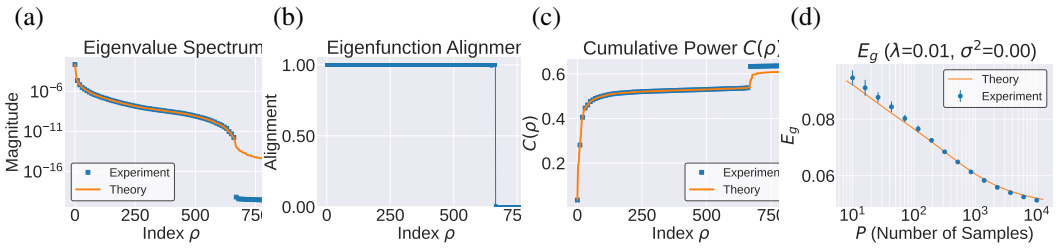


Figure E.2: Sanity check of the SSM-induced kernel eigendecomposition. (a) The empirical eigenvalue spectrum matches the theoretical prediction $\eta_\rho = s_\rho^2$ obtained from the singular values of the Toeplitz convolution operator. (b) The empirical and predicted eigenfunctions are well aligned across most modes. (c) The predicted eigenbasis accurately captures the cumulative task power $C(\rho)$. (d) Kernel regression generalization predicted from the eigenspectrum matches empirical KRR performance across sample sizes. Error bars indicate standard deviations over 50 independent runs.

Although the tail eigenvalues are small, their cumulative contribution to the empirical target expansion is not negligible. In the kernel generalization prediction, the leading L modes are described by the theoretical SSM-induced eigenspectrum, while the residual empirical tail behaves like an additional weak-noise component. Concretely, in Figure 2(d), the theoretical curve is computed from the predicted leading spectrum together with a constant offset accounting for the cumulative contribution of the empirical tail. This offset is non-trivial even though each individual tail eigenvalue is small, because it aggregates over many weak directions. This empirical tail does not contradict Theorem 3.1; rather, it reflects the difference between the ideal finite-dimensional kernel operator and its finite-sample numerical estimate. Similar finite-sample spectral tails and their effect on kernel learning curves have also been observed in prior kernel eigenlearning analyses [11, 12, 42].

E.2 Kernel Regression Experimental Details

Synthetic task generation. For the synthetic low- and high-frequency tasks, we generate labels based on MNIST images as follows. Let $\mathbf{U} \in \mathbb{R}^{P\times L}$ denote the mean-centered MNIST data matrix, with each image reshaped into a vector. We define a temporal grid $t \in [0,1]^L$ and construct frequency patterns corresponding to low- or high-frequency components,

$$p_{\text{low}}(t) = \cos(2\pi \cdot 20t) + \cos(2\pi \cdot 40t),$$
$$p_{\text{high}}(t) = \cos(2\pi \cdot 300t) + \sin(2\pi \cdot 350t),$$

which are L_2-normalized and used to generate labels as $\mathbf{Y} = \mathbf{U}p$. This procedure ensures that the synthetic labels inherit the desired frequency structure while preserving the input correlations of MNIST.

Design of low- and high-frequency biased SSMs. In this experiment, the SSMs are parameterized as S4D [24], including $\boldsymbol{A} = -\exp\log(-\operatorname{Re}\boldsymbol{A}) + i\operatorname{Im}\boldsymbol{A}$, output matrix $\boldsymbol{C}$, and time step $\Delta = \exp\log\Delta$, where $\log(-\operatorname{Re}\boldsymbol{A})$, $\operatorname{Im}\boldsymbol{A}$, $\boldsymbol{C}$, and $\log\Delta$ are trainable parameters. The parameters of the low-frequency biased SSM are set by

$$\Delta = \frac{0.5}{N}, \quad \operatorname{Im}\boldsymbol{A} = \pi \cdot \{0, 1, \cdots, \lfloor n/2 \rfloor\},$$

and those of the high-frequency biased SSM are

$$\Delta = \frac{1}{N}, \quad \operatorname{Im}\boldsymbol{A} = \pi \cdot \{\lfloor n/2 \rfloor, \cdots, n-1\},$$

where

$$\boldsymbol{C} \sim \mathcal{N}(0, \boldsymbol{I}), \quad \operatorname{Re}\boldsymbol{A} = \{-0.5, -0.5, \cdots, -0.5\}$$

are fixed.

Learning curves under more parameter settings. We also provide additional figures for Section 4.2, including learning curves under more parameter settings, shown in Figure E.3 and Figure E.4

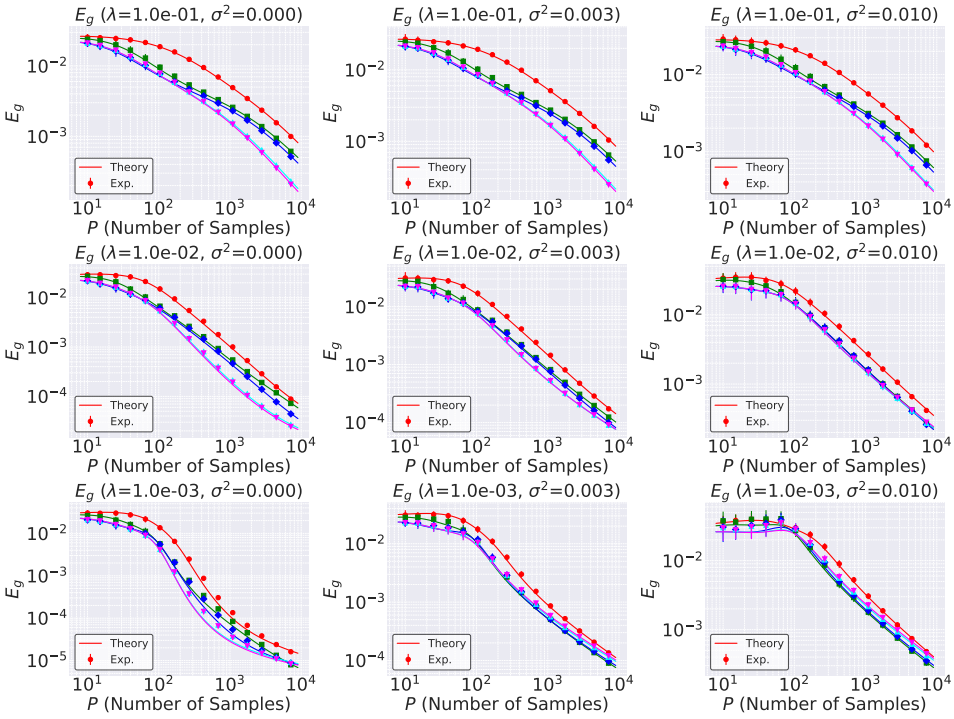


Figure E.3: Learning curves of *matched* case under more parameter settings.

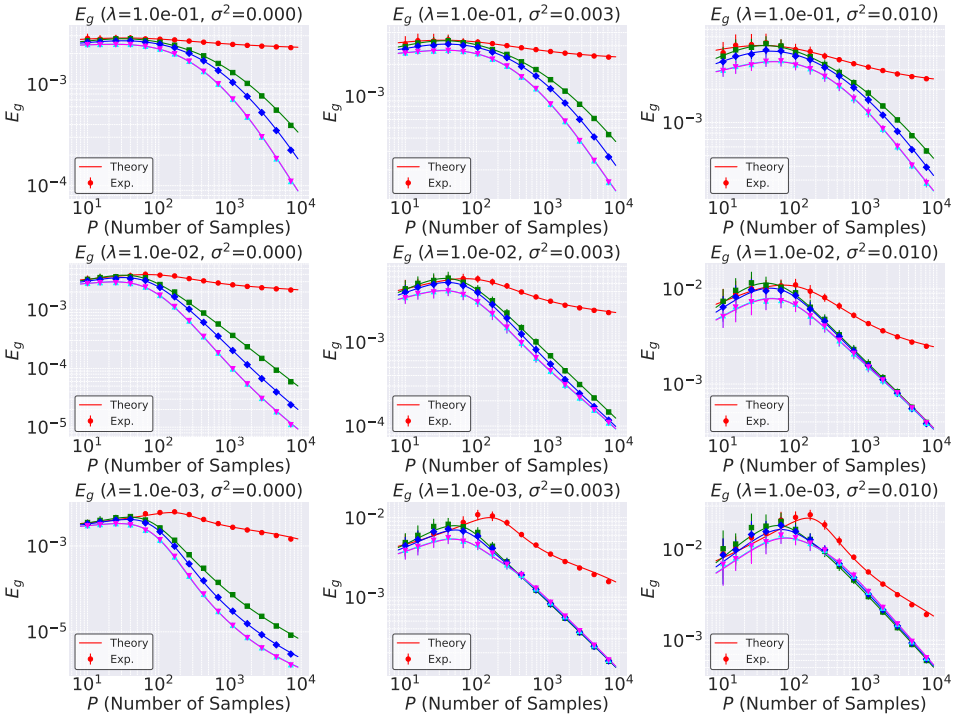


Figure E.4: Learning curves of *mismatched* case under more parameter settings.

Table E.3: **Dataset statistics used in the one-layer trainable SSM experiments.** The reported train/test counts correspond to the full dataset split before applying any training ratio.

Dataset	Train	Test	Classes	Seq. len.	Input dim.
Binary Freq.	40000	10000	2	256	1
ECG5000	500	4500	5	140	1
FordA	3601	1320	2	500	1
FordB	3636	810	2	500	1
MNIST	60000	10000	10	784	1
TwoPatterns	1000	4000	4	128	1
Wafer	1000	6164	2	152	1

E.3 Trainable One-Layer SSM Experimental Details

This appendix summarizes the experimental setup used for the Standard RNN, the one-layer SSM baseline, and TDI. All methods were evaluated under the same data splits, training ratios, random seeds, and optimization protocol unless noted otherwise. The complete final accuracy and early-epoch summary table for Standard RNN, the SSM baseline, and TDI is provided in Appendix F.1.

Model Structures.

Standard RNN. The Standard RNN branch uses a single-layer recurrent encoder followed by a two-layer MLP readout:

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1}), \qquad \hat{y} = \mathrm{MLP}(h_T). \tag{4}$$

In the code, this is implemented with `nn.RNN` and a readout of the form `Linear(128` $\rightarrow$ `64) + ReLU + Linear(64` $\rightarrow$ `#classes)`. The hidden size is 128 and the number of recurrent layers is 1.

Linear SSM baseline. The baseline uses the SSM implementation with a linear state update and a two-layer MLP readout:

$$h_t = Wh_{t-1} + W_{\mathrm{in}}x_t, \qquad \hat{y} = \mathrm{MLP}(h_T). \tag{5}$$

SSM is initialized from the HiPPO `foud` [22] parameterization, discretized with zero-order hold, and then trained end-to-end. In this setting, both `W` and `W_in` are trainable.

TDI. TDI uses the same architecture as the SSM baseline, but the state-space parameters are first adapted with task-dependent initialization before the supervised training stage. The implementation keeps the same hidden size, readout head, and optimization settings as the baseline. For details of TDI algorithm, see Appendix D.

Dataset Statistics. All inputs are treated as sequences with input dimension 1. For MNIST, the image is flattened row-wise into a length-784 sequence; the remaining datasets are already time-series or are generated as one-dimensional sequences. The details of these datasets are summarized in Table E.3.

Binary Frequency Classification. We construct a synthetic binary sequence classification dataset in which the label is determined solely by the amplitude of a designated frequency component. Each example is a length-256 univariate sequence. The target sinusoid has frequency 3 Hz and random phase, and its amplitude is sampled from a label-dependent interval to ensure class balance. Specifically, class 1 samples use amplitudes from $(0.6, 1.0)$, whereas class 0 samples use amplitudes from $(0.1, 0.4)$. We additionally superimpose distractor sinusoids at frequencies 1, 7, and 11 Hz with random amplitudes and phases, together with Gaussian noise ($\sigma = 0.1$). Each sequence is then normalized by its maximum absolute value.

Hyperparameters. We use random seeds $\{42, 123, 1024\}$ and training ratios $\{0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.00\}$ for all datasets. Other hyperparameters are summarized in Table E.4.

Table E.4: **Hyperparameters for the trainable one-layer SSM experiments.** The only dataset-dependent training hyperparameter is the number of epochs.

Dataset	Epochs
Binary Freq.	15
ECG5000	15
FordA	10
FordB	15
MNIST	15
TwoPatterns	15
Wafer	10

Shared setting	Value
Optimizer	AdamW
Learning rate	10^{-3}
Weight decay	10^{-2}
Gradient clipping	1.0
Batch size	128
Hidden size	128
Spectral radius	0.9
Scheduler	cosine annealing, $\eta_{\min} = 10^{-6}$

E.4 Deep SSM Experimental Details

This appendix summarizes the experimental setup used for the deep SSM baseline, TDI, and TDI-Frozen on Frequency Classification, sMNIST, pMNIST, PathFinder, CIFAR-10, ListOps, Speech Commands, and SC10. Unless otherwise noted, all methods were evaluated with the same data splits, training ratios, random seeds, optimizer, and learning-rate schedule.

Model Structures.

Deep SSM baseline. All deep SSM experiments use the S4D backbone with diagonal state-space parameterization, i.e., `diag-lin` [24]. For real-valued sequence inputs, the model first applies a linear input projection to dimension d_{model}, then processes the sequence with a stack of residual S4 blocks, and finally applies a sequence-classification decoder. For Frequency Classification, sMNIST, pMNIST, PathFinder, CIFAR-10, Speech Commands, and SC10, the decoder uses temporal mean pooling followed by a linear classifier. For ListOps, the pipeline uses an embedding encoder together with a length-aware last-token decoder inherited from the LRA text setup. In all experiments, only the first S4 layer is expanded to `n_ssm`$= d_{\text{model}}$, while the remaining layers keep the default setting.

TDI. TDI uses the same deep SSM backbone as the baseline, but adapts the first S4 layer with task-dependent initialization before supervised training. In the implementation, the task-spectrum configuration is injected only into the first S4 layer, while all later layers use the baseline initialization and are trained normally. All TDI runs use Fisher log-power as the task-spectrum estimator and Algorithm 3 for initialization. For numerical inputs, we estimate the task spectrum before the time-shared pointwise linear encoder, which mixes channels without mixing distinct temporal frequencies. For ListOps, we instead estimate the spectrum from numerical token IDs before embedding. This is a heuristic that depends on the arbitrary token numbering: token IDs have no intrinsic metric or spectral semantics, and their spectrum should not be interpreted as the spectrum of the learned embeddings.

TDI-Frozen. TDI-Frozen uses the same task-dependent initialization as TDI for the first S4 layer, but freezes the initialized first-layer S4 parameters during supervised training. All later layers, the decoder, and the remaining trainable parameters are trained normally. This provides a control for whether a fixed task-aligned spectral filter is sufficient, or whether allowing the TDI-initialized layer to adapt during downstream training is important.

Dataset Statistics. All tasks are treated as sequences. sMNIST is the standard sequential MNIST task obtained by flattening each image row-wise into a length-784 sequence. The pMNIST script uses the same 784-pixel sequence together with a fixed permutation implemented by the codebase. CIFAR-10 refers to grayscale sequential CIFAR-10: each 32×32 image is converted to grayscale and flattened into a length-1024 sequence. PathFinder uses 32×32 grayscale images and flattens them into length-1024 sequences. ListOps uses token sequences truncated to length 2048 with EOS appended. Speech Commands and SC10 use raw one-second audio waveforms of length 16000.

Frequency Classification. For Frequency Classification, we use the synthetic dataset where each sample is a length-1024 univariate signal. The class label determines a frequency band within the interval $[4, 64]$ Hz, which is partitioned into 10 bins. For each example, the principal sinusoid

Table E.5: **Dataset statistics used in the deep SSM experiments.** The reported train/validation/test counts correspond to the full split before applying any training subset ratio.

Dataset	Train	Val	Test	Classes	Seq. len.	Input dim.
Freq. Classification	10000	1000	1000	10	1024	1
sMNIST	54000	6000	10000	10	784	1
pMNIST	54000	6000	10000	10	784	1
PathFinder	480001	59999	59999	2	1024	1
CIFAR-10	45000	5000	10000	10	1024	1
ListOps	96000	2000	2000	10	2048	–
Speech Commands	84843	9981	11005	35	16000	1
SC10	24482	5246	5247	10	16000	1

Table E.6: **Hyperparameters used in the deep SSM experiments.** LR is learning rate and WD is weight decay. BN and LN refer to Batch Normalization and Layer Normalization, respectively; Pre-norm indicates whether normalization is applied before the S4 block.

Dataset	Depth	Features H	State Size N	Norm	Pre-norm	Dropout	LR	Batch Size	Epochs	WD	$(\Delta_{\min}, \Delta_{\max})$
Freq. Classification	2	64	64	BN	True	0	0.005	64	10	0.01	$(0.001, 0.1)$
sMNIST	4	256	64	LN	False	0.1	0.01	128	30	0.05	$(0.001, 0.1)$
pMNIST	4	256	64	LN	False	0.1	0.01	128	30	0.05	$(0.001, 0.1)$
CIFAR-10	4	256	64	LN	False	0.1	0.01	128	30	0.05	$(0.001, 0.1)$
Speech Commands	4	256	64	BN	False	0	0.01	16	30	0.05	$(0.001, 0.1)$
SC10	4	256	64	BN	False	0	0.01	16	30	0.05	$(0.001, 0.1)$
ListOps	4	256	64	BN	False	0	0.01	50	30	0.05	$(0.001, 0.1)$
PathFinder	4	256	64	BN	True	0	0.004	70	30	0.03	$(0.001, 0.1)$

frequency is sampled uniformly within the class-specific bin, its amplitude is sampled from $[0.8, 1.2]$, and its phase is random. The signal is further modulated by a low-frequency envelope with modulation depth 0.15 and modulation frequency sampled from $[0.25, 2.0]$ Hz. We additionally add a second harmonic with scale 0.3, a low-frequency drift term with scale 0.1, and Gaussian noise with standard deviation 0.15. Each sequence is finally normalized to zero mean and unit variance.

Hyperparameters. All runs use the five random seeds $\{0, 42, 218, 18, 2218\}$. The training ratios are logarithmically spaced between 10^{-3} and 1. PathFinder, ListOps, Speech Commands, and SC10 use

$$\{0.00100, 0.00215, 0.00464, 0.01000, 0.02154, 0.04642, 0.10000, 0.21544, 0.46416, 1.00000\},$$

Frequency Classification drops the smallest point and uses the remaining 9 ratios, and sMNIST, pMNIST, and CIFAR-10 use the last 8 ratios

$$\{0.00464, 0.01000, 0.02154, 0.04642, 0.10000, 0.21544, 0.46416, 1.00000\}.$$

All experiments use AdamW with cosine annealing and $\eta_{\min} = 10^{-6}$. No gradient clipping is applied. Other hyperparameters are summarized in Table E.6.

E.5 Compute Resources

Kernel regression and one-layer SSM experiments were run on NVIDIA RTX 4090 GPUs. Deep SSM experiments were run on NVIDIA RTX 4090 and RTX 5090 GPUs. Each RTX 4090/5090 GPU has 24GB/32GB memory. Each training job used a single GPU and corresponded to one dataset, one training ratio, one random seed, and one model variant. The one-layer SSM experiments evaluated 7 datasets, 8 training ratios, and 3 random seeds, while the deep SSM experiments evaluated 8 datasets, 10 training ratios, and 5 random seeds. Kernel regression runs take negligible compute time. Typical one-layer SSM runs take roughly 1–5 GPU-hours, while deep SSM runs take roughly 1–30 GPU-hours depending on dataset and training ratio. Across all experiments reported in the paper, including baseline, TDI, and TDI-Frozen, the total compute budget was approximately 400–500 GPU-hours. TDI adds only a task-spectrum estimation and initialization step before supervised training and does not change the architecture or inference cost.

E.6 Code Availability and Reproducibility

The code for reproducing our experiments is publicly available at `https://github.com/Criticality-Cognitive-Computation-Lab/Aligning-Inductive-Bias-for-State-Space-Models`. The repository contains the implementation of TDI, baseline training code, configuration files, and scripts for reproducing the main experiments reported in Section 4. The included README describes the software environment, dataset preparation steps, and example commands for running the kernel regression, one-layer SSM, and deep SSM experiments.

F Additional Experimental Results

F.1 Complete Results for Trainable One-Layer SSMs

In this section, we provide full per-dataset, per-ratio comparisons among the SSM baseline, TDI, and the Standard RNN baseline in Table F.7.

Table F.7: **Complete final performance for Standard RNN, the SSM baseline, and TDI.** The left block reports final test accuracy; the right block reports mean test accuracy over the first 5 epochs. Each cell reports mean ± standard deviation over 3 seeds. The best value within each block for each dataset-ratio row is bolded, and the second-best value is underlined.

		Final test accuracy			First-5 test accuracy		
Dataset	Ratio	Std. RNN	Baseline	TDI	Std. RNN	Baseline	TDI
Binary Freq.	0.01	$50.4 \pm 0.9\%$	$73.0 \pm 4.8\%$	$\mathbf{79.1 \pm 0.9\%}$	$50.2 \pm 0.7\%$	$55.1 \pm 3.5\%$	$\mathbf{58.6 \pm 7.3\%}$
Binary Freq.	0.02	$53.6 \pm 2.5\%$	$83.6 \pm 5.4\%$	$\mathbf{84.2 \pm 1.3\%}$	$50.8 \pm 0.8\%$	$\mathbf{55.9 \pm 3.9\%}$	$54.1 \pm 4.1\%$
Binary Freq.	0.04	$52.9 \pm 3.8\%$	$86.6 \pm 2.2\%$	$\mathbf{89.9 \pm 5.5\%}$	$51.0 \pm 1.0\%$	$65.2 \pm 12.1\%$	$\mathbf{70.9 \pm 4.4\%}$
Binary Freq.	0.08	$50.4 \pm 1.7\%$	$92.5 \pm 3.6\%$	$\mathbf{94.9 \pm 0.7\%}$	$51.2 \pm 0.9\%$	$70.7 \pm 3.5\%$	$\mathbf{79.1 \pm 2.1\%}$
Binary Freq.	0.16	$58.9 \pm 11.8\%$	$\mathbf{96.7 \pm 0.8\%}$	$95.0 \pm 2.2\%$	$51.8 \pm 2.1\%$	$77.6 \pm 4.6\%$	$\mathbf{78.5 \pm 5.9\%}$
Binary Freq.	0.32	$65.8 \pm 18.6\%$	$\mathbf{97.5 \pm 0.2\%}$	$97.0 \pm 0.8\%$	$53.2 \pm 1.1\%$	$87.5 \pm 0.8\%$	$\mathbf{87.6 \pm 2.1\%}$
Binary Freq.	0.64	$51.4 \pm 1.9\%$	$\mathbf{97.1 \pm 0.8\%}$	$\mathbf{97.1 \pm 0.4\%}$	$51.1 \pm 1.2\%$	$89.3 \pm 1.8\%$	$\mathbf{89.5 \pm 3.2\%}$
Binary Freq.	1.00	$50.8 \pm 1.5\%$	$\mathbf{97.8 \pm 0.0\%}$	$97.7 \pm 0.0\%$	$50.9 \pm 1.2\%$	$\mathbf{92.3 \pm 0.9\%}$	$91.5 \pm 1.4\%$
ECG5000	0.01	$\mathbf{55.1 \pm 22.9\%}$	$48.3 \pm 10.9\%$	$\mathbf{55.2 \pm 21.2\%}$	$32.2 \pm 1.5\%$	$21.5 \pm 13.7\%$	$\mathbf{42.4 \pm 9.2\%}$
ECG5000	0.02	$81.1 \pm 0.8\%$	$62.1 \pm 1.5\%$	$\mathbf{85.5 \pm 4.8\%}$	$\mathbf{38.7 \pm 8.2\%}$	$29.9 \pm 17.1\%$	$28.1 \pm 17.8\%$
ECG5000	0.04	$\mathbf{81.6 \pm 1.7\%}$	$65.1 \pm 17.7\%$	$77.5 \pm 8.4\%$	$42.3 \pm 9.0\%$	$31.6 \pm 16.3\%$	$\mathbf{50.4 \pm 11.6\%}$
ECG5000	0.08	$\mathbf{80.8 \pm 1.2\%}$	$74.4 \pm 2.8\%$	$76.6 \pm 4.3\%$	$\mathbf{41.7 \pm 8.3\%}$	$31.5 \pm 17.2\%$	$32.3 \pm 19.5\%$
ECG5000	0.16	$\mathbf{80.3 \pm 0.7\%}$	$64.8 \pm 3.3\%$	$69.2 \pm 5.3\%$	$\mathbf{42.1 \pm 9.5\%}$	$34.0 \pm 19.5\%$	$30.0 \pm 20.9\%$
ECG5000	0.32	$86.8 \pm 0.2\%$	$88.9 \pm 2.1\%$	$\mathbf{89.5 \pm 1.4\%}$	$\mathbf{62.1 \pm 3.6\%}$	$57.2 \pm 12.7\%$	$55.7 \pm 8.4\%$
ECG5000	0.64	$77.8 \pm 15.8\%$	$90.4 \pm 0.8\%$	$\mathbf{90.9 \pm 0.9\%}$	$\mathbf{63.6 \pm 2.4\%}$	$63.3 \pm 7.7\%$	$54.5 \pm 16.3\%$
ECG5000	1.00	$78.9 \pm 16.3\%$	$\mathbf{91.5 \pm 0.3\%}$	$91.1 \pm 0.3\%$	$59.4 \pm 12.3\%$	$67.9 \pm 7.1\%$	$\mathbf{75.6 \pm 7.2\%}$
FordA	0.01	$\mathbf{50.8 \pm 1.3\%}$	$50.5 \pm 3.1\%$	$50.0 \pm 2.5\%$	$\mathbf{51.2 \pm 0.5\%}$	$50.3 \pm 1.7\%$	$50.4 \pm 1.0\%$
FordA	0.02	$50.8 \pm 1.1\%$	$50.0 \pm 3.5\%$	$\mathbf{52.8 \pm 2.9\%}$	$51.3 \pm 0.4\%$	$\mathbf{51.4 \pm 2.7\%}$	$50.0 \pm 1.8\%$
FordA	0.04	$49.2 \pm 1.0\%$	$51.4 \pm 2.9\%$	$\mathbf{52.3 \pm 2.8\%}$	$50.4 \pm 0.8\%$	$51.0 \pm 2.4\%$	$\mathbf{51.1 \pm 0.6\%}$
FordA	0.08	$48.5 \pm 1.8\%$	$53.0 \pm 3.1\%$	$\mathbf{59.5 \pm 1.9\%}$	$50.1 \pm 0.8\%$	$51.6 \pm 2.8\%$	$\mathbf{51.8 \pm 0.8\%}$
FordA	0.16	$49.5 \pm 1.2\%$	$55.9 \pm 3.4\%$	$\mathbf{69.5 \pm 2.9\%}$	$50.0 \pm 0.5\%$	$51.5 \pm 1.6\%$	$\mathbf{57.6 \pm 2.4\%}$
FordA	0.32	$51.8 \pm 2.4\%$	$65.1 \pm 0.9\%$	$\mathbf{66.1 \pm 5.2\%}$	$51.3 \pm 1.5\%$	$55.0 \pm 1.3\%$	$\mathbf{58.3 \pm 3.3\%}$
FordA	0.64	$53.7 \pm 3.5\%$	$67.1 \pm 5.9\%$	$\mathbf{71.1 \pm 4.7\%}$	$51.7 \pm 1.3\%$	$57.8 \pm 1.0\%$	$\mathbf{62.3 \pm 7.5\%}$
FordA	1.00	$54.9 \pm 6.1\%$	$\mathbf{74.7 \pm 3.7\%}$	$73.9 \pm 3.0\%$	$52.1 \pm 1.0\%$	$\mathbf{61.4 \pm 6.0\%}$	$59.1 \pm 4.5\%$
FordB	0.01	$50.2 \pm 0.5\%$	$50.9 \pm 0.9\%$	$\mathbf{51.5 \pm 1.2\%}$	$49.6 \pm 0.3\%$	$\mathbf{50.7 \pm 0.2\%}$	$50.5 \pm 1.2\%$
FordB	0.02	$49.6 \pm 1.6\%$	$\mathbf{55.0 \pm 2.0\%}$	$52.4 \pm 1.8\%$	$50.0 \pm 1.1\%$	$\mathbf{52.3 \pm 2.0\%}$	$50.6 \pm 0.5\%$
FordB	0.04	$49.5 \pm 1.8\%$	$51.3 \pm 1.8\%$	$\mathbf{52.6 \pm 2.5\%}$	$50.0 \pm 1.0\%$	$51.3 \pm 0.9\%$	$\mathbf{52.1 \pm 1.0\%}$
FordB	0.08	$49.6 \pm 0.7\%$	$52.2 \pm 2.5\%$	$\mathbf{59.5 \pm 0.7\%}$	$49.9 \pm 0.5\%$	$49.8 \pm 0.4\%$	$\mathbf{50.6 \pm 0.6\%}$
FordB	0.16	$50.0 \pm 1.3\%$	$58.0 \pm 4.4\%$	$\mathbf{59.1 \pm 2.0\%}$	$50.0 \pm 0.5\%$	$51.1 \pm 0.9\%$	$\mathbf{52.5 \pm 2.3\%}$
FordB	0.32	$52.4 \pm 0.2\%$	$58.7 \pm 3.7\%$	$\mathbf{62.8 \pm 1.5\%}$	$50.2 \pm 0.8\%$	$\mathbf{53.2 \pm 1.4\%}$	$52.6 \pm 1.8\%$
FordB	0.64	$51.1 \pm 2.4\%$	$60.5 \pm 0.5\%$	$\mathbf{60.6 \pm 5.0\%}$	$50.4 \pm 0.7\%$	$\mathbf{56.4 \pm 1.1\%}$	$53.2 \pm 3.6\%$
FordB	1.00	$52.7 \pm 7.0\%$	$\mathbf{62.6 \pm 1.6\%}$	$61.7 \pm 1.0\%$	$49.6 \pm 0.9\%$	$\mathbf{57.1 \pm 1.4\%}$	$52.9 \pm 2.8\%$
MNIST	0.01	$10.3 \pm 0.9\%$	$42.8 \pm 2.3\%$	$\mathbf{60.5 \pm 4.9\%}$	$10.5 \pm 0.9\%$	$\mathbf{22.5 \pm 2.7\%}$	$18.4 \pm 3.3\%$
MNIST	0.02	$11.3 \pm 0.0\%$	$51.7 \pm 6.8\%$	$\mathbf{72.8 \pm 4.7\%}$	$10.9 \pm 0.7\%$	$\mathbf{29.0 \pm 4.8\%}$	$23.7 \pm 7.2\%$
MNIST	0.04	$10.8 \pm 1.0\%$	$64.2 \pm 3.8\%$	$\mathbf{81.1 \pm 2.0\%}$	$10.8 \pm 1.0\%$	$\mathbf{43.3 \pm 0.9\%}$	$36.7 \pm 3.8\%$
MNIST	0.08	$12.4 \pm 1.8\%$	$72.5 \pm 3.5\%$	$\mathbf{88.4 \pm 1.2\%}$	$11.3 \pm 0.1\%$	$\mathbf{51.1 \pm 0.6\%}$	$49.9 \pm 11.0\%$
MNIST	0.16	$11.3 \pm 0.0\%$	$74.2 \pm 7.0\%$	$\mathbf{89.2 \pm 2.6\%}$	$11.1 \pm 0.2\%$	$\mathbf{60.0 \pm 5.5\%}$	$56.7 \pm 7.4\%$
MNIST	0.32	$18.7 \pm 12.8\%$	$84.3 \pm 6.9\%$	$\mathbf{90.1 \pm 5.7\%}$	$11.3 \pm 0.2\%$	$67.1 \pm 6.0\%$	$\mathbf{72.0 \pm 12.1\%}$
MNIST	0.64	$11.3 \pm 0.0\%$	$89.2 \pm 2.1\%$	$\mathbf{92.2 \pm 3.3\%}$	$11.2 \pm 0.2\%$	$\mathbf{72.2 \pm 4.3\%}$	$69.1 \pm 15.6\%$
MNIST	1.00	$11.3 \pm 0.0\%$	$64.5 \pm 47.3\%$	$\mathbf{82.1 \pm 5.2\%}$	$11.1 \pm 0.4\%$	$66.4 \pm 20.1\%$	$\mathbf{67.6 \pm 5.1\%}$
TwoPatterns	0.01	$\mathbf{45.6 \pm 4.8\%}$	$26.7 \pm 0.7\%$	$27.2 \pm 3.3\%$	$\mathbf{28.5 \pm 2.9\%}$	$25.6 \pm 1.6\%$	$27.7 \pm 2.2\%$
TwoPatterns	0.02	$\mathbf{46.3 \pm 2.9\%}$	$27.0 \pm 1.2\%$	$27.1 \pm 2.8\%$	$\mathbf{29.0 \pm 4.1\%}$	$25.6 \pm 1.5\%$	$28.4 \pm 2.3\%$
TwoPatterns	0.04	$\mathbf{42.0 \pm 7.9\%}$	$26.6 \pm 1.3\%$	$26.6 \pm 2.7\%$	$\mathbf{29.1 \pm 4.8\%}$	$25.3 \pm 1.2\%$	$26.5 \pm 3.1\%$
TwoPatterns	0.08	$\mathbf{41.8 \pm 7.3\%}$	$28.0 \pm 1.4\%$	$33.2 \pm 6.8\%$	$29.0 \pm 3.5\%$	$26.0 \pm 1.8\%$	$\mathbf{32.6 \pm 3.8\%}$

		Final test accuracy			First-5 test accuracy		
Dataset	Ratio	Std. RNN	Baseline	TDI	Std. RNN	Baseline	TDI
TwoPatterns	0.16	**$50.3 \pm 0.4\%$**	$35.3 \pm 6.2\%$	$39.8 \pm 3.6\%$	**$34.8 \pm 2.4\%$**	$28.6 \pm 4.0\%$	$32.8 \pm 3.3\%$
TwoPatterns	0.32	**$50.1 \pm 0.4\%$**	$45.0 \pm 5.4\%$	$49.6 \pm 15.9\%$	**$41.2 \pm 4.0\%$**	$32.5 \pm 3.1\%$	$35.1 \pm 4.5\%$
TwoPatterns	0.64	$50.4 \pm 0.5\%$	**$74.9 \pm 1.6\%$**	$58.0 \pm 16.4\%$	**$45.3 \pm 3.1\%$**	$40.5 \pm 2.0\%$	$40.6 \pm 6.2\%$
TwoPatterns	1.00	$50.8 \pm 0.0\%$	**$82.6 \pm 3.6\%$**	$50.1 \pm 12.0\%$	**$47.9 \pm 4.2\%$**	$46.8 \pm 3.5\%$	$37.5 \pm 8.1\%$
Wafer	0.01	$79.7 \pm 14.7\%$	$78.6 \pm 9.9\%$	**$85.2 \pm 8.3\%$**	$50.3 \pm 18.1\%$	$81.5 \pm 11.7\%$	**$83.8 \pm 7.1\%$**
Wafer	0.02	$79.7 \pm 14.7\%$	$76.7 \pm 10.9\%$	**$83.8 \pm 10.2\%$**	$51.1 \pm 19.1\%$	$80.3 \pm 13.0\%$	**$88.3 \pm 1.9\%$**
Wafer	0.04	$78.9 \pm 14.2\%$	$75.9 \pm 11.8\%$	**$86.5 \pm 2.1\%$**	$56.7 \pm 25.6\%$	$80.5 \pm 13.0\%$	**$83.1 \pm 8.9\%$**
Wafer	0.08	$78.8 \pm 14.1\%$	$78.3 \pm 9.7\%$	**$89.7 \pm 4.0\%$**	$56.8 \pm 26.5\%$	$81.5 \pm 12.6\%$	**$85.2 \pm 7.3\%$**
Wafer	0.16	$80.6 \pm 15.0\%$	$89.4 \pm 1.0\%$	**$91.7 \pm 2.8\%$**	$72.2 \pm 18.3\%$	$82.8 \pm 6.2\%$	**$89.6 \pm 0.6\%$**
Wafer	0.32	$89.2 \pm 0.0\%$	$91.4 \pm 1.5\%$	**$91.9 \pm 2.7\%$**	$75.7 \pm 17.3\%$	$83.7 \pm 4.8\%$	**$85.8 \pm 6.6\%$**
Wafer	0.64	$89.2 \pm 0.1\%$	$95.4 \pm 0.5\%$	**$96.0 \pm 1.4\%$**	$84.4 \pm 5.7\%$	$88.2 \pm 1.4\%$	**$89.9 \pm 2.6\%$**
Wafer	1.00	$89.2 \pm 0.0\%$	$97.8 \pm 1.0\%$	**$98.6 \pm 0.4\%$**	$87.3 \pm 3.0\%$	$90.3 \pm 2.2\%$	**$93.5 \pm 0.8\%$**

F.2 Complete Results for Deep SSMs

In this section, we provide full per-dataset and per-ratio results in Table F.8, including the TDI-Frozen variant where the TDI-initialized first S4 layer is kept fixed during supervised training.

Table F.8: **Complete final performance for the SSM baseline, TDI, and TDI-Frozen.** The left block reports final test accuracy; the right block reports mean test accuracy over the first 5 epochs. Each cell reports mean $\pm$ standard deviation over 5 seeds. The best value within each block for each dataset-ratio row is bolded, and the second-best value is underlined.

		Final test accuracy			First-5 test accuracy		
Dataset	Ratio	Baseline	TDI	TDI-Frozen	Baseline	TDI	TDI-Frozen
CIFAR-10	0.005	**$15.0 \pm 1.3\%$**	**$15.0 \pm 1.0\%$**	**$15.0 \pm 0.9\%$**	$10.7 \pm 0.3\%$	**$11.3 \pm 0.4\%$**	**$11.3 \pm 0.4\%$**
CIFAR-10	0.010	$19.9 \pm 1.3\%$	**$23.8 \pm 2.4\%$**	$23.6 \pm 2.3\%$	$10.8 \pm 1.0\%$	**$11.0 \pm 0.7\%$**	**$11.0 \pm 0.7\%$**
CIFAR-10	0.022	$32.8 \pm 3.5\%$	**$38.3 \pm 1.4\%$**	$37.7 \pm 1.7\%$	$12.4 \pm 0.2\%$	**$12.8 \pm 1.3\%$**	**$12.8 \pm 1.4\%$**
CIFAR-10	0.046	$46.2 \pm 1.2\%$	**$46.4 \pm 1.1\%$**	$46.1 \pm 0.9\%$	$14.6 \pm 0.9\%$	**$14.9 \pm 1.1\%$**	$14.8 \pm 1.1\%$
CIFAR-10	0.100	**$55.4 \pm 1.1\%$**	$53.3 \pm 0.6\%$	$53.2 \pm 0.7\%$	$18.9 \pm 1.9\%$	**$22.0 \pm 3.6\%$**	$21.9 \pm 3.4\%$
CIFAR-10	0.215	**$64.7 \pm 0.8\%$**	$59.5 \pm 0.9\%$	$59.2 \pm 0.6\%$	$31.0 \pm 1.8\%$	$35.3 \pm 3.2\%$	**$35.6 \pm 3.1\%$**
CIFAR-10	0.464	**$74.7 \pm 0.9\%$**	$69.3 \pm 0.7\%$	$69.7 \pm 0.9\%$	$46.8 \pm 2.0\%$	**$47.7 \pm 1.7\%$**	$47.4 \pm 1.7\%$
CIFAR-10	1.000	**$81.6 \pm 0.9\%$**	$79.5 \pm 0.6\%$	$79.4 \pm 0.5\%$	**$61.1 \pm 1.1\%$**	$58.0 \pm 1.3\%$	$58.2 \pm 1.3\%$
Freq. Cls.	0.010	$57.9 \pm 13.5\%$	**$73.5 \pm 9.1\%$**	$73.4 \pm 9.3\%$	$18.6 \pm 9.5\%$	**$50.8 \pm 10.7\%$**	$50.5 \pm 10.9\%$
Freq. Cls.	0.022	**$97.3 \pm 0.8\%$**	$93.8 \pm 2.7\%$	$94.4 \pm 2.1\%$	$54.4 \pm 12.4\%$	$74.2 \pm 9.1\%$	**$74.3 \pm 9.0\%$**
Freq. Cls.	0.046	**$98.6 \pm 0.5\%$**	$98.5 \pm 0.4\%$	$98.4 \pm 0.6\%$	$83.7 \pm 3.5\%$	**$91.1 \pm 1.9\%$**	$90.4 \pm 2.0\%$
Freq. Cls.	0.100	$98.6 \pm 0.4\%$	$98.5 \pm 0.6\%$	**$98.7 \pm 0.7\%$**	**$96.8 \pm 0.7\%$**	$96.2 \pm 0.9\%$	$95.9 \pm 0.9\%$
Freq. Cls.	0.215	**$98.9 \pm 0.4\%$**	$98.7 \pm 0.6\%$	$98.8 \pm 0.5\%$	**$98.4 \pm 0.4\%$**	$97.8 \pm 0.2\%$	$97.8 \pm 0.4\%$
Freq. Cls.	0.464	**$99.3 \pm 0.4\%$**	$98.8 \pm 0.4\%$	$99.2 \pm 0.2\%$	**$98.8 \pm 0.3\%$**	$98.6 \pm 0.2\%$	$98.5 \pm 0.2\%$
Freq. Cls.	1.000	$99.3 \pm 0.3\%$	$99.3 \pm 0.2\%$	**$99.4 \pm 0.3\%$**	**$99.0 \pm 0.2\%$**	$98.8 \pm 0.2\%$	$98.9 \pm 0.2\%$
ListOps	0.001	**$12.6 \pm 2.4\%$**	$12.2 \pm 3.4\%$	$11.4 \pm 2.8\%$	$11.0 \pm 1.4\%$	$11.3 \pm 0.8\%$	**$11.4 \pm 0.9\%$**
ListOps	0.002	**$13.3 \pm 1.2\%$**	$12.6 \pm 0.7\%$	$12.8 \pm 0.7\%$	$11.0 \pm 2.5\%$	**$12.9 \pm 2.2\%$**	$12.5 \pm 2.2\%$
ListOps	0.005	**$13.1 \pm 0.5\%$**	$12.5 \pm 0.6\%$	$12.7 \pm 1.2\%$	$12.0 \pm 1.5\%$	$12.1 \pm 1.2\%$	**$12.6 \pm 0.7\%$**
ListOps	0.010	$12.4 \pm 0.5\%$	**$12.5 \pm 0.7\%$**	$11.9 \pm 0.4\%$	$11.8 \pm 1.1\%$	**$12.2 \pm 1.1\%$**	**$12.2 \pm 1.1\%$**
ListOps	0.022	**$13.0 \pm 0.9\%$**	$12.1 \pm 0.8\%$	$11.9 \pm 1.2\%$	$12.5 \pm 0.7\%$	**$13.4 \pm 0.5\%$**	$13.0 \pm 1.4\%$
ListOps	0.046	$20.7 \pm 7.0\%$	**$22.5 \pm 4.2\%$**	$16.1 \pm 6.9\%$	**$16.0 \pm 2.3\%$**	$15.2 \pm 0.5\%$	$14.7 \pm 0.7\%$
ListOps	0.100	**$31.0 \pm 1.1\%$**	$28.8 \pm 0.9\%$	$29.3 \pm 2.7\%$	**$24.8 \pm 5.9\%$**	$18.2 \pm 2.1\%$	$20.3 \pm 5.6\%$
ListOps	0.215	**$36.1 \pm 0.9\%$**	$36.0 \pm 1.1\%$	$33.9 \pm 1.5\%$	**$32.0 \pm 6.5\%$**	$21.7 \pm 6.6\%$	$22.8 \pm 5.3\%$
ListOps	0.464	**$48.5 \pm 1.7\%$**	$46.6 \pm 1.9\%$	$43.7 \pm 2.5\%$	**$36.9 \pm 3.8\%$**	$29.2 \pm 10.0\%$	$28.8 \pm 9.0\%$
ListOps	1.000	**$49.4 \pm 4.8\%$**	$47.0 \pm 5.4\%$	$47.9 \pm 4.1\%$	**$42.0 \pm 0.6\%$**	$32.9 \pm 9.4\%$	$29.8 \pm 9.5\%$
sMNIST	0.005	$10.0 \pm 0.3\%$	**$10.3 \pm 0.9\%$**	**$10.3 \pm 0.9\%$**	**$10.2 \pm 0.1\%$**	$10.0 \pm 0.1\%$	$10.0 \pm 0.1\%$
sMNIST	0.010	$10.7 \pm 0.8\%$	**$38.6 \pm 22.0\%$**	$33.7 \pm 21.7\%$	**$10.2 \pm 0.2\%$**	$10.1 \pm 0.4\%$	$10.1 \pm 0.3\%$
sMNIST	0.022	$64.9 \pm 30.8\%$	**$91.9 \pm 4.8\%$**	$80.8 \pm 26.6\%$	$10.2 \pm 0.1\%$	$10.7 \pm 0.3\%$	**$11.0 \pm 0.6\%$**
sMNIST	0.046	$97.6 \pm 0.2\%$	**$97.7 \pm 0.3\%$**	$97.5 \pm 0.4\%$	$10.3 \pm 0.4\%$	**$11.8 \pm 2.9\%$**	$11.3 \pm 1.9\%$
sMNIST	0.100	**$98.5 \pm 0.1\%$**	$98.4 \pm 0.2\%$	$98.4 \pm 0.0\%$	$12.0 \pm 3.7\%$	**$41.9 \pm 17.7\%$**	$39.2 \pm 15.7\%$
sMNIST	0.215	$98.8 \pm 0.2\%$	$98.8 \pm 0.2\%$	**$99.0 \pm 0.1\%$**	$59.6 \pm 9.1\%$	**$66.9 \pm 7.7\%$**	$65.6 \pm 8.1\%$
sMNIST	0.464	**$99.1 \pm 0.1\%$**	**$99.1 \pm 0.2\%$**	**$99.1 \pm 0.1\%$**	$81.5 \pm 5.9\%$	$91.3 \pm 5.8\%$	**$91.8 \pm 4.3\%$**
sMNIST	1.000	**$99.2 \pm 0.1\%$**	$99.1 \pm 0.2\%$	$99.1 \pm 0.1\%$	**$98.5 \pm 0.2\%$**	**$98.5 \pm 0.1\%$**	**$98.5 \pm 0.2\%$**
Pathfinder	0.001	$50.0 \pm 0.4\%$	**$50.1 \pm 0.2\%$**	$49.9 \pm 0.1\%$	**$50.0 \pm 0.2\%$**	**$50.0 \pm 0.2\%$**	$49.9 \pm 0.2\%$
Pathfinder	0.002	**$50.4 \pm 0.4\%$**	$50.1 \pm 0.1\%$	$50.1 \pm 0.4\%$	**$50.1 \pm 0.3\%$**	$49.9 \pm 0.2\%$	$49.9 \pm 0.2\%$
Pathfinder	0.005	$50.2 \pm 0.4\%$	**$50.4 \pm 0.5\%$**	$50.1 \pm 0.2\%$	**$50.1 \pm 0.1\%$**	$50.0 \pm 0.1\%$	$49.9 \pm 0.2\%$
Pathfinder	0.010	$50.2 \pm 0.4\%$	**$50.3 \pm 0.4\%$**	$50.2 \pm 0.6\%$	**$50.0 \pm 0.1\%$**	$49.9 \pm 0.2\%$	$49.9 \pm 0.1\%$
Pathfinder	0.022	**$50.7 \pm 0.6\%$**	**$50.7 \pm 0.4\%$**	$50.4 \pm 0.3\%$	**$50.0 \pm 0.2\%$**	**$50.0 \pm 0.1\%$**	**$50.0 \pm 0.1\%$**
Pathfinder	0.046	$51.6 \pm 0.5\%$	**$52.5 \pm 1.1\%$**	$51.5 \pm 0.9\%$	**$50.1 \pm 0.1\%$**	$50.0 \pm 0.2\%$	$50.0 \pm 0.1\%$
Pathfinder	0.100	$56.8 \pm 1.2\%$	**$57.4 \pm 1.4\%$**	$54.8 \pm 0.8\%$	**$50.2 \pm 0.2\%$**	$50.1 \pm 0.1\%$	$50.1 \pm 0.2\%$

		Final test accuracy			First-5 test accuracy		
Dataset	Ratio	Baseline	TDI	TDI-Frozen	Baseline	TDI	TDI-Frozen
Pathfinder	0.215	$62.8 \pm 2.5\%$	$\mathbf{64.5 \pm 0.8\%}$	$59.6 \pm 2.6\%$	$50.0 \pm 0.1\%$	$50.1 \pm 0.2\%$	$\mathbf{50.4 \pm 0.3\%}$
Pathfinder	0.464	$69.4 \pm 8.8\%$	$\mathbf{70.9 \pm 1.4\%}$	$66.3 \pm 1.1\%$	$50.1 \pm 0.3\%$	$50.1 \pm 0.2\%$	$\mathbf{50.2 \pm 0.1\%}$
Pathfinder	1.000	$67.8 \pm 16.4\%$	$69.7 \pm 14.1\%$	$\mathbf{80.8 \pm 2.4\%}$	$51.9 \pm 3.1\%$	$50.0 \pm 0.1\%$	$\mathbf{52.8 \pm 4.1\%}$
pMNIST	0.005	$10.0 \pm 0.3\%$	$\mathbf{10.3 \pm 0.6\%}$	$\mathbf{10.3 \pm 0.6\%}$	$\mathbf{10.2 \pm 0.1\%}$	$10.1 \pm 0.1\%$	$10.1 \pm 0.1\%$
pMNIST	0.010	$10.7 \pm 0.8\%$	$\mathbf{11.6 \pm 1.2\%}$	$11.3 \pm 0.8\%$	$\mathbf{10.3 \pm 0.2\%}$	$10.2 \pm 0.3\%$	$10.2 \pm 0.3\%$
pMNIST	0.022	$\mathbf{54.9 \pm 20.6\%}$	$54.2 \pm 26.0\%$	$51.1 \pm 26.5\%$	$10.2 \pm 0.2\%$	$\mathbf{10.3 \pm 0.3\%}$	$\mathbf{10.3 \pm 0.3\%}$
pMNIST	0.046	$91.6 \pm 0.7\%$	$\mathbf{92.1 \pm 0.6\%}$	$91.5 \pm 1.0\%$	$\mathbf{10.8 \pm 0.8\%}$	$10.1 \pm 0.3\%$	$10.1 \pm 0.3\%$
pMNIST	0.100	$94.0 \pm 1.2\%$	$94.7 \pm 0.5\%$	$\mathbf{94.8 \pm 0.5\%}$	$12.9 \pm 2.9\%$	$20.8 \pm 9.6\%$	$\mathbf{21.4 \pm 10.0\%}$
pMNIST	0.215	$96.1 \pm 0.4\%$	$95.4 \pm 0.7\%$	$\mathbf{96.3 \pm 0.4\%}$	$42.8 \pm 16.8\%$	$\mathbf{50.9 \pm 7.0\%}$	$49.0 \pm 7.2\%$
pMNIST	0.464	$\mathbf{97.0 \pm 0.3\%}$	$\mathbf{97.0 \pm 0.2\%}$	$96.9 \pm 0.1\%$	$\mathbf{79.9 \pm 1.9\%}$	$78.7 \pm 1.5\%$	$73.6 \pm 9.5\%$
pMNIST	1.000	$\mathbf{97.7 \pm 0.2\%}$	$97.6 \pm 0.1\%$	$97.6 \pm 0.2\%$	$94.2 \pm 0.9\%$	$93.6 \pm 1.8\%$	$\mathbf{94.5 \pm 1.3\%}$
SC	0.001	$7.6 \pm 2.5\%$	$7.5 \pm 3.8\%$	$\mathbf{10.8 \pm 3.0\%}$	$2.8 \pm 1.0\%$	$3.5 \pm 1.0\%$	$\mathbf{3.6 \pm 0.7\%}$
SC	0.002	$12.2 \pm 4.1\%$	$\mathbf{16.5 \pm 7.5\%}$	$15.6 \pm 5.7\%$	$3.0 \pm 0.6\%$	$\mathbf{4.4 \pm 0.6\%}$	$4.3 \pm 0.6\%$
SC	0.005	$35.2 \pm 6.7\%$	$\mathbf{44.7 \pm 4.4\%}$	$40.8 \pm 12.2\%$	$3.6 \pm 0.5\%$	$\mathbf{6.8 \pm 0.8\%}$	$\mathbf{6.8 \pm 1.4\%}$
SC	0.010	$62.3 \pm 3.1\%$	$\mathbf{65.5 \pm 5.7\%}$	$62.8 \pm 4.1\%$	$11.4 \pm 4.6\%$	$\mathbf{16.9 \pm 5.3\%}$	$16.7 \pm 5.3\%$
SC	0.022	$75.7 \pm 0.7\%$	$\mathbf{77.1 \pm 2.1\%}$	$74.6 \pm 2.3\%$	$25.0 \pm 3.1\%$	$\mathbf{41.5 \pm 1.8\%}$	$39.2 \pm 3.3\%$
SC	0.046	$\mathbf{82.9 \pm 1.0\%}$	$\mathbf{82.9 \pm 1.2\%}$	$82.1 \pm 0.5\%$	$50.0 \pm 6.8\%$	$\mathbf{62.1 \pm 3.6\%}$	$61.1 \pm 3.7\%$
SC	0.100	$\mathbf{86.9 \pm 0.8\%}$	$86.0 \pm 1.9\%$	$86.6 \pm 1.0\%$	$70.1 \pm 3.7\%$	$74.0 \pm 2.2\%$	$\mathbf{75.0 \pm 1.6\%}$
SC	0.215	$86.4 \pm 1.8\%$	$\mathbf{88.1 \pm 1.4\%}$	$87.6 \pm 1.3\%$	$78.4 \pm 4.5\%$	$\mathbf{83.1 \pm 0.6\%}$	$82.6 \pm 1.2\%$
SC	0.464	$87.4 \pm 1.8\%$	$\mathbf{88.1 \pm 1.4\%}$	$87.8 \pm 0.7\%$	$83.9 \pm 0.5\%$	$\mathbf{84.9 \pm 0.8\%}$	$84.4 \pm 0.7\%$
SC	1.000	$88.0 \pm 0.8\%$	$87.5 \pm 3.6\%$	$\mathbf{88.7 \pm 0.6\%}$	$85.5 \pm 1.0\%$	$85.6 \pm 0.6\%$	$\mathbf{86.2 \pm 0.8\%}$
SC10	0.001	$\mathbf{10.9 \pm 1.1\%}$	$10.6 \pm 0.7\%$	$10.7 \pm 1.0\%$	$10.1 \pm 0.2\%$	$\mathbf{10.2 \pm 0.3\%}$	$\mathbf{10.2 \pm 0.2\%}$
SC10	0.002	$17.5 \pm 5.3\%$	$17.8 \pm 6.3\%$	$\mathbf{18.4 \pm 4.7\%}$	$10.3 \pm 0.4\%$	$\mathbf{10.5 \pm 0.7\%}$	$\mathbf{10.5 \pm 0.7\%}$
SC10	0.005	$23.7 \pm 8.3\%$	$\mathbf{27.9 \pm 9.6\%}$	$25.3 \pm 6.9\%$	$10.3 \pm 0.3\%$	$\mathbf{12.9 \pm 1.2\%}$	$\mathbf{12.9 \pm 1.9\%}$
SC10	0.010	$43.0 \pm 10.9\%$	$\mathbf{59.5 \pm 3.1\%}$	$57.5 \pm 10.0\%$	$11.2 \pm 0.9\%$	$14.5 \pm 1.7\%$	$\mathbf{14.8 \pm 1.5\%}$
SC10	0.022	$76.7 \pm 2.7\%$	$\mathbf{79.0 \pm 2.7\%}$	$77.8 \pm 5.4\%$	$14.6 \pm 2.0\%$	$\mathbf{24.3 \pm 5.3\%}$	$22.5 \pm 4.0\%$
SC10	0.046	$84.5 \pm 4.4\%$	$\mathbf{85.8 \pm 1.1\%}$	$84.6 \pm 2.3\%$	$36.2 \pm 8.1\%$	$\mathbf{46.5 \pm 6.7\%}$	$40.9 \pm 3.4\%$
SC10	0.100	$87.8 \pm 2.8\%$	$\mathbf{89.4 \pm 1.0\%}$	$87.8 \pm 2.4\%$	$59.5 \pm 8.6\%$	$66.3 \pm 5.9\%$	$\mathbf{67.1 \pm 3.8\%}$
SC10	0.215	$91.4 \pm 0.8\%$	$\mathbf{92.1 \pm 0.5\%}$	$91.1 \pm 1.0\%$	$77.2 \pm 6.3\%$	$\mathbf{82.8 \pm 1.7\%}$	$82.2 \pm 2.5\%$
SC10	0.464	$92.5 \pm 0.5\%$	$\mathbf{92.7 \pm 0.8\%}$	$91.6 \pm 1.7\%$	$84.4 \pm 2.5\%$	$88.1 \pm 1.1\%$	$\mathbf{88.3 \pm 1.3\%}$
SC10	1.000	$\mathbf{93.4 \pm 0.9\%}$	$93.3 \pm 0.3\%$	$93.2 \pm 1.1\%$	$89.7 \pm 1.3\%$	$90.5 \pm 0.3\%$	$\mathbf{90.6 \pm 0.7\%}$

F.3 Early Learning versus Extended Training

The main experiments report performance under fixed training budgets. To examine whether the observed gains persist with longer training, we extend the training budget from 30 to 100 epochs on sMNIST and from 10 to 50 epochs on Frequency Classification. We compare the baseline and TDI using five matched seeds at the training ratios listed in Table F.9.

Evaluation metrics. We report three complementary measures. *Early-10* is the mean test accuracy over the first ten epochs, computed separately for each run and then averaged over five seeds. *Time-to-80%* is the first completed epoch at which a run reaches 80% test accuracy, reported as mean $\pm$ standard deviation across seeds. All runs reach this threshold. *Final* Δ*Acc.* is the TDI-minus-baseline difference in test accuracy at the final epoch of the extended training budget, computed for each matched seed pair and reported as mean $\pm$ standard deviation in percentage points.

Results and interpretation. On sMNIST, TDI reaches 80% test accuracy earlier and achieves higher Early-10 accuracy at both evaluated training ratios, while the baseline largely catches up by 100 epochs. For example, at 2.15% training data, the mean time-to-80% decreases from 34.2 to 20.0 epochs, whereas the final accuracy difference is only $+0.12 \pm 0.47$ percentage points. Frequency Classification shows a similar early-learning advantage at 2.15% and 10.0% training data, but little early-learning difference at 21.5%. After 50 epochs, final accuracy differences are small and can favor either initialization.

These experiments indicate that, in the evaluated settings, TDI primarily improves the learning trajectory without systematically increasing accuracy after extended training. They qualify the interpretation of the main experiments: a higher final-epoch accuracy under a fixed training budget does not by itself establish a persistent advantage with longer training. These finite-horizon results do not establish asymptotic performance or imply that the same behavior holds on every benchmark.

Table F.9: **Early learning and accuracy after extended training.** Time-to-80% reports the number of epochs required to reach 80% test accuracy (mean ± SD over five seeds). Early-10 reports mean test accuracy over the first ten epochs, averaged over seeds. Final ΔAcc. reports the paired TDI-minus-baseline final-epoch accuracy difference (mean ± SD), in percentage points (pp). Bold indicates the lower mean Time-to-80% and the higher Early-10 accuracy within each row; ties are both bolded.

			Time-to-80% (epochs)		Early-10 (%)		Final ΔAcc.
Task	Train ratio	Epochs	Baseline	TDI	Baseline	TDI	(pp)
sMNIST	2.15%	100	34.2 ± 4.5	**20.0 ± 4.6**	10.19	**14.15**	+0.12 ± 0.47
sMNIST	21.5%	100	4.6 ± 0.9	**3.0 ± 0.0**	69.71	**83.69**	−0.08 ± 0.07
Freq. Cls.	2.15%	50	11.6 ± 1.7	**9.6 ± 4.0**	32.72	**63.25**	−0.69 ± 0.73
Freq. Cls.	10.0%	50	2.0 ± 0.0	**1.2 ± 0.4**	92.04	**96.03**	+0.29 ± 0.29
Freq. Cls.	21.5%	50	**1.0 ± 0.0**	**1.0 ± 0.0**	**98.04**	97.93	+0.13 ± 0.15

F.4 First-Layer versus All-Layer TDI

We examine whether applying TDI beyond the first SSM layer improves performance. In the all-layer variant, we apply the same input-derived Fisher log-power spectrum to every SSM layer, whereas the main method applies TDI only to the first layer. The two variants use the same architecture and training protocol and differ only in which layers receive TDI. We use five matched random seeds and train both variants for 30 epochs on sMNIST, 30 epochs on CIFAR-10, and 20 epochs on Frequency Classification.

Table F.10: **First-layer versus all-layer TDI.** Test accuracy (%) is reported as mean ± standard deviation over five matched seeds. Δ denotes the paired per-seed accuracy difference (all-layer minus first-layer), reported as mean ± standard deviation in percentage points. Bold indicates the higher mean accuracy in each row.

Dataset	Train ratio	First-layer TDI	All-layer TDI	Δ (pp)
sMNIST	0.464%	10.47 ± 0.85	**30.39 ± 14.76**	+19.92 ± 14.81
sMNIST	1.000%	34.97 ± 23.93	**80.41 ± 5.61**	+45.45 ± 23.50
sMNIST	2.154%	91.64 ± 3.74	**95.22 ± 1.28**	+3.58 ± 4.15
CIFAR-10	0.464%	15.85 ± 1.36	**17.79 ± 1.65**	+1.94 ± 2.35
CIFAR-10	1.000%	22.63 ± 2.37	**27.58 ± 3.32**	+4.95 ± 3.43
CIFAR-10	2.154%	**37.03 ± 1.92**	33.93 ± 0.97	−3.09 ± 2.20
Freq. Cls.	0.464%	**78.34 ± 3.56**	72.60 ± 5.02	−5.74 ± 5.61
Freq. Cls.	1.000%	84.98 ± 6.26	**87.50 ± 5.80**	+2.52 ± 1.86
Freq. Cls.	2.154%	96.28 ± 1.33	**96.54 ± 0.93**	+0.26 ± 1.90

Table F.10 shows that all-layer TDI achieves higher mean accuracy at all three sMNIST ratios and the two smallest CIFAR-10 ratios, but underperforms first-layer TDI at the largest CIFAR-10 ratio and the smallest Frequency Classification ratio. The difference at the largest Frequency Classification ratio is small. Thus, all-layer TDI can help under the evaluated training budgets, but is not uniformly preferable to first-layer TDI.

This ablation reuses the input-level spectrum at every depth, although deeper layers receive representations transformed by preceding blocks. It therefore does not constitute a layer-dependent spectral initialization method. We retain first-layer TDI as the main method because it most directly follows our input-level analysis, while acknowledging that it is not optimal in every evaluated setting. Estimating task-relevant spectra from intermediate representations remains future work.

F.5 Label Permutation during Task-Spectrum Estimation

To examine whether the benefits of TDI depend on label-relevant spectral structure, we compare task spectra estimated using the true training labels (TDI-True) with those estimated after randomly permuting the training labels (TDI-Permuted). The permutation is applied only during task-spectrum

estimation; downstream supervised training uses the original, correct labels in both conditions. We also include the default initialization as a baseline. Table F.11 reports the number of epochs required to reach 80% test accuracy, defined as the first completed epoch at which this threshold is reached. Results are reported as mean ± standard deviation over five matched seeds.

Table F.11: **Effect of label permutation during task-spectrum estimation.** Entries report epochs to reach 80% test accuracy (mean ± SD over five matched seeds; lower is better). Bold indicates the lowest mean in each row.

Dataset	Train ratio	TDI-True	TDI-Permuted	Baseline
sMNIST	2.15%	**20.0 ± 4.6**	28.6 ± 8.7	34.2 ± 4.5
sMNIST	4.64%	**11.4 ± 2.6**	13.4 ± 2.9	15.4 ± 6.3
sMNIST	10%	**5.2 ± 1.3**	7.0 ± 2.1	10.6 ± 1.8
Freq. Cls.	2.15%	**9.6 ± 4.0**	21.6 ± 13.8	11.6 ± 1.7
Freq. Cls.	4.64%	**2.8 ± 1.6**	7.0 ± 3.7	4.2 ± 0.8
Freq. Cls.	10%	**1.2 ± 0.4**	3.8 ± 1.9	2.0 ± 0.0

TDI-True has the lowest mean time-to-80% in all six settings. Permuting labels increases the mean time by 1.8–12.0 epochs relative to TDI-True, and TDI-Permuted is not consistently faster than the baseline. These results support the role of label-dependent spectral structure in the observed early-learning benefits of TDI.

G Existing Assets

We use standard public benchmarks and existing open-source software only for research evaluation. Table G.12 lists the existing assets used in our experiments, their original sources, and license or terms-of-use information when available. We do not redistribute these datasets. For CIFAR-10, we download data from official CIFAR-10 website. For UCR Time Series Archive, we download data from the official archive website and cite the archive paper.

Table G.12: **Existing assets used in the experiments.**

Asset	Use in this paper	License / Terms
MNIST [18]	One-layer SSMs and Deep SSMs	MIT License
CIFAR-10 [32]	Deep SSMs	Official source
UCR Time Series Archive [17]	One-layer SSMs	Official source
PathFinder [47]	Deep SSMs	Apache-2.0 License
ListOps [35, 47]	Deep SSMs	Apache-2.0 License
Speech Commands / SC10 [48]	Deep SSMs	CC-BY 4.0 License
Kernel Regression implementation [11]	Kernel Regression	MIT License
S4/S4D implementation [24]	Deep SSMs	Apache-2.0 License

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims are stated in the abstract and Introduction, including the kernel-theoretic characterization of LTI SSM inductive bias and the proposed Task-Dependent Initialization method. The empirical scope of the claims is validated in Section 4, where we emphasize that the gains are strongest in low-data regimes and under spectral mismatch.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in Section 5, including the reliance of our theory on the LTI SSM case, its current focus on one-dimensional scalar sequences, and the restriction of TDI to first-layer initialization. We also clarify that TDI is not intended as a universal improvement and is expected to be most effective when task-relevant structure is spectral and mismatched with the default SSM bias.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The theoretical assumptions and statements are given in Sections 2–3, including the LTI SSM formulation, whitened-coordinate analysis, and absolute-summability condition for the frequency-response result. Proofs of the main theorems and supporting lemmas are provided in Appendix B.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide the information needed to reproduce the experiments supporting the main claims in Sections 3–4 and Appendices C–G, including the TDI procedure, dataset choices, training-ratio protocol, model configurations, hyperparameters, random seeds, and evaluation metrics. Appendices F–G further report full per-dataset and per-ratio results for the one-layer and deep SSM experiments.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

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

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

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

   Answer: [Yes]

   Justification: We provide a public code repository with environment files, README documentation, and scripts required to reproduce the main experiments. The repository documents the software environment, dataset preparation procedures, and commands for running the kernel regression, one-layer SSM, and deep SSM experiments. It also specifies how datasets are obtained or generated, including public datasets used in the experiments and synthetic datasets created by our code.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We describe the experimental settings in Section 4 and provide full details in Appendix E, including datasets, data splits, training-ratio protocols, model configurations, optimizers, hyperparameters, evaluation metrics, and random seed counts. Full per-dataset and per-ratio results are reported in Appendices F–G to make the reported trends interpretable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report variability across random seeds and ratio-level averages in Tables 1 and 3, where each cell gives the mean and standard deviation as defined in the captions. We further report paired TDI–baseline improvements with cluster-bootstrap 95% confidence intervals in Tables 2 and 4, matching runs by dataset, training ratio, and seed.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report compute resources in Appendix E, including GPU types, the one-GPU-per-run setup, experiment scale in terms of datasets, ratios, seeds, and model variants, and the approximate total budget of 400–500 GPU-hours. We also state that TDI only adds an initialization step and does not change inference cost.

   Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and believe that the research conducted in this paper conforms to it.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

Answer: [N/A]

Justification: The paper is foundational and methodological, focusing on the theoretical characterization of spectral inductive bias in LTI SSMs and a task-dependent initialization procedure. It does not introduce a deployed system, application-specific model, new dataset, or capability with a direct path to identifiable societal impacts beyond general improvements in sequence-model training.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release data or models with a high risk of misuse, such as pretrained language models, image generators, or scraped datasets. The experiments use public benchmark datasets and synthetic data, and the proposed method is an initialization procedure rather than a deployable high-risk model.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the original sources for the public datasets and existing code assets used in the experiments, and Appendix H lists the available license or terms-of-use information for each asset. For assets whose official sources do not specify a standard license, such as CIFAR-10 and the UCR Time Series Archive, we use the official sources, provide the requested citations, and do not redistribute the datasets.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The new assets released with the submission consist of an anonymized code package and accompanying documentation, including environment specifications, reproduction commands, and license information for each component. The release does not include newly collected human-subject data, and the experiments rely on synthetic or standard public datasets that are documented in the repository.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments, human-subject research, user studies, or participant data collection. All experiments use synthetic data or standard public machine-learning benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, human-subject research, user studies, or participant data collection. Therefore, IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development, theoretical analysis, and experimental evaluation in this paper do not involve LLMs as an important, original, or non-standard component. Any LLM assistance, if used, was limited to writing, editing, or formatting and did not affect the methodology, scientific claims, or experimental results.

    Guidelines:

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