WaLRUS: Wavelets for Long-range Representation Using SSMs

Hossein Babaei Mel White Sina Alemohammad Richard G. Baraniuk
Department of Electrical and Computer Engineering, Rice University
{hb26,mel.white,sa86,richb}@rice.edu

Abstract

State-Space Models (SSMs) have proven to be powerful tools for modeling long-range dependencies in sequential data. While the recent method known as HiPPO has demonstrated strong performance, and formed the basis for machine learning models S4 and Mamba, it remains limited by its reliance on closed-form solutions for a few specific, well-behaved bases. The SaFARi framework generalized this approach, enabling the construction of SSMs from arbitrary frames, including non-orthogonal and redundant ones, thus allowing an infinite diversity of possible "species" within the SSM family. In this paper, we introduce WaLRUS (Wavelets for Long-range Representation Using SSMs).We compare WaLRUS to HiPPO-based models, and demonstrate improved accuracy and more efficient implementations for online function approximation tasks.

1 Introduction

Sequential data is foundational to many machine learning tasks, including natural language processing, speech recognition, and video understanding [1–3]. These applications require models that can effectively process and retain information over long time horizons. A central challenge in this setting is the efficient representation of long-range dependencies in a way that preserves essential features of the input signal for downstream tasks, while remaining computationally tractable during both training and inference [4].

Recurrent neural networks (RNNs) are traditional choices for modeling sequential data, but struggle with long-term dependencies due to vanishing or exploding gradients during backpropagation through time [4–6]. While gated variants like LSTMs [7] and GRUs [8] mitigate some issues, they require significant tuning and lack compatibility with parallel processing, hindering scalability.

State-space models (SSMs) offer a linear and principled framework for encoding temporal information, and have re-emerged as a powerful alternative for online representation of sequential data [9–16]. By design, they enable the online computation of compressive representations that summarize the entire input history using a fixed-size state vector, ensuring a constant memory footprint regardless of sequence length. A major breakthrough came with HiPPO (High-order Polynomial Projection Operators), which reformulates online representation as a function approximation problem using orthogonal polynomial bases [9]. This approach underpins state-of-the-art models like S4 and Mamba, enabling compact representations for long-range dependencies [10, 11].

However, existing SSMs primarily rely on Legendre and Fourier bases, which, although effective for smooth or periodic signals, struggle with non-stationary and localized features [9, 10]. These challenges are especially evident in domains such as audio, geophysics, and biomedical signal processing, where rapid transitions and sparse structure are common.

To address this limitation, the SaFARi framework (State-Space Models for Frame-Agnostic Representation) extends HiPPO to arbitrary frames, including non-orthogonal and redundant bases [13, 14, 17].

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

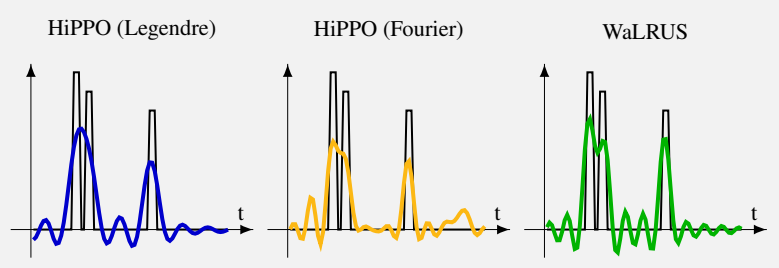

Figure 1: An input signal comprising three random spikes is sequentially processed by SSMs and reconstructed after observing the entire input. Only the wavelet-based SSM constructed using WaLRUS can clearly distinguish adjacent spikes.

This generalization enables SSM construction from any frame via numerical solutions of first-order linear differential equations, preserving HiPPO's memory efficiency and update capabilities without closed-form restrictions.

In this paper, we leverage the SaFARi method with wavelet frames to introduce a new model, WaLRUS (Wavelets for Long-range Representation Using SSMs). We derive our model using Daubechies wavelets with two variants: scaled-WaLRUS and translated-WaLRUS, designed for capturing non-smooth and localized features through compactly supported, multi-resolution wavelet decompositions [18]. These properties allow WaLRUS to retain fine-grained signal details typically lost by polynomial-based models.

We also provide a comparative analysis of WaLRUS and existing HiPPO variants (see Fig. 1). Empirical results demonstrate that the wavelet-based WaLRUS model consistently outperforms Legendre and Fourier-based HiPPO models in reconstruction accuracy, especially on signals with sharp transients. Furthermore, WaLRUS has been experimentally observed to be stably diagonalizable, which is the key enabler of efficient convolution-based implementations and parallel computation [13, 14].

These results highlight the practical advantages of WaLRUS models, particularly in scenarios where signal structure varies across time and scale. By bridging multiscale signal analysis and online function approximation, WaLRUS opens new directions for modeling complex temporal phenomena across disciplines.

2 Background

Recent advances in machine learning, computer vision, and large language models have pushed the frontier of learning from long sequences of data. These applications demand models that can (1) generate compact representations of input streams, (2) preserve long-range dependencies, and (3) support efficient online updates.

Classical linear methods, such as the Fourier transform, offer compact representations in the frequency domain [19–23]. However, they are ill-suited for online processing: each new input requires recomputing the entire representation, making them inefficient for streaming data and limited in their memory horizon. Nonlinear models like recurrent neural networks (RNNs) and their gated variants (LSTMs, GRUs) have been more successful in sequence modeling, but they face well-known issues such as vanishing/exploding gradients and limited parallelization [4–6, 8]. Moreover, their representations are task-specific, and not easily repurposed across different settings.

To resolve these issues, the HiPPO framework [9] casts online function approximation as a continuous projection of the input $u(t)$ onto a linear combination of the given basis functions $\mathcal{G}$. At every time T, it produces a compressed state vector $\vec{c}(T)$ that satisfies the update rule:

$$\frac{d}{dT}\vec{c}(T) = -A_{(T)}\vec{c}(T) + B_{(T)}u(T). \tag{1}$$

Here, $A_{(T)}$ and $B_{(T)}$ are derived based on the choice of polynomial basis and measure $\mu(t)$, which defines how recent history is weighted. Two commonly used measures are:

$$\mu_{tr}(t) = \frac{1}{\theta}\mathbb{1}_{t\in[T-\theta,T]}, \quad \mu_{sc}(t) = \frac{1}{T}\mathbb{1}_{t\in[0,T]}. \tag{2}$$

The translated measure μ_{tr} emphasizes recent history within a sliding window of length θ, while the scaled measure μ_{sc} compresses the entire input history into a fixed-length representation.

Despite its strengths, HiPPO is restricted to only a few bases (e.g., Legendre, Fourier), and deriving $A(t)$ and $B(t)$ in closed form is only tractable for specific basis-measure combinations.

SaFARi addressed this limitation by generalizing online function approximation to any arbitrary frame [17]. A frame $\Phi(t)$ is a set of elements $\{\phi_i(t)\}$ such that one can reconstruct any input $g(t)$ by knowing the inner products $\langle g(t), \phi_i(t)\rangle$. For a given frame Φ, its complex conjugate $\overline{\Phi}$, and its dual $\widetilde{\Phi}$, the scaled-SaFARi produces an SSM with A and B given by:

$$\frac{\partial}{\partial T}\vec{c}(T) = -\frac{1}{T}A\vec{c}(T) + \frac{1}{T}Bu(T), \quad A_{i,j} = \delta_{i,j} + \int_0^1 t' \frac{\partial}{\partial t}\overline{\phi}_i\Big|_{t=t'} \widetilde{\phi}_j(t')dt', \quad B_i = \overline{\phi}_i(1) \tag{3}$$

while the translated-SaFARi produces an SSM with the A and B given by:

$$\frac{\partial}{\partial T}\vec{c}(T) = -\frac{1}{\theta}A\vec{c}(T) + \frac{1}{\theta}Bu(T), \quad A_{i,j} = \overline{\phi}_i(0)\widetilde{\phi}_j(0) + \int_0^1 \frac{\partial}{\partial t}\overline{\phi}_i\Big|_{t=t'} \widetilde{\phi}_j(t')dt', \; B_i = \overline{\phi}_i(1) \tag{4}$$

In the appendix, we provide a some theoretical background on Eq. 3 and Eq. 4 from [17].

Incremental update of SSMs: The differential equation in Eq. 1 can be solved incrementally. Following [9], we adopt the Generalized Bilinear Transform (GBT) [24] given by Eq. 5 for its superior numerical accuracy in first order SSMs.

$$c(t+\Delta t) = (I + \delta t\alpha A_{t+\delta t})^{-1}\left[(I - \delta t(1-\alpha)A_t)c(t) + \delta t B(t)u(t)\right] \tag{5}$$

Diagonalization of A: Each GBT step involves matrix inversion and multiplication. If $A(t)$ has time-independent eigenvectors (e.g., $A(t) = g(t)A$), it can be diagonalized as $A(t) = V\Lambda(t)V^{-1}$, allowing a change of variables $\widetilde{c} = V^{-1}c$ and $\widetilde{B} = V^{-1}B(t)$, yielding:

$$\frac{\partial}{\partial t}\widetilde{c} = -\Lambda(t)\widetilde{c} + \widetilde{B}u(t), \tag{6}$$

This reduces each update to elementwise operations, significantly lowering computational cost.

2.1 Wavelet Frames

Wavelet frames offer a multiresolution analysis that captures both temporal and frequency characteristics of signals, making them particularly effective for representing non-stationary or long-range dependent data [25]. Initiated by [26] and formalized by [27], wavelet theory gained prominence with Ingrid Daubechies' seminal work [28], which introduced compactly supported orthogonal wavelets. Since then, wavelets have played a central role in modern signal processing [29].

Wavelet analysis decomposes a signal $f(t)$ into dilations and translations of a mother wavelet $\psi(t)$, enabling simultaneous localization in time and frequency. The *discrete wavelet transform* is

$$W(j,k) = \int_{-\infty}^{\infty} f(t)\psi^*_{j,k}(t)\,dt, \quad \psi_{j,k}(t) = \frac{1}{\sqrt{2^{-j}}}\psi\left(\frac{t-k}{2^{-j}}\right).$$

Unlike global bases such as Fourier or polynomials, which struggle with localized discontinuities, wavelets provide sparse representations of signals with singularities, such as jumps or spikes [18, 30]. Their local support yields small coefficients in smooth regions and large coefficients near singularities, enabling efficient compression and accurate reconstruction. These properties make wavelet frames a natural and powerful choice for time-frequency analysis in a wide range of practical applications.

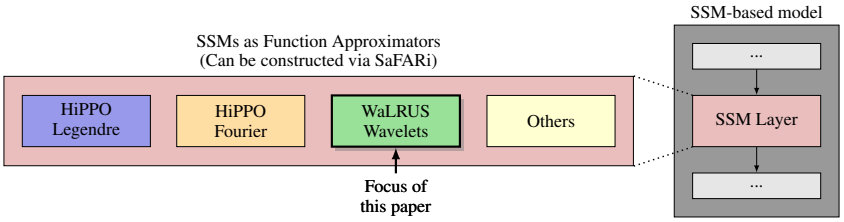

Figure 2: A diagram of the relationships between HiPPO, SaFARi, WaLRUS (this work), and SSM-based models such as S4 and Mamba. The focus of this work is on the development of a wavelet-based SSM in a function approximation task, which could later be used as a drop-in replacement for the SSM layer in a learned model.

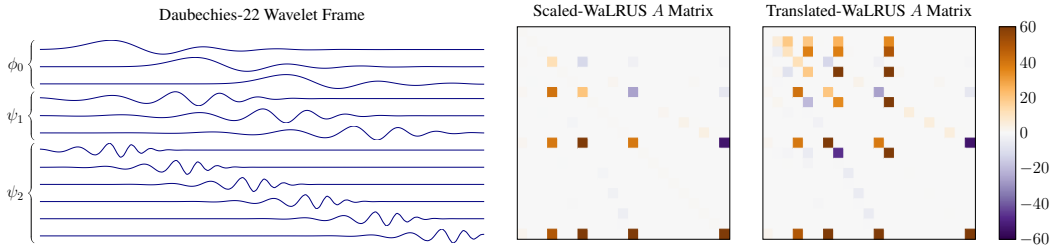

Figure 3: Left: Elements of a Daubechies-22 wavelet frame, with father wavelet ϕ, mother wavelet ψ, and two scales. Right: The scaled and translated A matrices for WaLRUS with $N = 21$.

3 WaLRUS: Wavelet-based SSMs

Daubechies wavelets [18, 28] provide a particularly useful implementation of a SaFARi SSM. While there are different types of commonly used wavelets, Daubechies wavelets are of particular interest in signal representation due to their maximal vanishing moments over compact support.

To construct the frame, we use the usual dyadic scaling for multiresolution analysis; that is, scaling the mother wavelets by a factor of two at each level. For each scale, different shifts along the x-axis are introduced. Compressive wavelet frames are truncated versions of wavelet frames that contain only a few of the coarser scales, and introduce overlapping shifts to keep the expressivity and satisfy the frame condition (See Mallat, [29]). The interplay between the retained scales and the minimum required overlap to maintain the expressivity is extensively studied in the wavelet literature [18, 28, 29]. If there is excess overlap in shifts, the wavelet frame becomes redundant, and redundancy has advantages in expressivity and robustness to noise.

Figure 3, left, gives a visual representation of how we construct such a frame. The frame consists of shifted copies of the father wavelet ϕ at one scale, and shifted copies of a mother wavelet ψ at different scales, with overlaps that introduce redundancy. Figure. 3, right, shows the resulting A matrices for the scaled and translated WaLRUS.[1]

Some recent works [31, 32] has conceptually connected the use of wavelets and SSM-based models (namely Mamba). These efforts are fundamentally distinct from ours in that they perform a multi-resolution analysis on the input to the model only. No change is made to the standard Mamba SSM layer.

This work, on the other hand, is the first to challenge the ubiquity of the Legendre-based SSM, and present alternative wavelet-based machinery for the core of powerful models like Mamba. WaLRUS could be used as a drop-in replacement for any existing SSM-based framework. However, before simply substituting a part in a larger system, we must first justify how and why a different SSM can improve performance. This paper presents a tool that stands alone as an online function approximator, and also provides a foundational building block for future integration in SSM-based models.

[1] Code to generate the matrices is available at the following repository: `https://github.com/echbaba/walrus`.

3.1 Redundancy of the wavelet frame and size of the SSM

In contrast to orthonormal bases, redundant frames allow more than one way to represent the same signal. This redundancy arises from the non-trivial null space of the associated frame operator, meaning that multiple coefficient vectors can yield the same reconstructed function. Although the representation is not unique, it is still perfectly valid, and this flexibility offers several key advantages in signal processing. In particular, redundancy can improve robustness to noise, enable better sparsity for certain signal classes, and enhance numerical stability in inverse problems [33–35].

We distinguish between the total number of frame elements N_{full} and the effective dimensionality N_{eff} of the subspace where the meaningful representations reside. In other words, while the frame may consist of N_{full} vectors, the actual information content lies in a lower-dimensional subspace of size N_{eff}. This effective dimensionality can be quantified by analyzing the singular-value spectrum of the frame operator [29, 33].

For the WaLRUS SSMs described in this work, we first derive $A_{N_{\text{full}}}$ using all elements of the redundant frame. We then diagonalize A and reduce it to a size of N_{eff}. This ensures that different frame choices, whether orthonormal or redundant, can be fairly and meaningfully compared in terms of computational cost, memory usage, and approximation accuracy. The exact relationship between the wavelet frame and the resulting N_{eff} of the A matrix depends not only on the overlap of the shifts in the frame, but also on the type (and order) of chosen wavelet, and number of scales. Determining the "optimal" overlap or N_{eff} is application-specific and an area for future research.

3.2 Computational complexity of WaLRUS

For a sequence of length L, scaled-SaFARi has $O(N^3L)$ complexity due to solving an N-dimensional linear system at each step, while translated-SaFARi can reuse matrix inverses, and thus has $O(N^2L)$ complexity, assuming no diagonalization [17]. When the state matrix A is diagonalizable, the complexity reduces to $O(NL)$ and can further accelerate to $O(L)$ with parallel processing on independent scalar SSMs.

We observe that each of the scaled and translated WaLRUS SSMs we implemented, regardless of dimension, were stably diagonalizable. Further research is required to determine whether Daubechies wavelets will always yield diagonalizable SSMs. Legendre-based SSMs, on the other hand, are not stably diagonalizable [9]. Although [9] proposed a fast sequential HiPPO-LegS update to achieve $O(NL)$ complexity, [17] showed that it cannot be parallelized to $O(L)$. Moreover, no efficient sequential update exists for HiPPO-LegT, leaving Legendre-based SSMs at a disadvantage during inference when sequential updates are needed.

As sequence length increases, step-wise updates become a bottleneck, especially during training when the entire sequence is available upfront. This can be mitigated by using convolution kernels instead of sequential updates. Precomputing the convolution kernel and applying it via convolution accelerates computation, leveraging GPU-based parallelism to achieve $O(\log L)$ run-time complexity for diagonalizable SSMs. This optimization is feasible for both WaLRUS and Fourier-based SSMs. Although Legendre-based SSMs can attain similar asymptotic complexity through structured algorithms [10, 12], their nondiagonal nature prevents decoupling into N independent SSMs.

3.3 Representation errors in the translated WaLRUS

Truncated representations in SSMs inevitably introduce errors, as discarding higher-order components limits reconstruction fidelity [17]. SaFARi only investigated these errors for scaled SSMs, leaving their approximation accuracy unquantified. Visualizing the convolution kernels generated by different SSMs offers some insight into the varying performance of different SSMs on the function approximation task. An "ideal" kernel would include a faithful representation for each element of the basis or frame from $T = 0$ to $T = W$, where W is the window width, and it would contain no non-zero elements between W and L. However, certain bases generate kernels with warping issues, as illustrated in Fig. 4.

The HiPPO-LegT kernel loses coefficients due to warping within the desired translating window (see areas B and C of Fig. 4). For higher degrees of Legendre polynomials, the kernel exhibits an all-zero region at the beginning and end of the sliding window. This implies that high-frequency information in the input is not captured at the start or end of the sliding window, and the extent of this dead zone

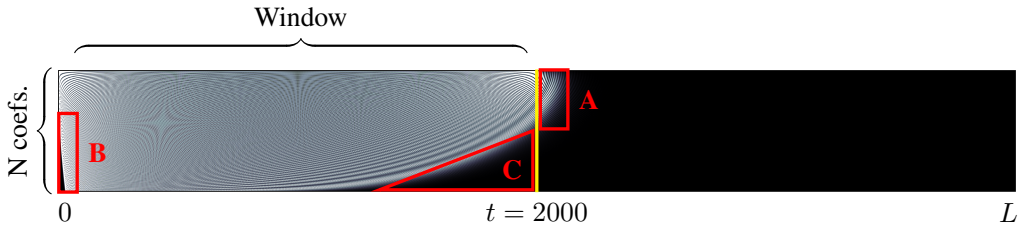

Figure 4: The kernel generated by HiPPO-LegT with window size $W = 2000$ and representation size $N = 500$. Three key non-ideal aspects of the kernel are noticeable. **A)** poor localization due to substantial non-zero values outside W, **B)** coefficient loss from at bottom left of the kernel, and **C)** coefficient loss at the bottom right of the kernel for $t \in (1500, 2000)$.

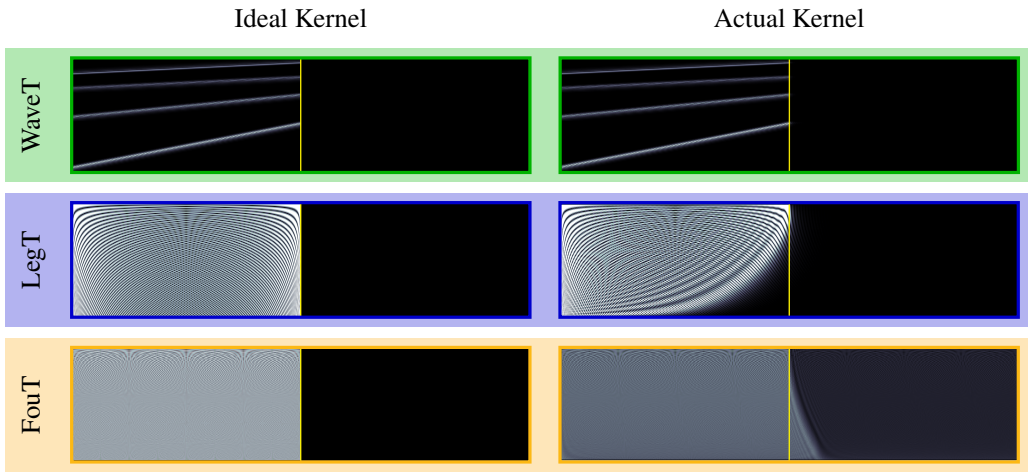

Figure 5: **Left:** The ideal kernels, which yield zero representation error, are shown for Translated-WaLRUS (using the D22 wavelet), HiPPO-LegT, and HiPPO-FouT. **Right:** The corresponding kernels generated by the translated models are presented for comparison. WaveT has superior localization within the window of interest compared to HiPPO-LegT and HiPPO-FouT.

increases with higher frequencies. The translated Fourier kernel primarily suffers from the opposite problem: substantial nonzero elements outside the kernel window indicate that LegT struggles to effectively "forget" historical input values. Thus contributions from input signals outside the sliding window appear as representation errors. LegT also has this problem, to a lesser extent–see area A of Fig. 4 for a closer view of the kernel.

A visual inspection of Fig. 5 reveals that the translated-WaLRUS kernel closely matches the idealized version, whereas both FouT and LegT exhibit significant errors in their computed kernels. We emphasize that the issues observed with LegT and FouT arise from inherent limitations of the underlying SSMs themselves and are not due to the choice of input signal classes.

4 Experiments

The following section deploys the WaLRUS SSM on synthetic and real signals for the task of function approximation, comparing its performance with extant models in the literature. We will evaluate performance in MSE as well as their ability to track important signal features like singularities, and show that using WaLRUS can have an edge over the state-of-the-art polynomial-based SSMs.

To benchmark WaLRUS against state-of-the-art SSMs, we implement two variants: *Scaled-WaLRUS* and *Translated-WaLRUS*, which we will call WaveS and WaveT respectively, following HiPPO's convention. These models are compared against the top-performing HiPPO-based SSMs. Further details on the wavelet frames used in each experiment are provided in Appendix A.2.4, and code can be found at `https://github.com/echbaba/walrus`.

We conduct experiments on the following datasets:

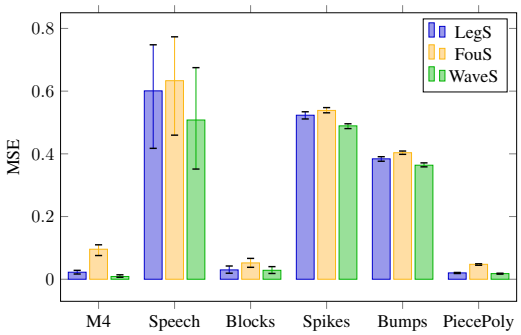

Figure 6: Comparing reconstruction MSE between WaveS, LegS, and FouS. Error bars represent the first and third quantile of MSE. WaveS produces the lowest MSE in each dataset.

Dataset	LegS	FouS	WaveS
M4	0%	0.47%	**99.53%**
Speech	4.25%	0%	**95.75%**
Blocks	0%	0%	**100%**
Spikes	0%	0%	**100%**
Bumps	0%	0%	**100%**
Piecepoly	1.00%	0%	**99.00%**

Table 1: Percent of tests where each basis had the lowest overall MSE.

M4 Forecasting Competition [36]: A diverse collection of univariate time series with varying sampling frequencies taken from domains such as demographic, finance, industry, macro, micro, etc.

Speech Commands [37]: A dataset of one-second audio clips featuring spoken English words from a small vocabulary, designed for benchmarking lightweight audio recognition models.

Wavelet Benchmark Collection [38]: A synthetic benchmark featuring signals with distinct singularity structures, such as Bumps, Blocks, Spikes, and Piecewise Polynomials. We generate randomized examples from each class, with further details and visualizations provided in Appendix A.2.2.

4.1 Comparisons among frames

We note that no frame is universally optimal for all input classes, as different classes of input signals exhibit varying decay rates in representation error. However, due to the superior localization and near-optimal error decay rate of wavelet frames, wavelet-based SSMs consistently show an advantage over Legendre and Fourier-based SSMs across a range of real-world and synthetic signals. These experiments position WaLRUS as a powerful and adaptable approach for scalable, high-fidelity signal representation.

4.1.1 Experimental setup

The performance of SSMs in online function approximation can be evaluated several ways. One metric is the mean squared error (MSE) of the reconstructed signal compared to the original. In the following sections, we compare the overall MSE for SSMs with a scaled measure, and the running MSE for SSMs with a translated measure.

Additionally, in some applications, the ability to capture *specific features* of a signal may be of greater interest than the overall MSE. As an extreme case, consider a signal that is nearly always zero, but contains a few isolated spikes. If our estimated signal is all zero, then the MSE will be small, but all of the information of interest has been lost.

In all the experiments, we use equal SSM sizes N_{eff}, as described in Sec. 3.1.

4.1.2 Function approximation with the scaled measure

In this experiment, we construct Scaled-WaLRUS, HiPPO-LegS, and HiPPO-FouS with equal effective sizes (see Appendix A.2.4). Frame sizes are empirically selected to balance computational cost and approximation error across datasets.

Fig. 6 shows the average MSE across random instances of multiple datasets. Not only is the average MSE lowest for WaLRUS for all datasets, but even where there is high variance in the MSE, all methods tend to keep the same *relative* performance. That is, the overlap in the error bars in Fig. 6 does not imply that the methods are indistinguishable; rather, for a given instance of a dataset, the MSE across all three SSM types tends to shift together, maintaining the MSE ordering WaveS <

	Dataset:	Spikes			Bumps		
	Basis/Frame:	Legendre	Fourier	Wavelets	Legendre	Fourier	Wavelets
Scaled	Peaks missed	2.5%	0.62%	**0%**	0.29%	0.30%	**0%**
	False peaks	1.6%	1.6%	**0.01%**	0.3%	1.9%	**0%**
	Instance-wise wins	76%	92.9%	**100%**	97.1%	96.9%	**100%**
	Relative amplitude error	16.2%	11.8%	**5.5%**	12.4%	16.2%	**6.5%**
	Average displacement	18.8	32.0	**10.0**	12.7	33.7	**7.1**
Translated	Peaks missed	6.4%	13.0%	**0.27%**	1.12%	29.76%	**0.08%**
	False peaks	1.1%	**0.05%**	0.22%	0.43%	0.28%	**0.20%**
	Instance-wise wins	36.9%	13.65%	**99.95%**	85.1%	0.2%	**100%**
	Relative amplitude error	19.6%	28.4%	**3.5%**	6.9%	28.4%	**2.5%**
	Average displacement	6.0	5.4	**4.3**	5.5	5.8	**4.8**

Table 2: Performance comparison of WaLRUS-Wavelets, HiPPO-Legendre, and HiPPO-Fourier for peak detection with the translated measure. WaLRUS shows a significant advantage in successfully remembering singularities over HiPPO SSMs.

LegS < FouS. To highlight this result, the percentage of instances where each SSM had the best performance is also provided in Table 1.

The representative power of WaLRUS is attributed to its ability to minimize truncation and mixing errors by selecting frames that capture signal characteristics with higher fidelity. See [17] for further details.

4.1.3 Peak detection with the scaled measure

In this experiment, we aim to detect the locations of random spikes in input sequences using Scaled-WaLRUS, FouS, and LegS, all constructed with an equal sizes. We generate random spike sequences, add Gaussian noise (SNR = 0.001), and compute their representations with Daubechies wavelets, Legendre polynomials, and Fourier series. The reconstructed signals are transformed into wavelet coefficients, and spike locations are identified following the method in [30].

To evaluate performance, we compare the relative amplitude and displacement of detected spikes with their ground truth (see Fig.7). This process is repeated for 1000 random sequences, each containing 10 spikes. Table 2 summarizes the average number of undetected spikes for each SSM and the instance-wise win percentage, representing the number of instances where each SSM had fewer or equal misses peaks than the other SSMs. Note that these percentages do not sum to 100, as some instances result in identical spike detection across all models.

As shown in Table 2, WaveS misses significantly fewer spikes than FouS and LegS, with lower displacement errors and reduced amplitude loss. Figure 1 illustrates an example where WaLRUS successfully captures closely spaced spikes that are missed by LegS and FouS, demonstrating its superior time resolution.

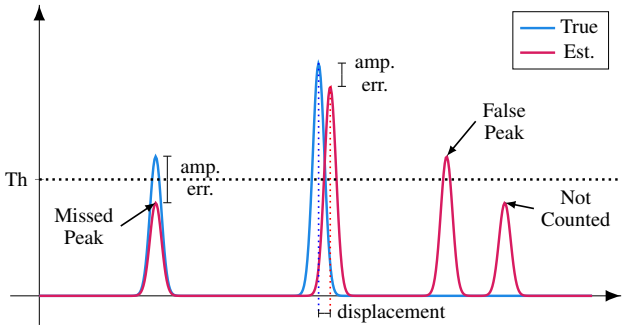

Figure 7: Illustration of the metrics to evaluate performance of SSMs on different datasets in Table 2.

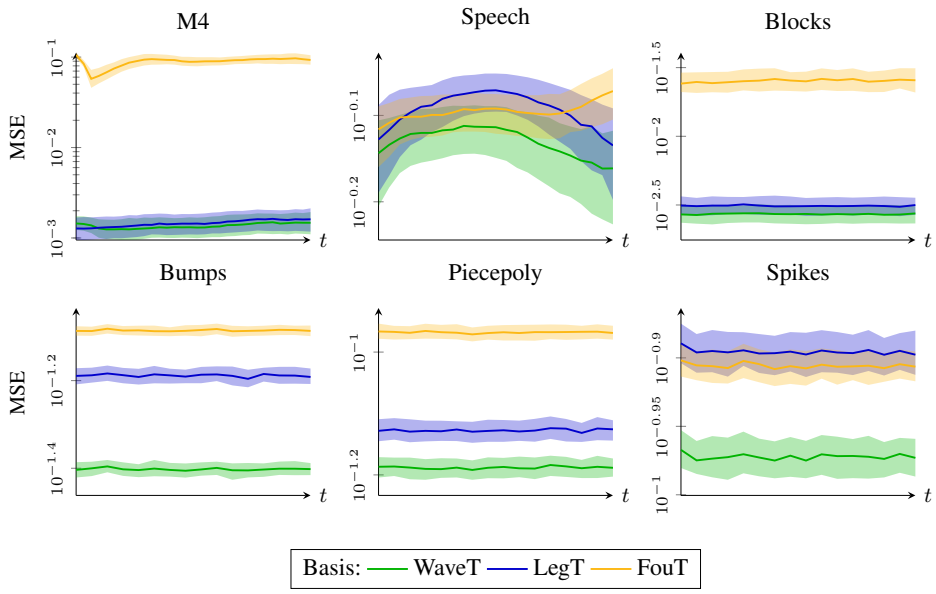

Figure 8: For each dataset, the median and $(0.4, 0.6)$ quantile of running reconstruction MSE across different instances is demonstrated in different colors for WaveT, LegT, and FouT. WaveT captures information in the input signals with a higher fidelity than LegT and FouT.

4.1.4 Function approximation with the translated measure

In this experiment, we construct WaveT, LegT, and FouT SSMs, all with equal effective sizes (see Appendix A.2.4). The chosen effective sizes are smaller than those we used for the scaled measure since the translated window contains lower frequency content within each window, making it possible to reconstruct the signal with smaller frames. Then, for each instance of input signal, the reconstruction MSE at each time step is calculated and plotted in Fig. 8.

For each input signal instance, we compute the running MSE at each time step, as shown in Fig. 8. This plot represents how the MSE evolves over time across multiple instances, providing a comparison of running MSEs for each SSM. The results demonstrate that Translated-WaLRUS consistently achieves slightly better fidelity than LegT and significantly outperforms FouT across all datasets.

As discussed in Section 3.3, the reconstruction error stems from two main factors: (1) non-idealities in the translated SSM kernel, affecting its ability to retain relevant information within the window while effectively forgetting data outside it (see Fig. 4), and (2) the extent to which these fundamental non-idealities are activated by the input signal. For example, signals with large regions of zero values are less impacted by kernel inaccuracies, as the weights outside the kernel contribute minimally to reconstruction.

WaveT achieves a modest, and in some cases negligible MSE improvement over LegT (e.g., M4 and Blocks). However, the kernel-based limitations highlighted in Section 3.3 may have a more pronounced effect on longer sequences or different datasets.

4.1.5 Peak detection with the translated measure

In this experiment, we evaluate the ability of WaveT, FouT, and LegT to retain information about singularities in signals, following the setup in Section 4.1.3, but with a translated SSM. We generate 2,000 random sequences, each containing 20 spikes. The average number of undetected spikes for each SSM, along with instance-wise win percentages, is reported in Table 2. As in the scaled measure experiment, the percentages do not sum to 100 due to ties across SSMs. Table 2 shows that WaveT consistently outperforms FouT and LegT, with fewer missed peaks, reduced displacement, and less amplitude loss.

5 Limitations

In this work we have implemented only one type of wavelet (Daubechies-22), as our purpose is to introduce practical and theoretical reasons to replace polynomial SSMs with wavelet SSMs. Other wavelets (biorthogonal, coiflets, Morlets, etc.) could also be used, with some caveats. First, we require a differentiable frame [17], so nondifferentiable wavelets like Haar wavelets or other lower-order Daubechies and Coiflets cannot be used with this method. Second, the redundancy of the frame (and the resulting N_{eff} of the A matrix) depends on the shape of the wavelet's function and the chosen shifts and scales of this function. Other wavelet types, and other choices of shift and scale, may exhibit better or worse performance and dimensionality reduction, and this is an important question for future work.

Additionally, we emphasize that the choice of frame is application-dependent. If the signal is known to be smooth and periodic, a wavelet-based SSM is not likely to outperform a Fourier-based SSM, for example. The introduction of WaLRUS is not intended to be a one-size-fits-all model, but rather a broadly-applicable tool that combines compressive online function-approximation SSMs with the expressive power of wavelets.

6 Conclusions

We have demonstrated in this paper how function approximation with SSMs, initially proposed by [9] and subsequently extended to general frames, can be improved using wavelet-based SSMs. SSMs constructed with wavelet frames can provide higher fidelity in signal reconstruction than the state-of-the-art Legendre and Fourier-based SSMs over both scaled and translated measures. Future work will explore alternate wavelet families, and the trade-offs in effective size, frequency space coverage, and representation capabilities of different frames.

Moreover, since the Legendre-based HiPPO SSM forms the core of S4 and Mamba, and WaLRUS provides a drop-in replacement for HiPPO, WaLRUS could be used to initialize SSM-based machine learning models–potentially providing more efficient training. As AI becomes ubiquitous, and the demand for computation explodes, smarter and more task-tailored ML architectures can help mitigate the strain on energy and environmental resources.

Acknowledgments

Special thanks to T. Mitchell Roddenberry for fruitful conversations and insights. This work was supported by NSF grants CCF-1911094, IIS-1838177, and IIS-1730574; ONR grants N00014-18-12571, N00014-20-1-2534, N00014-18-1-2047, and MURI N00014-20-1-2787; AFOSR grant FA9550-22-1-0060; DOE grant DE-SC0020345; and ONR grant N00014-18-1-2047. Additional support was provided by a Vannevar Bush Faculty Fellowship, the Rice Academy of Fellows, and Rice University and Houston Methodist 2024 Seed Grant Program.

References

[1] Nikola Zubić, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[2] Sina Alemohammad, Hossein Babaei, Randall Balestriero, Matt Y. Cheung, Ahmed Imtiaz Humayun, Daniel LeJeune, Naiming Liu, Lorenzo Luzi, Jasper Tan, Zichao Wang, and Richard G. Baraniuk. Wearing a mask: Compressed representations of variable-length sequences using recurrent neural tangent kernels. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2950–2954, 2021.

[3] 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. In *Advances in Neural Information Processing Systems*, 2022.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[5] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[6] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[7] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, 2020.

[10] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

[11] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[12] Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Re. How to train your HiPPO: State space models with generalized orthogonal basis projections. In *International Conference on Learning Representations*, 2023.

[13] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, 2024.

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

[15] Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023.

[16] Ramin Hasani, Mathias Lechner, Tsun-Hsuan Wang, Makram Chahine, Alexander Amini, and Daniela Rus. Liquid structural state-space models. In *International Conference on Learning Representations*, 2023.

[17] Hossein Babaei, Mel White, Sina Alemohammad, and Richard G Baraniuk. SaFARi: State-space models for frame-agnostic representation. *arXiv preprint arXiv:2505.08977*, 2025.

[18] Ingrid Daubechies. Ten lectures on wavelets. *SIAM Press*, 1992.

[19] Alan V Oppenheim. *Discrete-Time Signal Processing*. Pearson, 1999.

[20] Agostino Abbate, Casimer DeCusatis, and Pankaj K Das. *Wavelets and Subbands: Fundamentals and Applications*. Springer, 2012.

[21] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[22] John G Proakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Pearson, 2001.

[23] Paolo Prandoni and Martin Vetterli. *Signal Processing for Communications*. EPFL Press, 2008.

[24] Guofeng Zhang, Tongwen Chen, and Xiang Chen. Performance recovery in digital implementation of analogue systems. *SIAM Journal on Control and Optimization*, 45(6):2207–2223, 2007.

[25] Patrice Abry, Patrick Flandrin, and Murad S. Taqqu. Self-similarity and long-range dependence through the wavelet lens. In Paul Doukhan, George Oppenheim, and Murad S. Taqqu, editors, *Theory and Applications of Long-Range Dependence*, pages 527–556. Birkhäuser, 2003.

[26] Alfred Haar. *Zur Theorie der Orthogonalen Funktionensysteme*. PhD thesis, University of Göttingen, 1909.

[27] A. Grossmann and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736, 1984.

[28] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

[29] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3rd edition, 2008.

[30] Stephane Mallat and Wen Liang Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, 1992.

[31] Tianpei Zhang, Yiming Zhu, Jufeng Zhao, Guangmang Cui, and Yuchen Zheng. Exploring state space model in wavelet domain: An infrared and visible image fusion network via wavelet transform and state space model, 2025.

[32] Wenbin Zou, Hongxia Gao, Weipeng Yang, and Tongtong Liu. Wave-mamba: Wavelet state space model for ultra-high-definition low-light image enhancement. In *Proceedings of the 32nd ACM International Conference on Multimedia*, page 1534–1543. Association for Computing Machinery, 2024.

[33] O Christensen. *An Introduction to Frames and Riesz Bases*. Birkhauser, 2003.

[34] Karlheinz Gröchenig. *Foundations of Time-Frequency Analysis*. Springer, 2001.

[35] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[36] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020.

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

[38] David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

A Appendix

A.1 SaFARi derivation for arbitrary frame

Where HiPPO [9] provided closed-form solutions to construct A and B for a few polynomial bases, SaFARi [17] introduced a method to build A and B from any arbitrary frame. The derivations provided below follow [17], and are given here as convenient reference for the reader.

Take a signal f and frame ψ. To get a vector of weights representing a signal on a basis, we use the inner product:

$$c_n = \int f(t)\overline{\phi(t)}dt \tag{A.1}$$

So at some time T, we scale the magnitude of $f(t)$ and stretch the basis to match the length of f:

$$c_n(T) = \int_{t_0}^{T} f(t)\left(\frac{1}{T-t_0}\right)\overline{\phi\left(\frac{t-t_0}{T-t_0}\right)}dt \tag{A.2}$$

We are actually interested in the **change** in c. We will take the partial derivative with respect to T, since the coefficients update at each new time T. Call the start time t_0: this is 0 for the scaling case, and t_0 varies with the windowed case. If we call the size of the window θ, then $t_0 = T - \theta$. The derivation below will be a generic version, then we will separate the two cases.

$$\frac{d}{dT}c_n(T) = \frac{d}{dT}\int_{t_0}^{T} f(t)\left(\frac{1}{T-t_0}\right)\overline{\phi\left(\frac{t-t_0}{T-t_0}\right)}dt \tag{A.3}$$

We note that this is the partial derivative of an integral bounded by two variables. Thus we call on Leibniz' integration rule and find:

$$\begin{aligned}\frac{d}{dT}c_n(T) = f(T)\left(\frac{1}{T-t_0}\right)\overline{\phi(1)}\frac{\delta}{\delta T}(T) - f(t_0)\left(\frac{1}{T-t_0}\right)\overline{\phi(0)}\frac{\delta}{\delta T}(t_0) \\ + \int_{t_0}^{T} f(t)\underbrace{\frac{\delta}{\delta T}\left[\left(\frac{1}{T-t_0}\right)\overline{\phi\left(\frac{t-t_0}{T-t_0}\right)}\right]}_{\overline{h(t)}}dt\end{aligned} \tag{A.4}$$

Some manipulation of the $h(t)$ term yields:

$$\begin{aligned}h(t) = \left(\frac{1}{T-t_0}\right)\left[-\frac{\delta(t_0)}{\delta T}\left(\frac{1}{T-t_0}\right)\phi'\left(\frac{t-t_0}{T-t_0}\right) - \left(1-\frac{\delta(t_0)}{\delta T}\right)\left(\frac{t-t_0}{T-t_0}\right)\phi'\left(\frac{t-t_0}{T-t_0}\right)\right] \\ - \left(\frac{1}{T-t_0}\right)\left[\left(\frac{1-\frac{\delta(t_0)}{\delta T}}{T-t_0}\right)\phi\left(\frac{t-t_0}{T-t_0}\right)\right]\end{aligned} \tag{A.5}$$

Our h(t) term now has the derivative of our basis (ϕ') in it, but we'd like to be able to combine combine terms with ϕ. Therefore we can make a mapping from $\phi' \to \phi$ using the dual, $\widetilde{\phi}$:

$$\phi'\left(\frac{t-t_0}{T-t_0}\right) = \underbrace{\left\langle \phi'\left(\frac{t-t_0}{T-t_0}\right), \widetilde{\phi}\left(\frac{t-t_0}{T-t_0}\right)\right\rangle}_{P}\phi\left(\frac{t-t_0}{T-t_0}\right) \tag{A.6}$$

Likewise:

$$(t-t_0)\phi'\left(\frac{t-t_0}{T-t_0}\right) = \underbrace{\left\langle (t-t_0)\,\phi'\left(\frac{t-t_0}{T-t_0}\right), \widetilde{\phi}\left(\frac{t-t_0}{T-t_0}\right)\right\rangle}_{P_t}\phi\left(\frac{t-t_0}{T-t_0}\right) \tag{A.7}$$

This lets us do another simplification of $h(t)$, and group all the functions of ϕ. Let's also call $T - t_0 = \theta$ to save some space.

$$h(t) = \frac{1}{\theta}\phi\left(\frac{t-t_0}{T-t_0}\right)\left[-\frac{\delta(t_0)}{\delta T}\frac{1}{\theta}P - \left(1 - \frac{\delta(t_0)}{\delta T}\right)\frac{1}{\theta}P_t - \frac{1}{\theta}\left(1 - \frac{\delta(t_0)}{\delta T}\right)\right] \tag{A.8}$$

Now we can return to Eq. A.4. P is not a function of t, so it can be moved outside the integral. For the measures we are looking at, $\frac{d}{dT}$ is always constant with respect to t – it is either 0 or 1. We can substitute then group as follows:

$$\begin{aligned}\frac{d}{dT}c_n(T) = &\left(\frac{1}{T-t_0}\right)\left[f(T)\overline{\phi(1)} - f(t_0)\overline{\phi(0)}\frac{\delta}{\delta T}(t_0)\right] + \\ &\left(\frac{1}{T-t_0}\right)\left[-\frac{\delta(t_0)}{\delta T}\overline{P} - \left(1 - \frac{\delta(t_0)}{\delta T}\right)\overline{P_t} - \left(1 - \frac{\delta(t_0)}{\delta T}\right)\right]\underbrace{\int_{t_0}^{T} f(t)\left(\frac{1}{T-t_0}\right)\overline{\phi\left(\frac{t-t_0}{T-t_0}\right)}dt}_{c(T)}\end{aligned} \tag{A.9}$$

Noting that the final term in this equation contains Eq. A.2, we can simplify further:

$$\begin{aligned}\frac{d}{dT}c_n(T) = &\left(\frac{1}{T-t_0}\right)\left[f(T)\overline{\phi(1)} - f(t_0)\overline{\phi(0)}\frac{\delta(t_0)}{\delta T}\right] + \\ &\left(\frac{1}{T-t_0}\right)c(T)\left[\frac{-\delta(t_0)}{\delta T}\overline{P} - \left(1 - \frac{\delta(t_0)}{\delta T}\right)\overline{P_t} - \left(1 - \frac{\delta(t_0)}{\delta T}\right)\right]\end{aligned} \tag{A.10}$$

Unfortunately, we still have a term $f(t_0)$ that we don't have access to; this is the value of the function at the start of our window. But we have not stored this value; that would defeat the point of an online update in the first place. Instead, we will *approximate it* based on our current coefficient vector and our known basis.

$$\begin{aligned}c &= \langle \phi, f \rangle \\ f &= \langle \widetilde{\phi}, c \rangle \\ f(t_0) &= \langle \widetilde{\phi}(0), c(T) \rangle\end{aligned}$$

We now have an update rule for c that depends only on the frame ϕ, the current value of $c(T)$, and the new information from the signal, $f(T)$:

$$\begin{aligned}\frac{d}{dT}c(T) = &\left(\frac{1}{T-t_0}\right)\left[f(T)\overline{\phi(1)} - \widetilde{\phi}(0)c(T)\overline{\phi(0)}\frac{\delta(t_0)}{\delta T}\right] \\ &- \left(\frac{1}{T-t_0}\right)\left[c(T)\left[\frac{\delta(t_0)}{\delta T}\overline{P} + \left(1 - \frac{\delta(t_0)}{\delta T}\right)\overline{P_t} + \left(1 - \frac{\delta(t_0)}{\delta T}\right)\right]\right]\end{aligned} \tag{A.11}$$

A.1.1 The scaled case

In the case of scaling, $t_0 = 0$ and $\frac{\delta}{\delta T}(t_0) = 0$.

$$\frac{d}{dT}c_n(T) = \left(\frac{1}{T}\right)\left[f(T)\overline{\phi(1)} - \widetilde{\phi}(0)c(T)\overline{\phi(0)}\frac{\delta(t_0)}{\delta T}\right] \tag{A.12}$$

$$- \left(\frac{1}{T}\right)c(T)\left[\frac{\delta(t_0)}{\delta T}\overline{P} + \left(1 - \frac{\delta(t_0)}{\delta T}\right)\overline{P_t} + \left(1 - \frac{\delta(t_0)}{\delta T}\right)\right] \tag{A.13}$$

$$\frac{d}{dT}c_n(T) = \left(\frac{1}{T}\right)f(T)\overline{\phi(1)} - \left(\frac{1}{T}\right)c(T)(\overline{P_t} + 1) \tag{A.14}$$

The A matrix acts on the coefficient vector c, and B acts on the current input, f(T). Expressed in matrix notation:

$$\frac{d}{dT}c_n(T) = -\frac{1}{T}\underbrace{(\overline{P_t} + I)}_{A}c(T) + \frac{1}{T}\underbrace{\overline{\phi(1)}}_{B}f(T) \tag{A.15}$$

Equivalently,

$$\frac{d}{dT}c_n(T) = -\frac{1}{T}\underbrace{\left(\left\langle \widetilde{\phi}\left(\frac{t}{T}\right), t\phi\left(\frac{t}{T}\right)'\right\rangle + I\right)}_{A}c(T) + \frac{1}{T}\underbrace{\overline{\phi(1)}}_{B}f(T) \tag{A.16}$$

A.1.2 The translated case

Now $T - t_0 = \theta$ where θ is the window size, and $\frac{\delta}{\delta T}(t_0) = 1$. Following the same procedure as the previous section:

$$\frac{d}{dT}c_n(T) = \left(\frac{1}{\theta}\right)f(T)\overline{\phi(1)} - \left(\frac{1}{\theta}\right)c(T)\left[\widetilde{\phi}(0)\overline{\phi(0)} + \overline{P}\right] \tag{A.17}$$

$$\frac{d}{dT}c_n(T) = -\frac{1}{\theta}\underbrace{(\overline{P} + \widetilde{\phi}(0)\overline{\phi(0)})}_{A}c(T) + \frac{1}{\theta}\underbrace{\overline{\phi(1)}}_{B}f(T) \tag{A.18}$$

$$\frac{d}{dT}c_n(T) = -\frac{1}{\theta}\underbrace{\left(\left\langle \widetilde{\phi}\left(\frac{t}{\theta}\right), \phi'\left(\frac{t}{\theta}\right)\right\rangle + \widetilde{\phi}(0)\overline{\phi(0)}\right)}_{A}c(T) + \frac{1}{\theta}\underbrace{\overline{\phi(1)}}_{B}f(T) \tag{A.19}$$

A.2 Experiments

A.2.1 Datasets

In this paper, we conducted our experiments on these datasets:

M4 forecasting competition: The M4 forecasting competition dataset [36] consists of 100,000 univariate time series from six domains: demographic, finance, industry, macro, micro, and other. The data covers various frequencies (hourly, daily, weekly, monthly, quarterly, yearly) and originates from sources like censuses, financial markets, industrial reports, and economic surveys. It is designed to benchmark forecasting models across diverse real-world applications, accommodating different horizons and data lengths. We test on 3,000 random instances.

Speech commands: The speech commands dataset [37] is a set of 400 audio files, each containing a single spoken English word or background noise with about one second duration. These words are from a small set of commands, and are spoken by a variety of different speakers. This data set is designed to help train simple machine learning models.

Wavelet benchmark collection: Donoho [38] introduced a collection of popular wavelet benchmark signals, each designed to capture different types of singularities. This benchmark includes well-known signals such as Bumps, Blocks, Spikes, and Piecewise Polynomial. Following this model, we synthesize random signals belonging to the classes of bumps, blocks, spikes, and piecewise polynomials. Details and examples of these signals can be found in Appendix A.2.2.

A.2.2 Wavelet Benchmark Collection

Donoho [38] introduced a collection of popular wavelet benchmark signals, each designed to capture different types of singularities. This benchmark includes well-known signals such as Bumps, Blocks, Spikes, and Piecewise Polynomial.

Following this model, we synthesize random signals belonging to the classes of bumps, blocks, spikes, and piecewise polynomials in our experiments to compare the fidelity of DaubS to legS and fouS, and also to compare the fidelity of DaubT to LegT and FouT.

Figure 9 demonstrates a random instance from each of of the classes of the signals that we have in our wavelet benchmark collection.

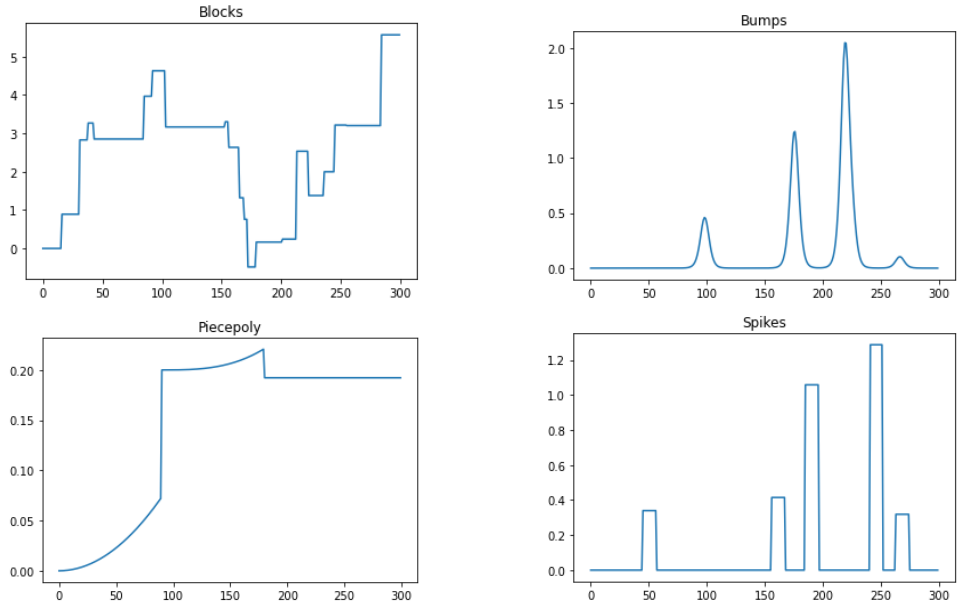

Figure 9: Instances of different signal types in the wavelet benchmark collection. **Top Left:** Blocks is a piecewise constant signal with random-hight sharp jumps placed randomly. **Top Right:** Bumps is a collection of random pulses where each pulse contains a cusp. **Bottom Left:** Piecepoly is a piecewise polynomial signal with discontinuity in the transition between different polynomial parts. **Bottom Right:** Spikes is a collection of rectangular pulses placed randomly with random positive hieght.

A.2.3 Description of metrics for 'Spikes' and 'Bumps' experiments

- **Peaks Missed** The number of true peaks in the signal is N_{tp}, and the number of detected peaks (that is, where the estimated signal surpasses an amplitude threshold Th_{amp}), is N_{dp}. $N_{dp|tp}$ is the number of detected peaks where a true peak is also within a displacement threshold (Th_{dis}) of the detected peak.

$$\text{Peaks Missed} = \left(1 - \frac{N_{dp|tp}}{N_{tp}}\right) \times 100\%$$

- **False Peaks** The metric False Peaks is calculated as the percentage of detected peaks that occurred when there was not a true peak within the displacement threshold. The number of detected peaks when there was no true peak is represented by $N_{dp|\overline{tp}}$.

$$\text{False Peaks} = \frac{N_{dp|\overline{tp}}}{N_{dp}} \times 100\%$$

- **Instance-wise Wins** In each of K time-series instances S, Each SSM m gets the instance win over other SSM models if it captures more true peaks than the other models.

$$\text{Instance-wise Wins} = \frac{1}{K}\sum_{k=1}^{K} w_k \times 100\%$$

$$w_k = \begin{cases} 1, & \text{if} \quad \text{Peaks Missed}_m \leq \text{Peaks Missed}_{\text{others}}, \\ 0, & \text{Ow} . \end{cases}$$

  In cases where multiple models achieve the same maximum, each tied model receives the credit for that time series instance. As a result, the sum of instance-wise wins for different SSMs may exceed 1.00.

- **Relative Amplitude Error** The relative amplitude error is calculated as the average percent error in the estimated amplitude of detected peaks, including false peaks.

$$\text{Relative Amplitude Error} = \frac{1}{N_{dp}} \left(\sum_{n=1}^{N_{dp|tp}} \frac{|A_{tp,n} - A_{dp|tp,n}|}{A_{tp,n}} \right) \times 100\%$$

- **Average Displacement** The location of a detected peak where a true peak was within a displacement threshold is given by $X_{dp|tp}$. The location of the true peak is denoted as X_{tp}.

$$\text{Average Displacement} = \frac{1}{N_{dp}} \sum_{n=1}^{N_{dp}} |X_{tp,n} - X_{dp|tp,n}|$$

A.2.4 Wavelet frames used for each experiment

Unlike HiPPO-based SSMs, which are fully characterized by their state size N, WaLRUS employs redundant wavelet frames that require additional parameters for identification. Once the wavelet frame is defined, the SaFARi framework constructs the unique A, B matrices corresponding to that frame. The key parameters for specifying a redundant wavelet frame in WaLRUS are as follows:

- **Wavelet Function:** Wavelet frames are built from a mother wavelet and a father wavelet, which capture high-frequency details and low-frequency approximations, respectively. Different families such as Daubechies, Morlet, Symlet, and Coifflet provide varied wavelet functions. For this work, we use the D22 wavelet from the Daubechies family.
- **L (Frame Length):** This represents the length of the wavelet frame. Increasing L increases numerical accuracy in the calculation of the A and B matrices at the cost of additional computation time. However, this initial computation need only be performed once, so it is best to choose a large L. For the experiments in this work, we set $L = 2^{19}$.
- ***Scale min* and N_{eff}:** The minimum scale sets the smallest feature of the signal that can be represented by the frame. This parameter should be chosen based on knowledge about the signal of interest and its component frequencies. Note that the size of the smallest feature is relative to the length of the signal under consideration, so this value may differ under scaling and translating measures.

  For wavelets, *scale min* also controls the effective rank, N_{eff}. Each new lower scale introduces a factor of two in the effective rank of the frame, owing to the additional shifted elements in each scale. Fig. 3 shows two scales, where there are 3 father wavelets (ϕ_0) and 3 coarse-scale mother wavelets (ψ_1). The next scale introduces 6 scaled and shifted mother wavelets (ψ_2), the next would include 12, and so on. Table 3 also illustrates this pattern, with *scale min* of 0 corresponding to N_{eff} of 2^6, *scale min* of -1 corresponding to N_{eff} of 2^7, and so on, with some margin of error for numerical accuracy and truncation.

  Our code includes another variable, *scale max*. Since smaller scales can also combine to represent larger scales, *scale max* in fact has no impact on N_{eff} (see [29] for further information). Fig. 10 demonstrates on an example implementation that varying scale max does not impact the size of N_{eff}. It is also easily shown that varying scale max results in the same diagonalized A; see our code supplement. Adding coarser scales can help improve numerical accuracy in the calculation of A, however. We do not include scale max in Table 3, but we do provide it in our code with each experiment for reproducibility.

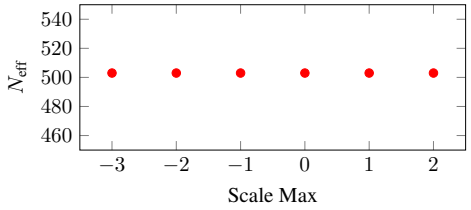

Figure 10: Effective Rank of WaLRUS A matrix with Scale Min=-3, shift=0.01

- **Shift:** At scale i, $2^{-i}m$ overlapping shifts are applied to the wavelets, where $0 < m \leq 1$ is a shift constant. Setting $m = 1$ corresponds to dyadic shifts. As our wavelet frames typically

Experiment	Basis/Measure	scale min	shift	N_{eff}
Scaled M4	WaveS	-3	0.01	501
	LegS	-	-	500
	FouS	-	-	500
Scaled Speech	WaveS	-5	0.01	1995
	LegS	-	-	1995
	FouS	-	-	1995
Scaled synthetic	WaveS	-3	0.01	501
	LegS	-	-	500
	FouS	-	-	500
Scaled peak detection	WaveS	0	0.01	65
	LegS	-	-	65
	FouS	-	-	65
Translated M4	WaveT	-1	0.01	128
	LegT	-	-	128
	FouT	-	-	128
Translated Speech	WaveT	-3	0.0025	500
	LegT	-	-	500
	FouT	-	-	500
Translated synthetic	WaveT	-1	0.01	128
	LegT	-	-	128
	FouT	-	-	128
Translated peak detection	WaveT	0	0.01	65
	LegT	-	-	65
	FouT	-	-	65

Table 3: Parameters for the redundant wavelet frame used by WaLRUS in different experiments. All of the above experiment share the parameters $L = 2^{19}$, and $\text{rcond} = 0.01$.

only contain a few dilation levels, using $m = 1$ can mean that the constructed set of vectors no longer satisfies the frame condition, and is lossy. We choose a small value (0.01 for most experiments), and tune this as needed.

- **rcond:** This parameter controls the numerical stability of the pseudo-inverse calculation for the dual frame. Singular values smaller than $\text{rcond} \times \sigma_{\max}$ are discarded during the inversion process to maintain numerical stability.

Note that all the above parameters are solely to identify the redundant wavelet frame, and that WaLRUS does not introduce any new parameters. Table 3 summarizes the settings for all experiments, alongside the SSM sizes for HiPPO-Legendre and HiPPO-Fourier.

A.2.5 Computational resources

Within the scope of this paper, no networks were trained and no parameters were learned. Only CPU resources were utilized, but speed could be improved with parallel resources on a GPU. Using WaLRUS to find representation has two different stages:

- Pre-computing: Computing SSM A matrices and diagonalizing them. This step can be computationally intensive, but need only be calculated once.
- Computation: Using SSM A matrices to find representations of signals.

For all our experiments except Scaled-Speech, the pre-computing stage takes less than 10 minutes. For scaled-speech, the pre-compute time is on the order of hours. Once the A matrices are computed and stored, run time is the same for all experiments.

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: Our abstract and introduction states that we introduce the use of wavelets in state-space models for online function representation, and show how these can outperform state-of-the-art polynomial models for certain data types. Section 3 describes the construction of wavelet-based SSMs, and section 4 experimentally supports our performance claims.

   Guidelines:

   - The answer NA 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 NA 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: Section 5 describes limitations, both in terms of what we have implemented in this work, as well as limitations in the use of our method.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All necessary theoretical background is given in Sec. 2 and full support for our results are in sections 3-4.

Guidelines:

- The answer NA 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: The Experiments section thoroughly describes what metrics were tested and how they were evaluated, as well as the publicly available datasets used. Scripts to replicate the experimental results are available at `https://github.com/echbaba/walrus`

Guidelines:

- The answer NA 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: Code and data are available at `https://osf.io/7kjcx/?view_only=5dc38b9776624deb9d1c0d8f88108658`

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the required information on both the datasets, and the exact experimental setting required to recreate the wavelet frame, are provided in the Appendix. This information can also be found in our code.

   Guidelines:

   - The answer NA 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: Error bars and quantiles are provided in figures 5 and 7, and explanations of their source are in the text and captions of the figures. Since MSE is not normally distributed, we chose to use quantiles and percentiles to reflect the distribution more accurately. We also provide tables 1 and 2 to describe additional nuances of the comparison data.

   Guidelines:

- The answer NA 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: Since our work did not involve any training, no GPU computation was necessary. More discussion is available in the Appendix (Sec. A.2.5).

   Guidelines:

   - The answer NA 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 conducted this research with integrity and reported our findings with honesty. The link to the Code of Ethics provided is broken, and so we have instead consulted this provisional copy of the document: `https://openreview.net/forum?id=zVoy8kAFKPr`.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: This work is a basic mathematical result that does not have a targeted end use. We do note in our conclusion that improved function approximators, like the one we present here, can reduce the computational resources required for training certain types of neural networks – resources that have recently become a major environmental concern.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This is a foundational and theoretical work that is primarily mathematical in nature: a compressive online approximation of time-series signals over a wavelet frame. The potential use cases for such a tool are similar in scope to that of a Fourier Transform; that is, it is too broad to responsibly hypothesize specific use cases or create guidelines.

Guidelines:

- The answer NA 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: the M4 dataset does not have a required license: `https://paperswithcode.com/dataset/m4`. The SpeechCommands dataset has a CC BY license, allowing for unrestricted use, with attribution to the author: `https://huggingface.co/datasets/google/speech_commands`. The four other data types we test on are generated by code that is made available with this paper, and based on [38].

Guidelines:

- The answer NA 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: An implementation of WaLRUS is provided with the code.

Guidelines:

- The answer NA 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: [NA]

Justification: There were no human subjects in this theoretical work.

Guidelines:

- The answer NA 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: [NA]

    Justification: There were no study participants in this theoretical work.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We have used LLMs only to assist in writing and polishing the grammar.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.