arXiv:2609.04134v2 [cs.LG] 7 Nov 2026

Prospective Coding Improves Learning in Deep Continuous-Time Recurrent Networks

Shivang Rawat[1,*] **Mirko Morello**[1] **Flaviano Morone**[1,2] **David J. Heeger**[1,2]

[1] Telepath
[2] New York University

[*]shivang@telepath.com

Abstract

Temporal integration gives continuous-time recurrent networks memory, but in deep stacks it also delays bottom-up signals and attenuates top-down errors. We compensate for this delay by making each layer's bottom-up input prospective, and we develop *Recursive Quadrature Filters* (RQFs), biologically motivated temporal band-pass filters that are a special case of diagonal state-space models (SSMs). Starting from an energy model, we derive the RQF dynamics and show that each RQF is a band-pass filter whose learnable parameters control its tuning frequency and bandwidth. We then make each layer's bottom-up input prospective using a parameter-free two-tap update that leaves the recurrent transition and parallel scan unchanged. We extend this correction to general diagonal SSMs and show that it mitigates depth-dependent gradient attenuation when temporal gradients are truncated, i.e., spatial-only backpropagation. We evaluate the performance of RQFs, S5, and ORGaNICs (a nonlinear gated RNN with built-in attentional modulation and recurrent normalization) trained using full backpropagation through time (BPTT) and spatial-only backpropagation. Under full BPTT, the prospective variants of these models match or outperform their non-prospective controls in every configuration. A non-residual width-32 six-layer RQF achieves $96.09 \pm 0.23\%$ accuracy on raw-audio Speech Commands with 31.9k parameters; a width-64 six-layer RQF achieves $83.36 \pm 1.71\%$ on the 16,384-step Path-X task. We also introduce selective RQFs (sRQFs), which make each filter's input gain and, optionally, its bandwidth and tuning frequency input-dependent while retaining the parallel scan. With the input gain alone, the same six-layer model achieves $86.31 \pm 0.51\%$ on Path-X; with all three input-dependent, a four-layer prospective sRQF solves Selective Copying ($99.83 \pm 0.07\%$). These results identify RQFs as a parameter-efficient recurrent substrate and prospective-input coding as an input-side correction for deep continuous-time recurrent networks.

1 Introduction

Biological neural circuits face a fundamental trade-off between memory and reactivity. A biological neuron is well modeled as a leaky integrator whose membrane time constant τ confers temporal context but also delays and attenuates fast input changes [1]. When stacked into an L-layer feedforward circuit, this filter cascades, so the effective relaxation time grows roughly as $L\tau$ [2]. The same trade-off is built into recurrent theories of cortical function [3], in which slow recurrent dynamics let the network settle into a representation that combines feedforward drive with prior expectation. This lag also creates a learning problem: when a top-down teaching signal is applied at the output layer, the activity it should instruct in early layers is already roughly $L\tau$ out of date. Signals that vary on timescales comparable to τ are therefore poorly served by the standard leaky cascade.

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

This paper contributes a recurrent substrate and an input-side correction. The substrate is the *Recursive Quadrature Filter* (RQF), a linear dynamical system inspired by neuroscience that is a special case of a diagonal state-space model (SSM), so it trains on sequences of tens of thousands of time steps with standard SSM algorithms. The input-side correction drives each layer with a prospective rather than an instantaneous bottom-up signal; it adds no parameters and applies to existing diagonal SSMs as well as to RQFs.

The biological motivation for this correction comes from *prospective* firing: under sinusoidally modulated somatic current injection, cortical pyramidal cells can fire at a phase that leads their input, and biophysical models explain this as a firing-rate advance relative to membrane voltage [4, 5]. Apical dendritic stimulation can likewise modulate the soma with little integration lag through dendritic resonance [6, 5]. Related prospective responses have been reported in retinal ganglion-cell populations, cerebellar Purkinje cells, and human muscle-spindle afferents [7–9]. Such phase leads can be modeled by applying the *prospective* (or look-ahead) operator [2, 10, 11]

$$\chi_\tau \;\equiv\; 1 + \tau \,\frac{\mathrm{d}}{\mathrm{d}t} \tag{1}$$

to the underlying voltage or firing rate. For a first-order low-pass filter, χ_τ is its exact differential inverse, so applying it cancels the filter's lag. When interpreted as a time-advance operator, however, it is accurate only to first order: expanding the unit time-shift in τ gives $u(t+\tau) \;=\; u(t) + \tau\,\dot{u}(t) + O(\tau^2) \;=\; \chi_\tau(u(t)) + O(\tau^2)$. Thus, χ_τ advances a smooth signal by one membrane time constant to first order in τ.

We apply this prospective operator in networks of RQFs, a class of biologically motivated complex-valued temporal band-pass filters [12]. We derive RQFs from an energy-based theory of cortical function [3], in which a state parameter tunes a compromise between feedforward sensory drive and an internally generated prior. Taking the prior to be the prospective image of the state under a phase-rotation model collapses the resulting dynamics into a scalar complex-valued band-pass filter with a learnable phase-prior frequency ω and a learnable bandwidth parameter γ (Section 2).

It has been suggested that the brain relies on a set of canonical neural microcircuits, across brain regions and modalities, implementing hierarchies of spatial band-pass filtering, output nonlinearities, and spatial pooling that extract and combine features [13–19], much as convolutional neural networks do. RQFs extend this paradigm to extract and combine features over time. Temporal band-pass RQFs have been used to model temporal-frequency selectivity and direction selectivity of neurons in primary visual cortex [12]. The encoding of sound in the cochlea and the auditory nerve can likewise be modeled as a bank of band-pass filters [20].

We prove that scalar RQFs are a special case of single-channel diagonal complex SSMs, so an RQF layer is a constrained diagonal SSM with biologically interpretable poles and inherits the parallel-scan and FFT-convolutional training algorithms used for SSMs [21, 22]. This distinguishes RQFs from recent prospective-coding cortical models: neuronal least-action [10] (NLA) and latent equilibrium [2] use prospective signals for local biological learning, while generalized latent equilibrium [11] (GLE) uses prospective and retrospective operators for online credit assignment in a feedforward architecture. RQFs instead provide recurrent temporal filters compatible with long-sequence SSM training.

We show, both theoretically and empirically, that deep RQF networks improve when each layer's bottom-up input drive is prospective. Prospective and instantaneous-input RQFs share the same prospective phase-rotation prior via recurrence; they differ only in whether the bottom-up drive is look-ahead coded. This input-side change is local to each layer, leaves the RQF parameters and state transition unchanged, admits a two-tap discretization applicable to diagonal SSMs (Section 3), and directly affects credit assignment under *spatial-only backpropagation*. Spatial-only backpropagation treats every earlier recurrent state as a stop-gradient variable, so it computes gradients only across layers, not over time. Under this regime, instantaneous inputs introduce an explicit $(h/\tau)^{L-\ell}$ attenuation (where h is the discretization time step), while prospective-input coding changes the discretization prefactor of each spatial hop from $O(h/\tau)$ to $O(1)$ as $h/\tau \to 0$.

We also introduce *selective RQFs* (sRQFs), an extension of RQFs with attention-like selection. Selective SSMs make the input matrix depend on the current input, so that the model controls which inputs enter its state [23]. One way to do this in an RQF layer is to modulate the input with an input-dependent gain: each sRQF filter multiplies its bottom-up drive by a positive gain computed from the preceding layer's output, analogous to the input gate of LSTMs and the input modulators of ORGaNICs [24, 25]. Akin to Mamba, which also makes its discretization step input-dependent [23],

an sRQF can additionally make the bandwidth γ and frequency ω parameters input-dependent, so that each filter's pole moves with the input. Because these quantities depend only on the preceding layer's output and not on the layer's own state, sRQFs remain diagonal linear recurrences, time-varying when γ and ω are selective, that can be computed efficiently using a parallel scan [26–28, 23], with either prospective or instantaneous input (Appendix F).

Contributions.

- *Architecture (machine learning).* We introduce RQFs, which are motivated by neuroscience but developed here for machine-learning applications. We prove that RQFs are a special case of diagonal SSMs (Section 2), making RQF layers compatible with the parallel-scan and FFT-convolution algorithms used for diagonal SSMs, thereby making RQF stacks capable of learning tasks such as the $16{,}384$-step Path-X. We also introduce sRQFs, which make the input gain, bandwidth γ, and frequency ω parameters input-dependent while retaining the parallel scan (Appendix F).
- *Prospective input (machine learning).* We replace each layer's instantaneous bottom-up drive with a prospective one and derive the corresponding discretization for any diagonal SSM (Section 3). The implementation adds no parameters, leaves the state transition and parallel scan untouched, and costs only a second input tap; under full BPTT, it yields higher mean test accuracy than the matched control in every RQF configuration.
- *Theory (computational neuroscience and machine learning).* We prove that under spatial-only backpropagation, instantaneous bottom-up inputs incur a geometric discretization-induced depth attenuation that prospective-input coding removes (Section 3 and Appendix C).
- *Performance (machine learning).* We evaluate prospective input on Speech Commands and Path-X with RQF stacks, S5, and the nonlinear ORGaNICs substrate [25], under full BPTT and spatial-only backpropagation, and compare RQFs with sRQFs on Path-X and Selective Copying (Section 4).

Scope of contributions. The RQF substrate and the input-side correction are evaluated as machine-learning contributions under standard BPTT; the cortical theory provides their motivation. The third contribution concerns spatial-only backpropagation, in which the recurrence is never differentiated through time. We adopt that regime because it is the one described by our derivation and isolates the effect of the input drive; it is a diagnostic, neither a prerequisite for the machine-learning contribution nor a recommended training procedure. Spatial-only backpropagation is temporally local, but exact error propagation across layers still uses conjugate-transposed forward weights and therefore retains the weight-transport problem [29]. We consequently describe RQFs as a biologically motivated circuit abstraction but do not suggest that backpropagation training is biologically plausible.

2 Recursive Quadrature Filters (RQFs)

The substrate for RQFs is a leaky integrator whose state relaxes toward a convex mixture of the external input and an internally predicted recurrent state:

$$\tau\,\dot{y} \;=\; -y + [\gamma\, z + (1-\gamma)\,\widehat{y}]\,, \tag{2}$$

where $y(t) \in \mathbb{C}$ is the filter state; $z(t) \in \mathbb{C}$ is the input drive; $\tau > 0$ is the base time constant; $\widehat{y}$ is the recurrent prediction; and $\gamma \in (0, 1)$ controls the balance between input tracking and recurrence. This convex blend mirrors the cortical energy model of Heeger [3]; Appendix A.1 derives Eq. (2) from that energy functional, with γ as the state parameter and $\widehat{y}$ as the prior drive.

The recurrent prediction $\widehat{y}$ in Eq. (2) requires a model of how y evolves locally. We take this model to be a *phase-rotation prior*: the undamped complex oscillator $\dot{y} = \iota\omega y$, the minimal generator on $\mathbb{C}$ of a single phase-prior frequency ω. This choice follows the internal-model principle: a system that prospectively tracks a structured signal should contain a model of that signal's dynamics [30]. Applying the prospective operator χ_τ from Eq. (1) to this prior gives

$$\widehat{y} \;\triangleq\; \chi_\tau(y)\big|_{\dot{y}=\iota\omega y} \;=\; (1 + \iota\omega\tau)\, y, \tag{3}$$

where $\chi_\tau(y(t))$ is causal because it uses only the instantaneous activity and its local time derivative.

Substituting Eq. (3) into Eq. (2) yields the *Recursive Quadrature Filter* (RQF), a scalar complex-valued temporal band-pass filter governed by

$$\tau\,\dot{y} \;=\; -y + \gamma\, z + (1-\gamma)(1 + \iota\omega\tau)\, y, \tag{4}$$

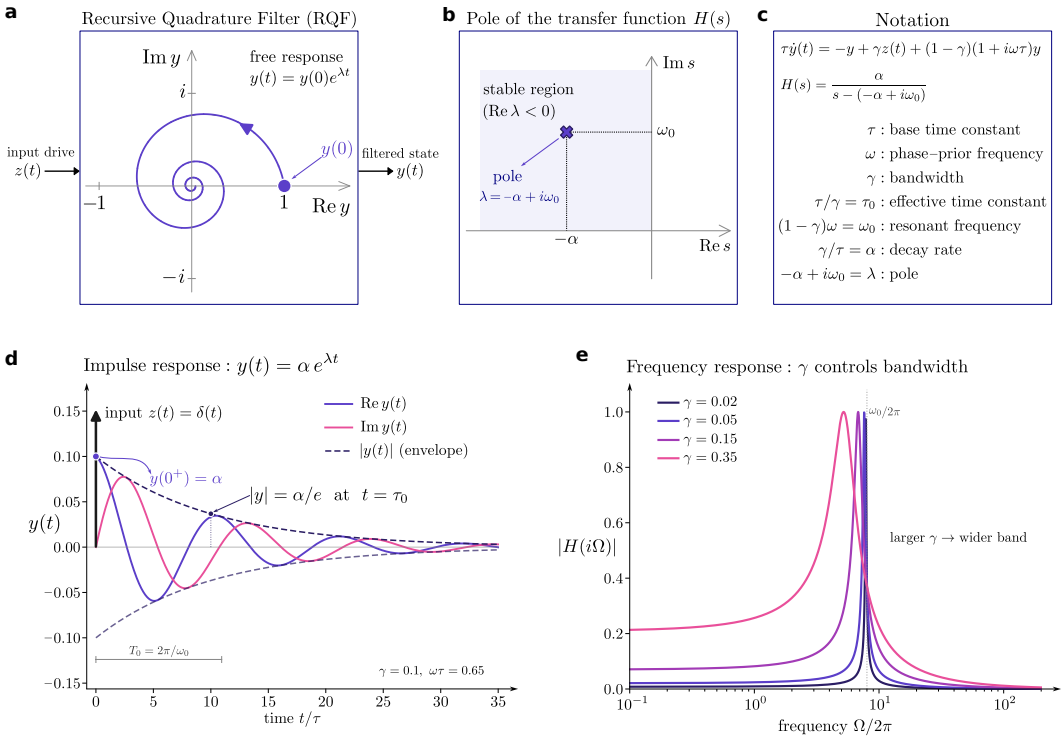


Figure 1: **The Recursive Quadrature Filter (RQF): pole, impulse response, and frequency response.** An RQF is a scalar complex-valued temporal filter with learnable phase-prior frequency ω and bandwidth parameter γ at a fixed base time constant τ, governed by Eq. (4). The derived quantities are the decay rate $\alpha = \gamma/\tau$, resonant frequency $\omega_0 = (1-\gamma)\omega$, effective time constant $\tau_0 = \tau/\gamma$, and pole $\lambda = -\alpha + \iota\omega_0$. **(a)** Free response of y in the complex plane, $y(t) = y(0)\, e^{\lambda t}$. **(b)** The pole lies in the stable region $\mathrm{Re}(\lambda) < 0$ for any $\gamma \in (0, 1)$. **(c)** Notation used throughout the paper. **(d)** Impulse response $y(t) = \alpha\, e^{\lambda t}$ to a Dirac drive $z(t) = \delta(t)$, a quadrature pair (i.e., with nearly equal amplitude responses and a nearly 90° phase shift) oscillating inside the envelope $|y(t)| = \alpha\, e^{-t/\tau_0}$ ($\gamma = 0.1$, $\omega\tau = 0.65$). **(e)** Frequency response amplitude for several values of γ, from sharp resonance to broad passband.

where $\omega > 0$ is the learnable frequency parameter and γ is the learnable bandwidth parameter. The state y is complex-valued even when z is real, with its real and imaginary parts representing the outputs of a quadrature pair of causal temporal band-pass filters (i.e., for narrow passbands, nearly equal amplitude responses and a nearly 90° phase shift). Hence, the modulus of the output $|y|$ follows the amplitude envelope of the filter's band regardless of phase. Defining $\alpha \triangleq \gamma/\tau$, $\omega_0 \triangleq (1-\gamma)\omega$, and $\lambda \triangleq -\alpha + \iota\,\omega_0$ collapses Eq. (4) into the compact scalar recurrence $\dot{y} = \lambda y + \alpha z$, so the RQF is a stable single-pole complex filter for any $\gamma \in (0, 1)$ (Fig. 1). The prospective factor $(1 + \iota\omega\tau)y$ sets the local recurrent prediction, whereas the homogeneous state evolves as $y(t + \Delta) = e^{\lambda\Delta}y(t)$ when $z = 0$. Appendix A.2 derives the transfer function (revealing the band-pass property), quality factor, damped-oscillator form, and the effect of cascading RQFs for fine-tuning the bandwidth.

Multi-layer architecture. A layer of an RQF stack bundles n_ℓ scalar filters operating in parallel, each with its own phase-prior frequency $\omega_i^{(\ell)}$ and bandwidth $\gamma_i^{(\ell)}$ but sharing a common time constant τ. Collecting the parameters into vectors $\boldsymbol{\omega}^{(\ell)}, \boldsymbol{\gamma}^{(\ell)} \in \mathbb{R}^{n_\ell}$ and writing the layer state as $\mathbf{y}^{(\ell)} \in \mathbb{C}^{n_\ell}$, the layer dynamics are the element-wise vectorization of Eq. (4):

$$\tau\, \dot{\mathbf{y}}^{(\ell)} \;=\; -\mathbf{y}^{(\ell)} \;+\; \boldsymbol{\gamma}^{(\ell)} \odot \mathbf{z}^{(\ell)} \;+\; (\mathbf{1} - \boldsymbol{\gamma}^{(\ell)}) \odot (\mathbf{1} + \iota\tau\, \boldsymbol{\omega}^{(\ell)}) \odot \mathbf{y}^{(\ell)}, \tag{5}$$

where $\mathbf{z}^{(\ell)} \in \mathbb{C}^{n_\ell}$ is the bottom-up input drive to the layer; $\odot$ denotes the element-wise product; and $\mathbf{1} \in \mathbb{R}^{n_\ell}$ is the all-ones vector. Defining $\boldsymbol{\alpha}^{(\ell)} \triangleq \boldsymbol{\gamma}^{(\ell)}/\tau$ and $\boldsymbol{\lambda}^{(\ell)} \triangleq -\boldsymbol{\alpha}^{(\ell)} + \iota\,(\mathbf{1} - \boldsymbol{\gamma}^{(\ell)}) \odot \boldsymbol{\omega}^{(\ell)}$, Eq. (5) collapses to the compact diagonal complex ODE:

$$\dot{\mathbf{y}}^{(\ell)} \;=\; \boldsymbol{\lambda}^{(\ell)} \odot \mathbf{y}^{(\ell)} \;+\; \boldsymbol{\alpha}^{(\ell)} \odot \mathbf{z}^{(\ell)}. \tag{6}$$

To build a deep network, we stack L layers. The input drive to layer ℓ is

$$\mathbf{z}^{(\ell)} \;=\; \mathbf{W}^{(\ell)}\mathbf{x}^{(\ell)}, \qquad \mathbf{x}^{(\ell)} \;=\; \rho\Big(\mathbf{y}^{(\ell-1)}\Big), \tag{7}$$

where $\rho : \mathbb{C}^{n_{\ell-1}} \to \mathbb{C}^{n_{\ell-1}}$ is an activation function (also called an output nonlinearity); $\mathbf{x}^{(\ell)} \in \mathbb{C}^{n_{\ell-1}}$ is the *layer input* to layer ℓ, with the boundary $\mathbf{x}^{(1)} \triangleq \mathbf{r}_{\text{in}}$; $\mathbf{W}^{(\ell)} \in \mathbb{C}^{n_\ell \times n_{\ell-1}}$ is a complex-valued inter-layer weight matrix; and $\mathbf{z}^{(\ell)} \in \mathbb{C}^{n_\ell}$ is the post-weight bottom-up input drive. The activation function ρ is not required to be holomorphic: nothing in the layer dynamics or in the discretization of Section 3 relies on complex differentiability, and Wirtinger calculus supplies gradients for any real-differentiable map of $(\mathrm{Re}(\mathbf{y}), \mathrm{Im}(\mathbf{y}))$. The default is the *pointwise* family $[\rho(\mathbf{y})]_i = \rho_0(y_i)$, with the canonical choice being the split-ReLU

$$\rho_0(y) \;=\; \max\{0, \mathrm{Re}(y)\} \;+\; \iota\, \max\{0, \mathrm{Im}(y)\}, \tag{8}$$

which applies a real-valued rectifier independently to the real and imaginary components of the quadrature pair, and is non-holomorphic by construction.

Discretization. For the implementation, we discretize the continuous-time layer dynamics in Eq. (6) with a *zero-order hold* (ZOH) on the bottom-up drive $\mathbf{z}^{(\ell)}$. ZOH treats $\mathbf{z}^{(\ell)}$ as piecewise constant on each integration interval $[t_n, t_{n+1}]$, with the constant value denoted by $\mathbf{z}^{(\ell)}_{n+1}$, and integrates the resulting forced linear ODE in closed form. It is the default discretization for diagonal SSMs, adopted by DSS, S4D, S5, and Mamba [22, 21, 28, 23], because it (i) is unconditionally stable for any pole in the open left half-plane, meaning that high-Q (narrow passband) filters impose no extra step-size constraint; (ii) integrates the linear part exactly, retaining the full $e^{\boldsymbol{\lambda} h}$ response at finite h; and (iii) agrees with forward Euler to first order in h/τ, with an $O((h/\tau)^2)$ difference. ZOH discretization of Eq. (6) yields the diagonal complex affine recurrence and coefficients

$$\mathbf{y}^{(\ell)}_{n+1} \;=\; \bar{\mathbf{A}}^{(\ell)}_h \odot \mathbf{y}^{(\ell)}_n \;+\; \bar{\mathbf{B}}^{(\ell)}_h \odot \mathbf{z}^{(\ell)}_{n+1}, \tag{9}$$

$$\bar{\mathbf{A}}^{(\ell)}_h \;\triangleq\; e^{\boldsymbol{\lambda}^{(\ell)} h}, \qquad \bar{\mathbf{B}}^{(\ell)}_h \;\triangleq\; \frac{\boldsymbol{\alpha}^{(\ell)}}{\boldsymbol{\lambda}^{(\ell)}} \odot \Big(\bar{\mathbf{A}}^{(\ell)}_h - \mathbf{1}\Big), \tag{10}$$

where $\mathbf{z}^{(\ell)}_{n+1} = \mathbf{W}^{(\ell)}\rho(\mathbf{y}^{(\ell-1)}_{n+1})$ samples the bottom-up input drive in Eq. (7) at t_{n+1}. The coefficients $\bar{\mathbf{A}}^{(\ell)}_h$ and $\bar{\mathbf{B}}^{(\ell)}_h$ depend only on the parameters $(\boldsymbol{\gamma}^{(\ell)}, \boldsymbol{\omega}^{(\ell)})$, the base time constant τ, and the step size h, and can be precomputed once per parameter update and reused across all time steps. The full derivation is given in Appendix B.1.

RQFs are a special case of diagonal state-space models. A continuous-time linear state-space model (SSM) maps an input signal $\mathbf{x}(t) \in \mathbb{R}^U$ through a latent state $\mathbf{y}(t) \in \mathbb{R}^P$ to an output $\mathbf{o}(t) \in \mathbb{R}^M$ via

$$\dot{\mathbf{y}}(t) \;=\; \mathbf{A}\,\mathbf{y}(t) + \mathbf{B}\,\mathbf{x}(t), \qquad \mathbf{o}(t) \;=\; \mathbf{C}\,\mathbf{y}(t) + \mathbf{D}\,\mathbf{x}(t), \tag{11}$$

with state matrix $\mathbf{A} \in \mathbb{R}^{P\times P}$, input matrix $\mathbf{B} \in \mathbb{R}^{P\times U}$, output matrix $\mathbf{C} \in \mathbb{R}^{M\times P}$, and feedthrough $\mathbf{D} \in \mathbb{R}^{M\times U}$ [31, 28]. All SSMs share this template but differ in how the recurrent core is parameterized and how the feature channels are coupled. S4 applies a bank of single-input single-output (SISO) SSMs across feature channels [31]; DSS and S4D replace S4's structured state matrix with diagonal variants [22, 21]; S5 uses one multi-input multi-output (MIMO) SSM whose effective recurrent matrix is complex-valued and diagonal [28]; the LRU uses a discrete diagonal linear recurrence with dense input and output projections [32]; and Mamba retains a diagonal state transition while making the discretized parameters input-dependent [23]. In the fixed-parameter diagonal case, the recurrent core can be written directly in a complex diagonal form,

$$\dot{\widetilde{\mathbf{y}}}(t) \;=\; \boldsymbol{\Lambda}\,\widetilde{\mathbf{y}}(t) + \mathbf{B}\,\mathbf{x}(t), \qquad \mathbf{o}(t) \;=\; \mathbf{C}\,\widetilde{\mathbf{y}}(t) + \mathbf{D}\,\mathbf{x}(t), \tag{12}$$

with $\boldsymbol{\Lambda} \in \mathbb{C}^{P\times P}$ diagonal, $\mathbf{B} \in \mathbb{C}^{P\times U}$, and $\mathbf{C} \in \mathbb{C}^{M\times P}$. ZOH discretization at step h gives $\widetilde{\mathbf{y}}_{k+1} = \bar{\mathbf{A}}_h\,\widetilde{\mathbf{y}}_k + \bar{\mathbf{B}}_h\,\mathbf{x}_{k+1}$, with

$$\bar{\mathbf{A}}_h \;\triangleq\; e^{\boldsymbol{\Lambda} h}, \qquad \bar{\mathbf{B}}_h \;\triangleq\; \boldsymbol{\Lambda}^{-1}\big(\bar{\mathbf{A}}_h - \mathbf{I}\big)\,\mathbf{B}. \tag{13}$$

The temporal modes are controlled by the entries of $\boldsymbol{\Lambda}$: an eigenvalue $\lambda_p \in \mathbb{C}$ in the open left half-plane produces a damped oscillation at angular frequency $\mathrm{Im}(\lambda_p)$ with decay rate $|\mathrm{Re}(\lambda_p)|$.

The RQF layer is the following constrained special case of Eq. (12), obtained by substituting the layer drive from Eq. (7) into Eq. (6):

$$\dot{\mathbf{y}}^{(\ell)} \;=\; \mathrm{diag}(\boldsymbol{\lambda}^{(\ell)})\,\mathbf{y}^{(\ell)} + \mathrm{diag}(\boldsymbol{\alpha}^{(\ell)})\,\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell)}. \tag{14}$$

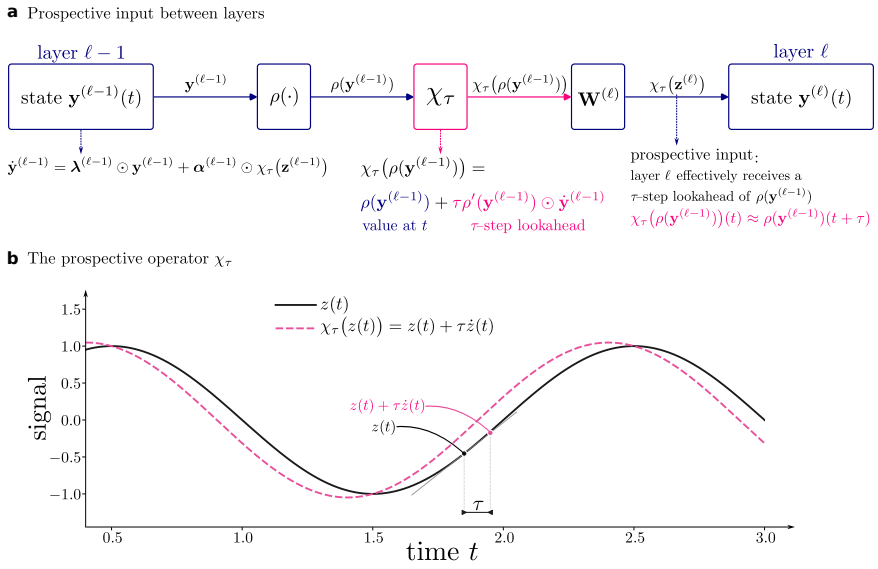


Figure 2: **Prospective-input coding in multi-layer RQFs.** The *prospective-input* variant replaces the bottom-up post-nonlinearity signal entering layer ℓ with its τ-step look-ahead $\chi_\tau(\rho(\mathbf{y}^{(\ell-1)}))$ defined in Eq. (16); the *instantaneous-input* baseline feeds $\rho(\mathbf{y}^{(\ell-1)})$ directly into $\mathbf{W}^{(\ell)}$. **(a)** The state $\mathbf{y}^{(\ell-1)}$ is passed through the activation ρ, advanced by the prospective operator χ_τ, and mixed by the inter-layer weight matrix $\mathbf{W}^{(\ell)}$ to form the bottom-up drive $\chi_\tau(\mathbf{z}^{(\ell)})$. **(b)** Action of χ_τ on a sinusoidal signal $z(t)$ (black). The look-ahead $\chi_\tau(z(t)) = z(t) + \tau\,\dot{z}(t)$ (pink, dashed) advances z by approximately τ by extrapolating along the local tangent.

Thus, the SSM input $\mathbf{x}(t)$ corresponds to the RQF layer input $\mathbf{x}^{(\ell)}$, and the dense input projection is the inter-layer weight matrix $\mathbf{W}^{(\ell)}$. In the canonical SSM notation of Eq. (12), the full continuous-time input matrix is $\mathbf{B} = \mathrm{diag}(\boldsymbol{\alpha}^{(\ell)})\mathbf{W}^{(\ell)}$; the diagonal factor is not an extra SSM parameter but the RQF input gain already present in Eq. (6). The RQF-specific constraint is that $\boldsymbol{\alpha}^{(\ell)} = \boldsymbol{\gamma}^{(\ell)}/\tau$ and $\boldsymbol{\lambda}^{(\ell)} = -\boldsymbol{\alpha}^{(\ell)} + \iota(\mathbf{1} - \boldsymbol{\gamma}^{(\ell)}) \odot \boldsymbol{\omega}^{(\ell)}$, which identifies the channel input gain with the inverse decay timescale $1/\tau_0^{(\ell)}$. The inclusion is one-way: a general diagonal SSM need not satisfy this pole–input-gain coupling and therefore need not be an RQF. The remaining SSM machinery transfers without modification: any output projection $\mathbf{C} \in \mathbb{C}^{M \times n_\ell}$ and feedthrough $\mathbf{D} \in \mathbb{C}^{M \times U}$ may be appended to read out from $\mathbf{y}^{(\ell)}$, and the per-block residual connections used in S4, S5, and Mamba stacks are admissible at the layer level. Unless stated otherwise, we set $\mathbf{C} = \mathbf{I}$ and $\mathbf{D} = \mathbf{0}$ throughout, so each layer's output is its own complex state $\mathbf{y}^{(\ell)}$ and inter-layer mixing is delegated to $\mathbf{W}^{(\ell+1)}$; residual connections are treated as an explicit configuration choice in our experiments (Section 4).

Selective RQFs (sRQFs). An sRQF multiplies each filter's bottom-up drive by a positive gain $\boldsymbol{\kappa}^{(\ell)}$ computed from the preceding layer's output, so that the input matrix of Eq. (14) becomes input-dependent:

$$\mathbf{B}_{\boldsymbol{\kappa}}^{(\ell)} = \mathrm{diag}(\boldsymbol{\kappa}^{(\ell)})\,\mathrm{diag}(\boldsymbol{\alpha}^{(\ell)})\,\mathbf{W}^{(\ell)}. \tag{15}$$

An sRQF can additionally make the bandwidth $\boldsymbol{\gamma}^{(\ell)}$ and frequency $\boldsymbol{\omega}^{(\ell)}$ parameters input-dependent, so that each filter's pole, and hence its state transition, varies with the input, as Mamba's state transition does through its input-dependent discretization step [23]. For the results reported below, only the input gain was input-dependent for the Path-X benchmark, whereas the gain, bandwidth $\boldsymbol{\gamma}$, and frequency $\boldsymbol{\omega}$ were all input-dependent for the Selective Copying benchmark (Section 4). Appendix F gives the parameterization, the discretized updates, and their parallel scan.

3 Prospective-input coding in multi-layer RQFs

We use the terms *instantaneous-input* and *prospective-input* to refer only to the bottom-up input drive from layer $\ell - 1$ to layer ℓ. Both variants retain the RQF recurrent prediction $\widehat{y} = (1 + \iota\omega\tau)y$ from Section 2; the comparison changes the input path, not the RQF dynamics. With the notation

of Eq. (7), the instantaneous choice uses $\mathbf{x}^{(\ell)} = \rho(\mathbf{y}^{(\ell-1)})$, while the prospective-input alternative applies the look-ahead operator before the inter-layer weights act (Fig. 2):

$$\widetilde{\mathbf{x}}^{(\ell)} \triangleq \chi_\tau\Big(\mathbf{x}^{(\ell)}\Big) = \rho\Big(\mathbf{y}^{(\ell-1)}\Big) + \tau\,\dot{\rho}\Big(\mathbf{y}^{(\ell-1)}\Big), \qquad \mathbf{z}_\chi^{(\ell)} = \mathbf{W}^{(\ell)}\widetilde{\mathbf{x}}^{(\ell)}, \tag{16}$$

where the dot denotes the time derivative. Since $\mathbf{W}^{(\ell)}$ is fixed over the integration interval, applying χ_τ to $\mathbf{x}^{(\ell)}$ and then multiplying by $\mathbf{W}^{(\ell)}$ is equivalent in the forward pass to applying χ_τ to $\mathbf{z}^{(\ell)}$; the discretization below uses this compact input drive notation.

Discretization. Applying the ZOH update to the prospective input drive $\chi_\tau\left(\mathbf{z}^{(\ell)}\right)$ gives the following recurrence; Appendix B.2 provides the derivation:

$$\mathbf{y}_{n+1}^{(\ell)} = \bar{\mathbf{A}}_h^{(\ell)} \odot \mathbf{y}_n^{(\ell)} + \left[\bar{\mathbf{B}}_h^{(\ell)} + \boldsymbol{\gamma}^{(\ell)} \odot \bar{\mathbf{A}}_h^{(\ell)}\right] \odot \mathbf{z}_{n+1}^{(\ell)} - \boldsymbol{\gamma}^{(\ell)} \odot \bar{\mathbf{A}}_h^{(\ell)} \odot \mathbf{z}_n^{(\ell)}, \tag{17}$$

where $\mathbf{z}_k^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{x}_k^{(\ell)}$ and $\mathbf{x}_k^{(\ell)} = \rho(\mathbf{y}_k^{(\ell-1)})$. In standard signal-processing terms, Eq. (17) is a *first-order IIR filter with a two-tap feedforward path*: the recurrence on $\mathbf{y}^{(\ell)}$ has a single feedback pole $\bar{\mathbf{A}}_h^{(\ell)}$, while the compact drive enters through weighted taps at $\mathbf{z}_{n+1}^{(\ell)}$ and $\mathbf{z}_n^{(\ell)}$.

Prospective input adds no parameters and leaves the diagonal state transition and associative scan unchanged. For a length-T sequence, it adds only $O(Tn_\ell)$ element-wise work and one lagged read of the drive, so the asymptotic cost of inference and full BPTT is unchanged (Appendix B.2).

Why prospective-input coding improves the spatial gradient path. Spatial-only backpropagation freezes the temporal recurrence at the previous time step and leaves only the within-step chain through depth. An instantaneous-input Euler step sends each inter-layer error through $\mathbf{g}_{\mathrm{np}} = (h/\tau)\boldsymbol{\gamma}$, whereas prospective input uses $\mathbf{g}_+ = (1 + h/\tau)\boldsymbol{\gamma}$. As $h/\tau \to 0$, therefore, the discretization prefactor multiplying each spatial hop changes from $O(h/\tau)$ to $O(1)$ and the explicit $(h/\tau)^{L-\ell}$ factor disappears. This statement concerns that prefactor only: the common product of activation slopes, weight norms, and damping gains may still attenuate with depth, and the upper bounds in Appendix C do not guarantee an $O(1)$ complete gradient. The implemented ZOH update has the same small-h leading-order coefficients. Spatial-only backpropagation is local in time but still uses $\mathbf{W}^\dagger$ to propagate exact adjacent-layer errors, so it retains the weight-transport problem and is used here only to isolate the analyzed path.

4 Experiments[1]

RQFs are band-pass filters by construction (Appendix A.2), so speech provides a natural first test of whether prospective-input coding improves learning through the RQF cascade. We use the 10-class subset of Google Speech Commands v0.02 [33], comprising one-second utterances of the keywords *yes*, *no*, *up*, *down*, *left*, *right*, *on*, *off*, *stop*, and *go*. We consider two representations of each utterance: the raw waveform, supplied as a sequence of $16{,}000$ scalar samples, and 20 Mel-frequency cepstral coefficients (MFCCs), supplied as a sequence of 161 frames. Thus, the two representations pose the same classification problem at very different sequence lengths and levels of temporal abstraction.

We contrast two regimes for assigning gradient credit through the recurrence. Full backpropagation through time (BPTT) differentiates through the complete recurrent trajectory. Spatial-only backpropagation instead detaches every previous recurrent state and retains only the within-step chain through depth. We use the latter as a mechanistic diagnostic, not as the recommended optimizer, because it removes temporal gradient credit and exposes the spatial attenuation analyzed in Section 3.

The cleanest test is the RQF stack itself. We therefore begin with non-residual RQF models of depth $L \in \{2, 4, 6\}$ and width $n \in \{32, 64\}$, using split-ReLU between layers and no normalization or gating. Avoiding residual connections leaves no bypass around the recurrent substrate, so changes in spatial credit flow are attributed to the input drive. Within every RQF comparison, the prospective and non-prospective variants match in architecture, optimizer, schedule, seeds, and parameter count; the look-ahead scale and base time constant are both $\tau = 5h$. All tables report test accuracy as the mean $\pm$ sample standard deviation over five seeds (for Selective Copying, over the seeds that exceed chance performance), and Appendices E and F.1 give the parameterization and training protocol.

[1] Code for the experiments is available at `https://github.com/Sequel-Institute/prospective-rqf`.

Table 1: **RQF on Speech Commands.** Test accuracy for non-residual width-64 stacks trained with **(a)** full BPTT and **(b)** spatial-only backpropagation.

	Raw audio			MFCC		
Depth	Params	Prospective	Non-pros.	Params	Prospective	Non-pros.
(a) Full BPTT						
2	52.3k	94.94±0.10	94.40±0.17	53.5k	96.42±0.26	96.30±0.33
4	68.9k	96.27±0.16	95.77±0.22	70.2k	96.65±0.24	96.41±0.14
6	85.6k	96.35±0.20	96.13±0.12	86.8k	96.70±0.19	96.49±0.07
(b) Spatial-only backpropagation						
2	52.3k	67.76±7.19	78.11±0.71	53.5k	94.89±0.20	94.20±0.28
4	68.9k	84.70±3.17	81.70±1.96	70.2k	92.63±0.85	89.04±0.83
6	85.6k	82.92±5.42	75.34±1.18	86.8k	91.55±0.76	76.38±1.57

Table 2: **αP-S5 versus native S5 on Speech Commands, full BPTT.** Residual width-64 S5 blocks.

	Raw audio			MFCC		
Depth	Params	αP-S5	Native S5	Params	αP-S5	Native S5
2	48.9k	95.55±0.39	94.94±0.42	50.1k	96.52±0.20	95.89±0.38
4	74.2k	96.06±0.35	95.96±0.27	75.4k	96.40±0.17	96.31±0.25
6	99.5k	96.37±0.31	96.21±0.23	100.7k	96.41±0.20	96.36±0.14

Table 3: **Discrete ORGaNICs on MFCC Speech Commands.** Residual width-64 models.

		Full BPTT		Spatial-only	
Depth	Params	Prospective	Non-pros.	Prospective	Non-pros.
2	58.3k	93.28±0.37	92.10±0.21	79.88±1.28	81.49±1.46
4	124.3k	93.21±0.59	91.91±0.53	82.61±1.21	81.67±0.88
6	190.3k	91.92±0.19	91.92±0.11	83.43±1.73	80.56±1.41

Under full BPTT, prospective-input coding raises mean test accuracy in all matched depth, width, and feature comparisons (Tables 1 and 6). Adding residual connections between layers leaves these results unchanged: prospective input still gives the higher mean in every configuration (Appendix D.2, Table 8). The smallest raw-audio models achieve $94.59 \pm 0.28\%$, $95.66 \pm 0.23\%$, and $96.09 \pm 0.23\%$ at two, four, and six layers, respectively, while using only 23.5k–31.9k parameters (Table 6). The spatial-only runs reveal the predicted depth dependence more clearly (Table 1): on MFCC inputs, the prospective advantage grows from 0.69 to 3.59 to 15.17 points as depth increases.

The direct gradient measurements connect this behavioral result to the analysis. Under spatial-only backpropagation, prospective inputs produce larger spatial per-hop gains and early-layer weight-gradient norms than instantaneous inputs. Thus, the advantage of prospective input is evident when learning depends on the spatial path isolated by the theory (Appendix D.3; Tables 9 and 10).

We next test whether the prospective construction transfers to a conventional residual SSM by comparing αP-S5 with native S5 [28]. We define αP-S5 by scaling each row of the learned S5 input matrix with the corresponding native-clock pole decay rate $\alpha_p = -\mathrm{Re}(\lambda_p)$, i.e., $\mathbf{B}_c = \mathrm{diag}(\boldsymbol{\alpha})\widetilde{\mathbf{B}}$. Prospective input uses the shared physical look-ahead horizon $\tau = 5h$, matching the RQF models. Appendix E.3 and Eq. (E.11) give the complete architecture and discretization. The mean test accuracy of αP-S5 is higher at every tested depth, width, and input type (Tables 2 and 7). These experiments place the minimal RQF architecture in context. At width 32, the RQF is more accurate than αP-S5 at every raw-audio depth while using fewer parameters; at six layers, it achieves $96.09 \pm 0.23\%$ with 31.9k parameters, versus $95.06 \pm 0.99\%$ with 40.9k for αP-S5 (Tables 6 and 7).

To determine whether the effect depends on a linear RQF or SSM recurrence, we also apply the input correction to discrete residual ORGaNICs, a nonlinear recurrent substrate with dynamic, recurrent normalization [25, 34]. Prospective-input coding with full BPTT improves performance at two and four layers and ties at six layers. Under spatial-only backpropagation, the prospective model improves at four and six layers (Table 3). The correction therefore transfers beyond the linear RQF core.

Table 4: **Path-X, full BPTT.** Six-layer, width-64 models on the 16,384-step task (chance: 50%). The sRQF uses only the input gain.

Model	Prospective	Non-pros.
RQF	83.36 ± 1.71	81.25 ± 0.91
sRQF	86.31 ± 0.51	85.13 ± 0.29

Table 5: **Selective Copying, full BPTT.** Four-layer, width-32 models (chance: 7.1%). The sRQF makes γ, ω, and the input gain input-dependent. Parentheses give the number of seeds, out of five, that exceed chance performance.

Model	Params	Prospective	Non-pros.
RQF	17.8k	69.83 ± 1.78 (4/5)	68.98 ± 2.21 (2/5)
sRQF	33.0k	99.83 ± 0.07 (5/5)	99.72 ± 0.23 (2/5)

Speech is well matched to a bank of temporal band-pass filters, but it does not establish whether RQFs with prospective input can learn structure that is neither acoustic nor naturally localized in frequency. We therefore turn to Path-X from Long Range Arena [35], a binary classification task formed by flattening a 128×128 image into a 16,384-step sequence whose relevant endpoints can be separated across the full input. We retain the forward-only, non-residual RQF model, use six layers of width 64, and train with full BPTT using a Path-X-specific optimization schedule (Appendix E.4). The prospective and non-prospective models achieve $83.36 \pm 1.71\%$ and $81.25 \pm 0.91\%$, respectively (Table 4). Therefore, without bidirectionality, normalization, gating, or an expanded state, the minimal RQF stack learns a fully non-local 16,384-step task more than 30 points above chance. Prospective input yields better performance. When only the input gain is selective (γ and ω fixed), the sRQF improves further to $86.31 \pm 0.51\%$ with prospective input and $85.13 \pm 0.29\%$ without it.

Finally, we compare RQFs and sRQFs on Selective Copying [23], in which 16 data tokens are placed at random positions in a 4,096-step sequence of blank tokens and must be reproduced in order at 16 final recall positions. The task requires content-aware selection of the tokens to be memorized [23]. Gu and Dao [23] argue that gating the input cannot affect the random spacing between these tokens; on this task, the sRQF therefore makes γ, ω, and the input gain input-dependent (Eq. (15) and Appendix F). We train four-layer, width-32 RQFs and sRQFs with both input rules for 600 epochs of 1,000 updates using full BPTT, supervise only the recall positions, and use five seeds per configuration. Unlike the Speech Commands and Path-X models, these models apply divisive normalization across filters after the split-ReLU (Appendix F.1). With prospective input, the sRQF solves the task in all five seeds, achieving $99.83 \pm 0.07\%$ test recall-token accuracy (Table 5). Without prospective input, only two seeds exceed chance ($99.72 \pm 0.23\%$), and the other three remain at chance. Prospective input likewise raises the number of RQF seeds that exceed chance, from two to four of five.

5 Related work

Prospective coding in recurrent neural circuits. Prospective coding has been used to address timing mismatch and credit assignment in biologically motivated learning rules. Latent equilibrium uses the look-ahead variable $u + \tau\dot{u}$ to recover fast feedforward computation and local error propagation [2]. Neuronal least-action formulates a least-action principle over future-discounted voltages, yielding prospective firing rates and prospective somato-dendritic mismatch errors [10]. Generalized latent equilibrium separates retrospective membrane integration from prospective output dynamics, yielding local spatio-temporal credit assignment [11]. Recent work has also identified cellular mechanisms for prospective and retrospective firing [5] and framed prospective neurons as a solution to teaching-signal synchronization [36]. Our work complements these circuit theories by showing that prospectivity can also serve as an input-side architectural addition for deep recurrent networks.

Normalization, ORGaNICs, and quadrature dynamics. The RQF construction also belongs to a line of recurrent cortical circuit models in which dynamics, prediction, and normalization are coupled. Heeger [3] modeled neural activity as a compromise between feedforward drive, feedback drive, and prior drive, with prediction over time implemented by recursive quadrature pairs. ORGaNICs extend this idea to gated recurrent neural integrator circuits with dynamic normalization [25], a canonical neural computation [18]. Most prior ORGaNICs work has focused on stability and normalization [34, 37, 38]; RQFs were also combined with ORGaNICs to model temporal-frequency selectivity and direction selectivity of neurons in primary visual cortex [12]. The RQF is the corresponding single-channel complex primitive, whose complex state carries the quadrature pair.

Credit assignment and trainability. The trainability problem we study is related to, but distinct from, the classical vanishing-gradient problem in RNNs, which arises from products of recurrent Jacobians through time [39]. Online alternatives such as RTRL avoid reverse-time unrolling but introduce large sensitivity states [40]; recent work makes this more tractable by exploiting independent recurrent

modules across layers [41]. Equilibrium propagation, Lagrangian extensions, and Hamiltonian echo methods instead recover gradient information from the response or time reversal of physical dynamical systems [42–45]. Our learning mode of spatial-only backpropagation removes reverse-time temporal credit but does not solve the separate weight-transport problem across layers [29].

State-space models and discretization upgrades. SSMs have made diagonal recurrent substrates practical for long sequences. S4 introduced efficient structured state-space layers [31], while DSS and S4D showed that complex diagonal state matrices can retain much of this performance with simpler implementations [22, 21]. S5 and LRU refine this family through MIMO scans and carefully parameterized linear recurrences [28, 32]. Mamba addresses a different limitation by making discretized parameters input-dependent [23]. We deliberately keep the RQF parameters input-independent so that the prospective-input intervention and its gradient scaling can be isolated; sRQFs relax this. Mamba-3 also uses two input taps, but its convex endpoint interpolation is analytically distinct from the signed prospective difference (Appendix B.3) [46].

6 Discussion

Takeaways for machine learning. Prospective-input coding is a two-tap change to a diagonal SSM's input path that adds no parameters and preserves the parallel scan. Prospective RQFs yield higher mean test accuracy than their matched controls in all configurations trained with full BPTT; the prospective αP-S5 construction likewise exceeds native S5 in all comparisons. Separately, RQFs are a special case of diagonal SSMs with clear advantages: at width 32, a stack of these biologically motivated filters, carrying no normalization, no gating, and no residual path, is more accurate than S5 at every raw-audio depth while using fewer parameters.

Takeaways for computational neuroscience. Prospective firing has been measured in biological neurons, and the look-ahead operator that models it inverts a first-order membrane low-pass filter exactly. Our results give that inverse a specific role: carrying a teaching signal through a deep cascade of leaky integrators. The evidence is mechanistic rather than only behavioral. Under spatial-only backpropagation in a deep recurrent network, we measure the error gain surviving each backward hop across layers, and a prospective drive preserves several times more of it than an instantaneous one. The consequence is temporally local learning: a six-layer stack that never differentiates its recurrence through time achieves 83% on $16{,}000$-step raw audio, far longer than the sequences on which the local rules of NLA and GLE have been demonstrated.

Limitations. The prospective-input correction is not directly applicable to discrete-time gated RNNs such as LSTMs and GRUs [24, 47]. The operator $\chi_\tau = 1 + \tau\,\mathrm{d}/\mathrm{d}t$ is defined on a continuous-time trajectory and uses τ as an intrinsic integration timescale, whereas standard LSTMs and GRUs are specified by per-step algebraic update equations. A discrete architecture can be embedded in many possible continuous flows, but without such a substrate, there is no canonical derivative or effective time constant to invert. Our proof isolates the linear RQF substrate under spatial-only backpropagation. Nonlinear activations, residual connections, normalization layers, input-dependent parameters, and imperfect choices of the look-ahead scale can all change the size of the empirical gain. Finally, we did not attempt to maximize performance in this paper; future work should learn or adapt the effective time constants, use multi-timescale look-ahead operators, optimize the pole initialization, and test residual and normalization choices.

Selective RQFs and future work. The input gain alone raises Path-X accuracy by three to four points under both input rules (Table 4). Making γ and ω input-dependent as well enables ceiling performance on Selective Copying (Table 5). Associative recall, state tracking, and language modeling would test whether prospective input complements selectivity beyond these tasks.

Spatial-only backpropagation isolates within-step spatial credit assignment but deliberately discards credit through recurrent time. Extending the prospective-input mechanism to local learning rules that retain useful temporal credit is therefore an open problem. Temporal credit assignment in RNNs is difficult because gradients propagated through time can vanish or explode [39]; deriving local alternatives has consequently been a long-standing goal [40, 41, 10, 11]. Biological synapses do not receive gradients produced by unrolling an entire recurrent computation, yet the brain learns from temporally extended experience. The goal is to use neuroscience principles to identify recurrent dynamical models whose learning rules use only local information (local errors and activity) while retaining useful credit assignment through time.

Acknowledgments and Disclosure of Funding

We thank Stefano Martiniani, Jenny Listman, Kyle Stratis, Ruben Coen-Cagli, and Wayne Mackey for helpful discussions. This research was funded by Telepath. S.R. and M.M. are employees of Telepath. F.M. is a paid consultant for Telepath. D.J.H. is a cofounder of and paid consultant for Telepath. The authors declare no other competing interests.

References

[1] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[2] Paul Haider, Benjamin Ellenberger, Laura Kriener, Jakob Jordan, Walter Senn, and Mihai A Petrovici. Latent equilibrium: a unified learning theory for arbitrarily fast computation with arbitrarily slow neurons. *Advances in Neural Information Processing Systems*, 34:17839–17851, 2021.

[3] David J Heeger. Theory of cortical function. *Proceedings of the National Academy of Sciences*, 114(8):1773–1782, 2017.

[4] Harold Köndgen, Caroline Geisler, Stefano Fusi, Xiao-Jing Wang, Hans-Rudolf Lüscher, and Michele Giugliano. The dynamical response properties of neocortical neurons to temporally modulated noisy inputs in vitro. *Cerebral Cortex*, 18(9):2086–2097, 2008.

[5] Simon Brandt, Mihai Alexandru Petrovici, Walter Senn, Katharina Anna Wilmes, and Federico Benitez. Prospective and retrospective coding in cortical neurons. *arXiv preprint arXiv:2405.14810*, 2024.

[6] Daniel Ulrich. Dendritic resonance in rat neocortical pyramidal cells. *Journal of Neurophysiology*, 87(6):2753–2759, 2002.

[7] Stephanie E Palmer, Olivier Marre, Michael J Berry, and William Bialek. Predictive information in a sensory population. *Proceedings of the National Academy of Sciences*, 112(22):6908–6913, 2015.

[8] Srdjan Ostojic, Germán Szapiro, Eric Schwartz, Boris Barbour, Nicolas Brunel, and Vincent Hakim. Neuronal morphology generates high-frequency firing resonance. *Journal of Neuroscience*, 35(18):7056–7068, 2015.

[9] Michael Dimitriou and Benoni B Edin. Human muscle spindles act as forward sensory models. *Current Biology*, 20(19):1763–1767, 2010.

[10] Walter Senn, Dominik Dold, Akos F Kungl, Benjamin Ellenberger, Jakob Jordan, Yoshua Bengio, João Sacramento, and Mihai A Petrovici. A neuronal least-action principle for real-time learning in cortical circuits. *eLife*, 12:RP89674, 2024.

[11] Benjamin Ellenberger, Paul Haider, Federico Benitez, Jakob Jordan, Kevin Max, Ismael Jaras, Laura Kriener, and Mihai A Petrovici. Backpropagation through space, time and the brain. *Nature Communications*, 17(1):66, 2026. doi: 10.1038/s41467-025-66666-z.

[12] David J Heeger and Klavdia O Zemlianova. A recurrent circuit implements normalization, simulating the dynamics of V1 activity. *Proceedings of the National Academy of Sciences*, 117 (36):22494–22505, 2020.

[13] Rodney J Douglas, Christof Koch, Misha Mahowald, Kevan AC Martin, and Humbert H Suarez. Recurrent excitation in neocortical circuits. *Science*, 269(5226):981–985, 1995.

[14] David J Heeger, Eero P Simoncelli, and J Anthony Movshon. Computational models of cortical visual processing. *Proceedings of the National Academy of Sciences*, 93(2):623–627, 1996.

[15] Eero P Simoncelli and David J Heeger. A model of neuronal responses in visual area MT. *Vision Research*, 38(5):743–761, 1998.

[16] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[17] Maximilian Riesenhuber and Tomaso Poggio. Neural mechanisms of object recognition. *Current Opinion in Neurobiology*, 12(2):162–168, 2002.

[18] Matteo Carandini and David J Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51–62, 2012.

[19] Timothy D Oleskiw, Justin D Lieber, Eero P Simoncelli, and J Anthony Movshon. Foundations of visual form selectivity for neurons in macaque areas V1 and V2. *bioRxiv*, 2024. doi: 10.1101/2024.03.04.583307.

[20] Georg von Békésy. *Experiments in hearing.* McGraw Hill, 1960.

[21] Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022.

[22] 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.

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

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

[25] David J Heeger and Wayne E Mackey. Oscillatory recurrent gated neural integrator circuits (ORGaNICs), a unifying theoretical framework for neural dynamics. *Proceedings of the National Academy of Sciences*, 116(45):22783–22794, 2019.

[26] Guy E Blelloch. Prefix sums and their applications. Technical Report CMU-CS-90-190, School of Computer Science, Carnegie Mellon University, 1990.

[27] Eric Martin and Chris Cundy. Parallelizing linear recurrent neural nets over sequence length. In *International Conference on Learning Representations*, 2018.

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

[29] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7(1):13276, 2016.

[30] Bruce A Francis and Walter Murray Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976.

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

[32] Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR, 2023.

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

[34] Shivang Rawat, David J Heeger, and Stefano Martiniani. Unconditional stability of a recurrent neural circuit implementing divisive normalization. *Advances in Neural Information Processing Systems*, 37:14712–14750, 2024.

[35] 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.

[36] Nicolas Zucchet, Qianqian Feng, Axel Laborieux, Friedemann Zenke, Walter Senn, and João Sacramento. Teaching signal synchronization in deep neural networks with prospective neurons. *arXiv preprint arXiv:2511.14917*, 2025.

[37] Flaviano Morone, Shivang Rawat, David J Heeger, and Stefano Martiniani. Stabilization of recurrent neural networks through divisive normalization. *Proceedings of the National Academy of Sciences*, 123(30):e2601841123, 2026. doi: 10.1073/pnas.2601841123.

[38] Asit Pal, Shivang Rawat, David J Heeger, and Stefano Martiniani. Hierarchical neural circuit theory of normalization and inter-areal communication. *eLife*, Aug 2026. doi: 10.7554/eLife.111297.1.

[39] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318. PMLR, 2013.

[40] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[41] Nicolas Zucchet, Robert Meier, Simon Schug, Asier Mujika, and João Sacramento. Online learning of long-range dependencies. *Advances in Neural Information Processing Systems*, 36: 10477–10493, 2023.

[42] Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, 11:24, 2017.

[43] Serge Massar. Equilibrium propagation for learning in Lagrangian dynamical systems. *Physical Review E*, 112(3):035304, 2025.

[44] Víctor López-Pastor and Florian Marquardt. Self-learning machines based on Hamiltonian echo backpropagation. *Physical Review X*, 13(3):031020, 2023.

[45] Guillaume Pourcel and Maxence Ernoult. Learning long range dependencies through time reversal symmetry breaking. *arXiv preprint arXiv:2506.05259*, 2025.

[46] Aakash Lahoti, Kevin Y Li, Berlin Chen, Caitlin Wang, Aviv Bick, J Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. *arXiv preprint arXiv:2603.15569*, 2026.

[47] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

A Details on Recursive Quadrature Filters (RQFs)

This appendix collects the technical material supporting the construction in Section 2. We first recover the leaky-integrator substrate of Eq. (2) by modeling y as a dynamical process that minimizes a single-channel slice of the cortical energy function of Heeger [3] over time, and then collect the elementary linear dynamical systems facts that motivate the band-pass interpretation, the special-case relationship to SSMs, and the equivalence to a damped harmonic oscillator.

A.1 Energy-based derivation of RQFs

The starting point is Eq. 1 of Heeger [3], which includes a feedforward residual and a prior residual at every layer:

$$E_H = \sum_i \alpha_H^{(i)} \left[\lambda_H^{(i)} \sum_j \big(y_j^{(i)} - z_j^{(i)}\big)^2 + \big(1 - \lambda_H^{(i)}\big) \sum_j \big(y_j^{(i)} - \widehat{y}_j^{(i)}\big)^2 \right]. \tag{A.1}$$

We select one response coordinate, omit the terms contributed by the next layer, and condition on the supplied feedforward target z and prior target $\widehat{y}$. We extend this scalar compromise to two real quadrature coordinates with shared parameters and encode them as $y = y_1 + \iota y_2$ (and likewise for z and $\widehat{y}$). The sum of the paired real squared residuals is then a squared complex modulus. Although Heeger's $\alpha_H(t)$ and $\lambda_H(t)$ may vary, the RQF construction holds them fixed during relaxation, takes α_H to be common to a quadrature pair, and identifies a learned per-channel $\gamma = \lambda_H$ (the subscript distinguishes Heeger's symbols from the RQF decay rate α and pole λ used elsewhere). A channel-dependent α_H would instead induce channel-dependent effective base time constants.

Let τ_H denote Heeger's relaxation time. Gradient descent on the selected quadrature pair gives

$$\tau_H \dot{y} = -2\alpha_H \left[\gamma(y - z) + (1 - \gamma)(y - \widehat{y})\right]. \tag{A.2}$$

Defining $\tau \triangleq \tau_H/(2\alpha_H)$ absorbs the common scale. Equivalently, the normalized pair energy is

$$E = \gamma|y - z|^2 + (1 - \gamma)|y - \widehat{y}|^2, \qquad \frac{\partial E}{\partial y^\dagger} = \gamma(y - z) + (1 - \gamma)(y - \widehat{y}), \tag{A.3}$$

and, with z and $\widehat{y}$ held fixed, its Wirtinger-gradient dynamics $\tau\dot{y} = -\partial E/\partial y^\dagger$ are

$$\tau\dot{y} = -y + \gamma z + (1 - \gamma)\widehat{y}. \tag{A.4}$$

Along this flow,

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{\partial E}{\partial y}\dot{y} + \frac{\partial E}{\partial y^\dagger}\dot{y}^\dagger = -\frac{2}{\tau}\left|\frac{\partial E}{\partial y^\dagger}\right|^2 \leq 0.$$

Thus, the energy is nonincreasing and decreases strictly away from stationary points, while the dynamics recover Eq. (2).

The recurrent prior ($\widehat{y}$) comes from the quadrature predictor in Eq. 3 of Heeger [3]. For angular frequency Ω, its real-coordinate form is

$$\begin{bmatrix} \widehat{y}_1(t) \\ \widehat{y}_2(t) \end{bmatrix} = \begin{bmatrix} \cos(\Omega\Delta t) & -\sin(\Omega\Delta t) \\ \sin(\Omega\Delta t) & \cos(\Omega\Delta t) \end{bmatrix} \begin{bmatrix} y_1(t - \Delta t) \\ y_2(t - \Delta t) \end{bmatrix}, \tag{A.5}$$

or $\widehat{y}(t) = e^{\iota\Omega\Delta t} y(t - \Delta t)$: the state sampled Δt earlier is rotated forward to predict the present. The RQF applies the same rotation to the present state to predict one look-ahead horizon ahead, $\widehat{y} = e^{\iota\omega\tau} y$ with $\omega = \Omega$ and $\Delta t = \tau$, and keeps the first-order term

$$e^{\iota\omega\tau} y = \big[1 + \iota\omega\tau + O\big((\omega\tau)^2\big)\big]\, y, \qquad \widehat{y} \triangleq \chi_\tau(y)|_{\dot{y} = \iota\omega y} = (1 + \iota\omega\tau) y. \tag{A.6}$$

Substitution into Eq. (A.4) yields

$$\tau\dot{y} = -y + \gamma z + (1 - \gamma)(1 + \iota\omega\tau) y,$$

which recovers the RQF in Eq. (4).

A.2 State-space form, oscillator equivalence, and cascade bandwidth

Two-dimensional real state-space form. We write the scalar complex-valued equation in Eq. (4) as a real two-dimensional coupled dynamical system to show that the RQF *is* a damped harmonic oscillator. Decomposing the complex state $y = u + \imath v$ and the input $z = z_R + \imath z_I$ into their real and imaginary parts reveals that Eq. (4) is equivalent to the two-dimensional real linear system

$$\dot{u} \;=\; -\alpha u - \omega_0 v + \alpha z_R, \qquad \dot{v} \;=\; \omega_0 u - \alpha v + \alpha z_I, \tag{A.7}$$

or, in matrix form, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}$ with $\mathbf{x} = (u, v)^\top$, $\mathbf{z} = (z_R, z_I)^\top$, $\mathbf{A} = -\alpha\mathbf{I} + \omega_0\mathbf{J}$, $\mathbf{B} = \alpha\mathbf{I}$, and $\mathbf{J} = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, the standard planar rotation generator. The matrix $\mathbf{A}$ therefore implements an isotropic exponential decay at rate α combined with a planar rotation at angular velocity ω_0. The eigenvalues follow from $\det(\mathbf{A} - \lambda\mathbf{I}) = (\lambda+\alpha)^2 + \omega_0^2 = 0$, giving the complex conjugate pair $\lambda_{1,2} = -\alpha \pm \imath\omega_0$; both lie strictly in the open left half-plane for any $\gamma \in (0, 1)$, so the filter is unconditionally stable. The free response is $\mathbf{x}(t) = e^{-\alpha t}\,\mathbf{R}(\omega_0 t)\,\mathbf{x}(0)$ with $\mathbf{R}(\theta)$ a planar rotation, and the impulse response of the scalar complex form is the damped complex exponential $h(t) = \alpha\, e^{-\alpha t}(\cos(\omega_0 t) + \imath \sin(\omega_0 t))$ for $t \geq 0$, the same primitive that diagonal complex SSMs convolve with their input sequence.

Frequency response and quality factor. The band-pass interpretation follows directly from the scalar form $\dot{y} = \lambda y + \alpha z$. With a zero initial condition, the transfer function from z to y is $H(s) = \alpha/(s - \lambda) = \alpha/(s + \alpha - \imath\omega_0)$, so its magnitude on the imaginary axis is

$$|H(\imath\Omega)| \;=\; \frac{\alpha}{\sqrt{\alpha^2 + (\Omega - \omega_0)^2}}, \tag{A.8}$$

whose square is a Lorentzian; it peaks at unity gain when $\Omega = \omega_0$ and has full half-power bandwidth $\Delta\Omega_{3\mathrm{dB}} = 2\alpha$. The two learnable parameters set the two axes of this response: ω sets the phase-prior frequency and, through $\omega_0 = (1 - \gamma)\omega$, the resonance location, while γ sets the bandwidth through $\alpha = \gamma/\tau$. Combining them gives the band-pass quality factor

$$Q \;=\; \frac{\omega_0}{2\alpha} \;=\; \frac{(1-\gamma)\,\omega\,\tau}{2\gamma}. \tag{A.9}$$

Small γ therefore yields a narrow band-pass filter centered near the phase-prior frequency ω, whereas large γ broadens the gain toward first-order low-pass integration.

Equivalence with a damped harmonic oscillator. A scalar damped harmonic oscillator with natural (undamped) frequency ω_n and damping ratio ζ obeys the standard second-order equation

$$\ddot{x} \;+\; 2\zeta\omega_n\,\dot{x} \;+\; \omega_n^2\,x \;=\; F(t), \tag{A.10}$$

where $F(t)$ is an external forcing; the damping ratio ζ controls the qualitative regime, with the system being undamped when $\zeta = 0$, underdamped when $0 < \zeta < 1$ (the impulse response is a decaying sinusoid), critically damped when $\zeta = 1$, and overdamped when $\zeta > 1$. To put the RQF in this form, we differentiate the first equation in Eq. (A.7), substitute $\dot{v}$ from the second, and eliminate the residual v using the first equation again. The result is a single second-order ODE in u,

$$\ddot{u} \;+\; 2\alpha\,\dot{u} \;+\; (\alpha^2 + \omega_0^2)\,u \;=\; \alpha\big(\dot{z}_R + \alpha z_R\big) \;-\; \alpha\omega_0\, z_I, \tag{A.11}$$

which has exactly the form of Eq. (A.10) with the external forcing

$$F(t) \;=\; \alpha\big(\dot{z}_R(t) + \alpha\, z_R(t)\big) \;-\; \alpha\omega_0\, z_I(t), \tag{A.12}$$

built from the in-phase and quadrature components of z: a proportional-plus-derivative response $\alpha(\dot{z}_R + \alpha z_R)$ to the real part, plus a static coupling $-\alpha\omega_0 z_I$ to the imaginary part. Factoring α out of the real-part term and introducing the effective time constant $\tau_0 \equiv 1/\alpha = \tau/\gamma$ exhibits the in-phase drive as a *prospective* signal on the timescale τ_0: $\alpha(\dot{z}_R + \alpha z_R) = \alpha^2(z_R + \tau_0 \dot{z}_R) = \alpha^2\,\chi_{\tau_0}(z_R)$. The forcing can therefore be written compactly as

$$F(t) \;=\; \alpha^2\,\chi_{\tau_0}(z_R(t)) \;-\; \alpha\omega_0\, z_I(t). \tag{A.13}$$

Matching coefficients between Eqs. (A.10) and (A.11) reads off the oscillator parameters as

$$\omega_n \;=\; \sqrt{\alpha^2 + \omega_0^2} \;=\; |\lambda|, \qquad \zeta \;=\; \frac{\alpha}{\sqrt{\alpha^2 + \omega_0^2}} \;=\; \frac{\alpha}{|\lambda|}, \tag{A.14}$$

and substituting the RQF parameterization $\alpha = \gamma/\tau$ and $\omega_0 = (1-\gamma)\omega$ gives the closed-form map from the learnable parameters (ω, γ, τ) to the oscillator parameters,

$$\omega_n \;=\; \frac{1}{\tau}\sqrt{\gamma^2 + (1-\gamma)^2\,\omega^2\tau^2}, \qquad \zeta \;=\; \frac{\gamma}{\sqrt{\gamma^2 + (1-\gamma)^2\,\omega^2\tau^2}}. \tag{A.15}$$

For any $\gamma \in (0,1)$ and $\omega\tau > 0$, the inequality $\zeta < 1$ holds strictly, so an RQF is always underdamped: its impulse response is a sinusoid at the damped oscillation frequency $\omega_0 = \omega_n\sqrt{1-\zeta^2}$ contained in an $e^{-\alpha t}$ envelope, and its squared magnitude response is the Lorentzian of Eq. (A.8). The standard oscillator quality factor $Q_{\mathrm{osc}} = 1/(2\zeta) = \omega_n/(2\alpha)$ and the band-pass quality factor $Q = \omega_0/(2\alpha)$ from Eq. (A.9) therefore differ by the factor $\omega_n/\omega_0 = 1/\sqrt{1-\zeta^2}$, which approaches unity in the high-Q (low-ζ) regime, and the two coincide as $\zeta \to 0$.

Cascade bandwidth. Here, we summarize the effect of cascading RQFs, which could be an architectural knob for adjusting per-channel selectivity. A single RQF has a Lorentzian squared magnitude response with full half-power bandwidth $\mathrm{BW}_1 = 2\alpha$ set by Eq. (A.8). Sharper per-channel selectivity at fixed α can be obtained by cascading N identical RQFs *within* a layer, i.e., in series with no nonlinearity between them. With N identical RQFs in series, the transfer function is $H_N(s) = H(s)^N$, so by Eq. (A.8) the cascade magnitude squared is

$$|H_N(\iota\Omega)|^2 \;=\; \frac{\alpha^{2N}}{\left(\alpha^2 + (\Omega-\omega_0)^2\right)^N}, \tag{A.16}$$

which peaks at unity at $\Omega = \omega_0$ for every N, since stacking unity-gain filters preserves unity gain at resonance. Setting $|H_N(\iota\Omega)|^2 = 1/2$ and solving for the frequency offset,

$$\left(\alpha^2 + (\Omega-\omega_0)^2\right)^N \;=\; 2\,\alpha^{2N} \iff (\Omega-\omega_0)^2 \;=\; \left(2^{1/N}-1\right)\alpha^2, \tag{A.17}$$

gives the full half-power bandwidth $\mathrm{BW}_N = 2\alpha\sqrt{2^{1/N}-1}$ and therefore the cascade bandwidth ratio

$$\frac{\mathrm{BW}_N}{\mathrm{BW}_1} \;=\; \sqrt{2^{1/N}-1} \;\xrightarrow[N\to\infty]{}\; \sqrt{\frac{\ln 2}{N}} \;\approx\; \frac{0.833}{\sqrt{N}}, \tag{A.18}$$

where the asymptotic form follows from $2^{1/N} = 1 + (\ln 2)/N + O(1/N^2)$. A four-fold within-layer cascade, for example, would narrow the per-channel bandwidth to $\sqrt{2^{1/4}-1} \approx 0.435$ of a single-stage filter at fixed α while requiring $4\times$ the computation on the linear path and no extra learnable parameters (the cascade shares the channel's (ω_0, α)). We do not exploit this knob in the present work; every layer uses $N = 1$, but it remains an inexpensive design choice.

B Discretization

This appendix gives the full derivations of the ZOH discrete update in Eq. (9), stated in Section 2, and the impulse-ZOH update for prospective-input coding that produces the two-tap input path referenced in Section 3. As in the main text, $\mathbf{z}^{(\ell)}$ denotes the compact post-weight drive used for forward discretization; Appendix C expands $\mathbf{z}^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{x}^{(\ell)}$ when deriving learning rules.

B.1 Instantaneous input

Starting from the per-layer ODE $\dot{\mathbf{y}}^{(\ell)} = \boldsymbol{\lambda}^{(\ell)} \odot \mathbf{y}^{(\ell)} + \boldsymbol{\alpha}^{(\ell)} \odot \mathbf{z}^{(\ell)}$ in Eq. (6), the element-wise decoupling reduces the vector ODE to n_ℓ independent scalar problems, each admitting the variation-of-constants solution over $[t_n, t_n + h]$:

$$\mathbf{y}^{(\ell)}(t_n + h) \;=\; e^{\boldsymbol{\lambda}^{(\ell)}h} \odot \mathbf{y}^{(\ell)}(t_n) \;+\; \boldsymbol{\alpha}^{(\ell)} \odot \int_0^h e^{\boldsymbol{\lambda}^{(\ell)}(h-s)} \odot \mathbf{z}^{(\ell)}(t_n + s)\,\mathrm{d}s, \tag{B.1}$$

where the exponential is evaluated element-wise because $\boldsymbol{\lambda}^{(\ell)}$ is a vector. The zero-order hold scheme replaces $\mathbf{z}^{(\ell)}(t_n + s)$ on $(0, h]$ by the right-endpoint sample $\mathbf{z}^{(\ell)}_{n+1} = \mathbf{W}^{(\ell)}\rho(\mathbf{y}^{(\ell-1)}_{n+1})$, pulls it out of the integral, and evaluates the remaining kernel integral via $u = h - s$:

$$\int_0^h e^{\boldsymbol{\lambda}^{(\ell)}(h-s)}\,\mathrm{d}s \;=\; \int_0^h e^{\boldsymbol{\lambda}^{(\ell)}u}\,\mathrm{d}u \;=\; \frac{e^{\boldsymbol{\lambda}^{(\ell)}h} - \mathbf{1}}{\boldsymbol{\lambda}^{(\ell)}}, \tag{B.2}$$

with division taken element-wise (well-defined because $\mathrm{Re}(\lambda_i^{(\ell)}) = -\alpha_i^{(\ell)} < 0$). Substituting the coefficients $\bar{\mathbf{A}}_h^{(\ell)} \triangleq e^{\boldsymbol{\lambda}^{(\ell)} h}$ and $\bar{\mathbf{B}}_h^{(\ell)} \triangleq (\boldsymbol{\alpha}^{(\ell)}/\boldsymbol{\lambda}^{(\ell)}) \odot (\bar{\mathbf{A}}_h^{(\ell)} - \mathbf{1})$ from Eq. (10) yields the diagonal complex affine recurrence of Eq. (9). Both coefficients depend only on the per-filter parameters $(\boldsymbol{\gamma}^{(\ell)}, \boldsymbol{\omega}^{(\ell)})$, the base time constant τ, and the step size h, and can therefore be precomputed once per parameter update and reused across all time steps. The homogeneous recurrence is unconditionally stable, since $|e^{\lambda h}| = e^{-\alpha h} < 1$ for any $\alpha > 0$, and the ZOH update differs from forward Euler by $O((h/\tau)^2)$ for $h \ll \tau$.

B.2 Prospective input

For the prospective-input variant, applying χ_τ to the layer input $\mathbf{x}^{(\ell)}$ before $\mathbf{W}^{(\ell)}$ is equivalent in the forward pass to driving Eq. (6) by $\chi_\tau(\mathbf{z}^{(\ell)}) = \mathbf{z}^{(\ell)} + \tau\, \dot{\mathbf{z}}^{(\ell)}$ in place of $\mathbf{z}^{(\ell)}$. The variation-of-constants integral of Eq. (B.1) therefore picks up an additional $\tau\, \dot{\mathbf{z}}^{(\ell)}$ contribution. We model $\mathbf{z}^{(\ell)}$ on $[t_n, t_n + h]$ as a right-continuous piecewise-constant signal that holds the value $\mathbf{z}_{n+1}^{(\ell)}$ throughout the interval and that jumps from $\mathbf{z}_n^{(\ell)}$ to $\mathbf{z}_{n+1}^{(\ell)}$ at the left endpoint $s = 0$; this is the same hold convention used for the instantaneous-input ZOH update of Appendix B.1, made explicit at the boundary because the prospective term involves $\dot{\mathbf{z}}^{(\ell)}$. Under this convention, $\dot{\mathbf{z}}^{(\ell)}$ is a Dirac impulse at $s = 0$ of magnitude equal to the jump,

$$\dot{\mathbf{z}}^{(\ell)}(t_n + s) \;=\; \left(\mathbf{z}_{n+1}^{(\ell)} - \mathbf{z}_n^{(\ell)}\right) \delta(s), \tag{B.3}$$

and the look-ahead $\chi_\tau(\mathbf{z}^{(\ell)})$ splits additively into a constant part $\mathbf{z}_{n+1}^{(\ell)}$ on $[0, h]$ and an impulsive part $\tau(\mathbf{z}_{n+1}^{(\ell)} - \mathbf{z}_n^{(\ell)})\delta(s)$ at $s = 0$. Each part contributes in closed form to the integral of Eq. (B.1). The constant part reproduces the calculation of Appendix B.1 and contributes $\bar{\mathbf{B}}_h^{(\ell)} \odot \mathbf{z}_{n+1}^{(\ell)}$. The impulsive part is sampled by the kernel at $s = 0$, where $e^{\boldsymbol{\lambda}^{(\ell)}(h-s)} = e^{\boldsymbol{\lambda}^{(\ell)} h} = \bar{\mathbf{A}}_h^{(\ell)}$ (we use the standard convention that the boundary impulse at $s = 0$ is included in $\int_0^h$). Multiplying the impulse strength by this kernel value and by the prefactor $\boldsymbol{\alpha}^{(\ell)}$ from Eq. (B.1) gives

$$\boldsymbol{\alpha}^{(\ell)}\, \tau \odot \bar{\mathbf{A}}_h^{(\ell)} \odot \left(\mathbf{z}_{n+1}^{(\ell)} - \mathbf{z}_n^{(\ell)}\right) \;=\; \boldsymbol{\gamma}^{(\ell)} \odot \bar{\mathbf{A}}_h^{(\ell)} \odot \left(\mathbf{z}_{n+1}^{(\ell)} - \mathbf{z}_n^{(\ell)}\right), \tag{B.4}$$

where the equality uses $\boldsymbol{\alpha}^{(\ell)}\tau = \boldsymbol{\gamma}^{(\ell)}$ from $\boldsymbol{\alpha}^{(\ell)} = \boldsymbol{\gamma}^{(\ell)}/\tau$. Summing the two contributions and grouping by input sample yields the impulse-ZOH update of Eq. (17) in the main text. No finite-difference approximation has been introduced: the update is exact under the right-continuous piecewise-constant model of $\mathbf{z}^{(\ell)}$.

Computational cost. Prospective input adds no trainable parameters to an RQF layer and leaves its diagonal state transition and associative scan unchanged. For a sequence of length T, layer ℓ maps $n_{\ell-1}$ inputs to n_ℓ RQF channels and computes the dense drive $\mathbf{z}_t^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{x}_t^{(\ell)}$ in $O(Tn_\ell n_{\ell-1})$ work under either input rule. Prospective-input coding adds $O(Tn_\ell)$ element-wise computation and one lagged read of the already materialized drive, but no additional full-sequence activation storage; streaming execution needs only an $O(n_\ell)$ rolling buffer for the previous drive. The diagonal RQF scan remains $O(Tn_\ell)$ with $O(\log T)$ parallel depth. Thus, the asymptotic complexity of both inference and full BPTT is unchanged, although the second tap adds a small lower-order operation.

B.3 Two-tap form for diagonal SSMs

The preceding derivation does not rely on the RQF parameterization. To transfer the prospective-input correction to the diagonal SSM core of Eq. (12), we keep the SSM parameters fixed and replace the input signal entering the continuous-time state equation by its look-ahead:

$$\dot{\tilde{\mathbf{y}}}(t) \;=\; \boldsymbol{\Lambda}\, \tilde{\mathbf{y}}(t) + \mathbf{B}\, \chi_\tau(\mathbf{x}(t)) \;=\; \boldsymbol{\Lambda}\, \tilde{\mathbf{y}}(t) + \mathbf{B}[\mathbf{x}(t) + \tau\, \dot{\mathbf{x}}(t)], \quad \mathbf{o}(t) \;=\; \mathbf{C}\, \tilde{\mathbf{y}}(t) + \mathbf{D}\, \mathbf{x}(t), \tag{B.5}$$

where τ is the look-ahead scale in χ_τ and $(\boldsymbol{\Lambda}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ are exactly the SSM parameters from Eq. (12). Thus, the intervention is not a reparameterization of the SSM input matrix: $\mathbf{B}$ remains the continuous-time input matrix, and the only change is that the state receives $\chi_\tau(\mathbf{x})$ instead of $\mathbf{x}$. The feedthrough path is written with $\mathbf{x}(t)$ because the prospective-input correction concerns the recurrent state input; applying the same look-ahead to the direct path would be a separate architectural choice.

Under the same right-continuous hold convention used in Appendix B.2, $\mathbf{x}(t_n + s)$ takes the value $\mathbf{x}_{n+1}$ on $[0, h)$ and has a jump from $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$ at $s = 0$. Therefore, $\dot{\mathbf{x}}(t_n + s) = (\mathbf{x}_{n+1} - \mathbf{x}_n)\delta(s)$, and the variation-of-constants formula gives

$$\widetilde{\mathbf{y}}_{n+1} \;=\; \bar{\mathbf{A}}_h\, \widetilde{\mathbf{y}}_n + \bar{\mathbf{B}}_h\, \mathbf{x}_{n+1} + \tau\, \bar{\mathbf{A}}_h \mathbf{B}\, (\mathbf{x}_{n+1} - \mathbf{x}_n), \tag{B.6}$$

where $\bar{\mathbf{A}}_h = e^{\mathbf{\Lambda} h}$ and $\bar{\mathbf{B}}_h = \mathbf{\Lambda}^{-1}(\bar{\mathbf{A}}_h - \mathbf{I})\mathbf{B}$ are the standard ZOH coefficients from Eq. (13). Grouping terms by input sample yields the two-tap input path

$$\widetilde{\mathbf{y}}_{n+1} \;=\; \bar{\mathbf{A}}_h\, \widetilde{\mathbf{y}}_n + \bar{\mathbf{B}}_+\, \mathbf{x}_{n+1} + \bar{\mathbf{B}}_-\, \mathbf{x}_n, \qquad \bar{\mathbf{B}}_+ \;\triangleq\; \bar{\mathbf{B}}_h + \tau\, \bar{\mathbf{A}}_h \mathbf{B}, \qquad \bar{\mathbf{B}}_- \;\triangleq\; -\tau\, \bar{\mathbf{A}}_h \mathbf{B}. \tag{B.7}$$

Eq. (B.7) is the direct SSM analogue of Eq. (17). The recurrent dynamics are unchanged: the state is advanced by the same diagonal transition $\bar{\mathbf{A}}_h$, while the input enters through a two-tap feedforward path on the current and previous samples. For S4D, this amounts to replacing the one-tap ZOH input coefficient $\bar{\mathbf{B}}_h$ by the pair $(\bar{\mathbf{B}}_+, \bar{\mathbf{B}}_-)$, whose values are determined by the original continuous-time input matrix $\mathbf{B}$, the transition $\bar{\mathbf{A}}_h$, and the look-ahead scale τ; the pole parameterization, state transition, output projection, and feedthrough term are otherwise unchanged. The same coefficient replacement applies to any diagonal state-space model whose continuous-time core has the form of Eq. (12).

Comparison with Mamba-3. This two-tap form permits a concrete comparison with the exponential-trapezoidal rule of Mamba-3 [46]. Writing $\mathbf{v}_t = \mathbf{B}_t \mathbf{x}_t$, that rule uses (in Mamba-3's notation)

$$\mathbf{h}_t = \boldsymbol{\alpha}_t \odot \mathbf{h}_{t-1} + (1 - \lambda_t)\Delta_t\, \boldsymbol{\alpha}_t \odot \mathbf{v}_{t-1} + \lambda_t \Delta_t\, \mathbf{v}_t, \qquad \lambda_t \in [0, 1], \tag{B.8}$$

so its endpoint weights form a convex blend. Prospective coding instead discretizes $\mathbf{B}[\mathbf{x}(t) + \tau\dot{\mathbf{x}}(t)]$ and gives the signed previous-input tap $\bar{\mathbf{B}}_- \mathbf{x}_n = -\tau \bar{\mathbf{A}}_h \mathbf{B} \mathbf{x}_n$ of Eq. (B.7). For a matched scalar real LTI system with $\Delta_t = h$, matching even this previous-input coefficient in Eq. (B.8) would require $\lambda_t = 1 + \tau/h > 1$. The constructions therefore share a two-tap form but are not equivalent, and prospective input is not a special case of Mamba-3's admissible interpolation.

C Spatial-only backpropagation gradient derivation: full proof

This appendix gives the gradient calculation supporting Section 3. We consider an arbitrary real-valued loss local in time,

$$\mathcal{L}_{k+1} \;=\; \Phi_{k+1}\Big(\mathbf{y}_{k+1}^{(L)}, \mathbf{y}_{k+1}^{(L)\dagger}\Big), \tag{C.1}$$

where Φ_{k+1} may include labels, readouts, or other quantities sampled at the same time step but does not backpropagate through earlier recurrent states. Spatial-only backpropagation freezes $\mathbf{y}_k^{(\ell)}$ for every layer ℓ and differentiates only through the within-step spatial chain $\mathbf{y}_{k+1}^{(0)} \to \cdots \to \mathbf{y}_{k+1}^{(L)}$. Unlike the forward-discretization derivation, we expand the compact drive as $\mathbf{z}^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{x}^{(\ell)}$ because the gradients must pass through $\mathbf{W}^{(\ell)}$. We write $\mathbf{x}_{k+1}^{(\ell)} = \rho(\mathbf{y}_{k+1}^{(\ell-1)})$ and $\mathbf{x}_k^{(\ell)} = \rho(\mathbf{y}_k^{(\ell-1)})$, with $\mathbf{x}^{(1)}$ supplied by the external input.

Because ρ may be non-holomorphic, the exact chain rule is most transparent in real coordinates. For a complex vector and matrix, respectively, define

$$\mathcal{R}(\mathbf{v}) \;\triangleq\; \begin{bmatrix} \mathrm{Re}(\mathbf{v}) \\ \mathrm{Im}(\mathbf{v}) \end{bmatrix}, \qquad \mathcal{R}(\mathbf{W}) \;\triangleq\; \begin{bmatrix} \mathrm{Re}(\mathbf{W}) & -\mathrm{Im}(\mathbf{W}) \\ \mathrm{Im}(\mathbf{W}) & \mathrm{Re}(\mathbf{W}) \end{bmatrix}, \tag{C.2}$$

so that $\mathcal{R}(\mathbf{W}\mathbf{v}) = \mathcal{R}(\mathbf{W})\mathcal{R}(\mathbf{v})$. For a real vector $\mathbf{g}$, let $\mathcal{D}(\mathbf{g}) \triangleq \mathcal{R}(\mathrm{diag}(\mathbf{g}))$, and define the real activation Jacobian and layer error by

$$\mathbf{J}_\rho(\mathbf{y}) \;\triangleq\; \frac{\partial \mathcal{R}(\rho(\mathbf{y}))}{\partial \mathcal{R}(\mathbf{y})}, \qquad \boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell)} \;\triangleq\; \nabla_{\mathcal{R}(\mathbf{y}_{k+1}^{(\ell)})} \mathcal{L}_{k+1}. \tag{C.3}$$

For split-ReLU, $\mathbf{J}_\rho$ is diagonal, with the real and imaginary activity masks on its two diagonal blocks, and $\|\mathbf{J}_\rho\|_2 \le 1$ (using any standard subgradient at zero). Moreover, $\|\mathcal{R}(\mathbf{W})\|_2 = \|\mathbf{W}\|_2$ and $\|\mathcal{D}(\mathbf{g})\|_2 = \|\mathbf{g}\|_\infty$. For clarity, we use forward Euler in the derivation because it exposes the factor of h/τ directly. The ZOH implementation has the same leading-order spatial gradient coefficients in the small-step limit: $\bar{\mathbf{A}}_h^{(\ell)} = \mathbf{1} + \boldsymbol{\lambda}^{(\ell)} h + O(h^2)$ and $\bar{\mathbf{B}}_h^{(\ell)} = (h/\tau)\boldsymbol{\gamma}^{(\ell)} + O(h^2)$, while the impulse-ZOH prospective coefficients satisfy $\bar{\mathbf{B}}_h^{(\ell)} + \boldsymbol{\gamma}^{(\ell)} \odot \bar{\mathbf{A}}_h^{(\ell)} = \boldsymbol{\gamma}^{(\ell)} + O(h)$ and $-\boldsymbol{\gamma}^{(\ell)} \odot \bar{\mathbf{A}}_h^{(\ell)} = -\boldsymbol{\gamma}^{(\ell)} + O(h)$. Thus, the Euler calculation below proves the leading-order ZOH scaling as $h/\tau \to 0$.

C.1 Spatial-only backward pass, instantaneous input

Forward Euler discretization of the instantaneous-input RQF layer is

$$\mathbf{y}_{k+1}^{(\ell)} \;=\; \mathbf{A}^{(\ell)} \odot \mathbf{y}_{k}^{(\ell)} + \mathbf{g}_{\mathrm{np}}^{(\ell)} \odot \left(\mathbf{W}^{(\ell)} \mathbf{x}_{k+1}^{(\ell)}\right), \qquad \mathbf{g}_{\mathrm{np}}^{(\ell)} \;\triangleq\; \frac{h}{\tau}\, \boldsymbol{\gamma}^{(\ell)}, \tag{C.4}$$

where $\mathbf{A}^{(\ell)} = \mathbf{1} + h\boldsymbol{\lambda}^{(\ell)}$. Since $\mathbf{y}_{k}^{(\ell)}$ is frozen, the only live path from layer ℓ to layer $\ell+1$ runs through $\mathbf{x}_{k+1}^{(\ell+1)} = \rho(\mathbf{y}_{k+1}^{(\ell)})$ and then through the matrix $\mathbf{W}^{(\ell+1)}$. Applying the chain rule gives, for $\ell < L$,

$$\boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell)} \;=\; \mathbf{J}_{\rho}\left(\mathbf{y}_{k+1}^{(\ell)}\right)^{\top} \mathcal{R}\left(\mathbf{W}^{(\ell+1)}\right)^{\top} \mathcal{D}\left(\mathbf{g}_{\mathrm{np}}^{(\ell+1)}\right)^{\top} \boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell+1)}. \tag{C.5}$$

Consequently, in the Euclidean norm,

$$\left\|\boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell)}\right\|_2 \;\le\; \prod_{m=\ell+1}^{L} \left(\left\|\mathbf{J}_{\rho}\left(\mathbf{y}_{k+1}^{(m-1)}\right)\right\|_2 \left\|\mathbf{W}^{(m)}\right\|_2 \left\|\mathbf{g}_{\mathrm{np}}^{(m)}\right\|_{\infty}\right) \left\|\boldsymbol{\delta}_{\mathbb{R},k+1}^{(L)}\right\|_2. \tag{C.6}$$

When the weight norms, activation-Jacobian norms, and entries of $\boldsymbol{\gamma}^{(m)}$ are $O(1)$, Eq. (C.6) contains the explicit factor $(h/\tau)^{L-\ell}$. This is the discretization-induced depth attenuation: it is separate from any ordinary conditioning effects caused by the weights or nonlinearities. To state the parameter gradient without ambiguity about Wirtinger factors, let $\mathcal{C}$ invert vector realification, $\mathcal{C}([\mathbf{u}^{\top}, \mathbf{v}^{\top}]^{\top}) = \mathbf{u} + \iota\mathbf{v}$, and set $\boldsymbol{\delta}_{\mathbb{C},k+1}^{(\ell)} = \mathcal{C}(\boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell)})$. Define the packed complex gradient

$$\nabla_{\mathbf{W}}^{\mathrm{pack}} \mathcal{L} \;\triangleq\; \nabla_{\mathrm{Re}(\mathbf{W})} \mathcal{L} + \iota\, \nabla_{\mathrm{Im}(\mathbf{W})} \mathcal{L} \;=\; 2\, \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{*}}. \tag{C.7}$$

The corresponding local gradient contribution has the outer-product form

$$\nabla_{\mathbf{W}^{(\ell)}}^{\mathrm{pack}} \mathcal{L}_{k+1} \;=\; \left(\mathbf{g}_{\mathrm{np}}^{(\ell)} \odot \boldsymbol{\delta}_{\mathbb{C},k+1}^{(\ell)}\right) \mathbf{x}_{k+1}^{(\ell)\dagger}. \tag{C.8}$$

C.2 Spatial-only backward pass, prospective input

For the prospective case, the look-ahead is applied to the layer input $\mathbf{x}^{(\ell)}$ before multiplication by $\mathbf{W}^{(\ell)}$. Using a backward difference at t_{k+1} gives

$$\widetilde{\mathbf{x}}_{k+1}^{(\ell)} \;=\; \mathbf{x}_{k+1}^{(\ell)} + \tau\, \frac{\mathbf{x}_{k+1}^{(\ell)} - \mathbf{x}_{k}^{(\ell)}}{h} \;=\; \frac{h+\tau}{h}\, \mathbf{x}_{k+1}^{(\ell)} - \frac{\tau}{h}\, \mathbf{x}_{k}^{(\ell)}. \tag{C.9}$$

The forward Euler step is therefore

$$\mathbf{y}_{k+1}^{(\ell)} \;=\; \mathbf{A}^{(\ell)} \odot \mathbf{y}_{k}^{(\ell)} + \mathbf{g}_{+}^{(\ell)} \odot \left(\mathbf{W}^{(\ell)} \mathbf{x}_{k+1}^{(\ell)}\right) + \mathbf{g}_{-}^{(\ell)} \odot \left(\mathbf{W}^{(\ell)} \mathbf{x}_{k}^{(\ell)}\right), \tag{C.10}$$

with

$$\mathbf{g}_{+}^{(\ell)} \;\triangleq\; \frac{h+\tau}{\tau}\, \boldsymbol{\gamma}^{(\ell)}, \qquad \mathbf{g}_{-}^{(\ell)} \;\triangleq\; -\boldsymbol{\gamma}^{(\ell)}. \tag{C.11}$$

The previous input $\mathbf{x}_{k}^{(\ell)}$ is frozen as a state variable for spatial error propagation, so it does not create a live path from $\mathbf{y}_{k+1}^{(\ell-1)}$ to $\mathbf{y}_{k+1}^{(\ell)}$. The current weight matrix multiplying that frozen input is still trainable. For the inter-layer error recursion, only the current-input coefficient $\mathbf{g}_{+}^{(\ell+1)}$ appears:

$$\boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell)} \;=\; \mathbf{J}_{\rho}\left(\mathbf{y}_{k+1}^{(\ell)}\right)^{\top} \mathcal{R}\left(\mathbf{W}^{(\ell+1)}\right)^{\top} \mathcal{D}\left(\mathbf{g}_{+}^{(\ell+1)}\right)^{\top} \boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell+1)}. \tag{C.12}$$

The corresponding norm bound is

$$\left\|\boldsymbol{\delta}_{\mathbb{R},k+1}^{(\ell)}\right\|_2 \;\le\; \prod_{m=\ell+1}^{L} \left(\left\|\mathbf{J}_{\rho}\left(\mathbf{y}_{k+1}^{(m-1)}\right)\right\|_2 \left\|\mathbf{W}^{(m)}\right\|_2 \left\|\mathbf{g}_{+}^{(m)}\right\|_{\infty}\right) \left\|\boldsymbol{\delta}_{\mathbb{R},k+1}^{(L)}\right\|_2. \tag{C.13}$$

Since $\|\mathbf{g}_{+}^{(m)}\|_{\infty} = (1 + h/\tau)\|\boldsymbol{\gamma}^{(m)}\|_{\infty}$, the explicit small factor $(h/\tau)^{L-\ell}$ is absent. More precisely, at fixed depth and fixed parameters, the discretization prefactor on each hop is $O(1)$ rather than

$O(h/\tau)$ as $h/\tau \to 0$. The bounds in Eqs. (C.6) and (C.13) are upper bounds and provide no lower bound: the remaining product of activation-Jacobian norms, weight norms, and $\boldsymbol{\gamma}$ can still attenuate with depth. We therefore make no claim that the complete prospective gradient is $O(1)$ or depth-independent. The local weight gradient contains both input taps:

$$\nabla^{\mathrm{pack}}_{\mathbf{W}^{(\ell)}} \mathcal{L}_{k+1} \;=\; \left(\mathbf{g}^{(\ell)}_{+} \odot \boldsymbol{\delta}^{(\ell)}_{\mathbb{C},k+1}\right) \mathbf{x}^{(\ell)\dagger}_{k+1} + \left(\mathbf{g}^{(\ell)}_{-} \odot \boldsymbol{\delta}^{(\ell)}_{\mathbb{C},k+1}\right) \mathbf{x}^{(\ell)\dagger}_{k}. \tag{C.14}$$

Eq. (C.14) is local in time up to the one-step input memory introduced by the two-tap path, and local in space once the adjacent real-coordinate error $\boldsymbol{\delta}^{(\ell)}_{\mathbb{R},k+1}$ is available. Exact spatial backpropagation supplies that error through $\mathcal{R}(\mathbf{W}^{(\ell+1)})^{\top}$ and the activation Jacobian in Eqs. (C.5) and (C.12).

D Experimental results

D.1 Full BPTT results with 32 filters per layer

Tables 6 and 7 report the width-32 results for RQF and αP-S5, respectively.

Table 6: **RQF, full BPTT.** Non-residual stacks with 32 filters per layer on Speech Commands.

	Raw audio			MFCC		
Depth	Params	Prospective	Non-pros.	Params	Prospective	Non-pros.
2	23.5k	94.59 ± 0.28	93.96 ± 0.28	24.1k	96.38 ± 0.13	96.17 ± 0.22
4	27.7k	95.66 ± 0.23	95.29 ± 0.22	28.3k	96.33 ± 0.23	96.12 ± 0.31
6	31.9k	96.09 ± 0.23	95.27 ± 0.22	32.5k	96.35 ± 0.17	95.73 ± 0.27

Table 7: **αP-S5 versus native S5, full BPTT.** Residual S5 blocks with 32 units per layer on Speech Commands; the complete αP construction changes the continuous-time input gain, adds the second tap, and uses the pole clipping described in Appendix E.3.

	Raw audio			MFCC		
Depth	Params	αP-S5	Native S5	Params	αP-S5	Native S5
2	27.9k	93.46 ± 0.93	92.34 ± 0.50	28.6k	96.04 ± 0.32	95.36 ± 0.24
4	34.4k	94.74 ± 0.59	93.31 ± 1.09	35.1k	96.31 ± 0.32	95.84 ± 0.32
6	40.9k	95.06 ± 0.99	94.69 ± 0.66	41.5k	96.44 ± 0.28	95.84 ± 0.18

D.2 Full BPTT results with residual connections

Table 8 repeats the full BPTT comparison of Table 1(a) with identity residual connections between layers. Each of the L square RQF layers adds its input to its output through an identity connection, which adds no parameters; all other settings match Table 1(a). Every mean test accuracy is within 0.28 points of its non-residual counterpart, and prospective input still gives the higher mean in every configuration.

Table 8: **Residual RQF, full BPTT.** Stacks with 64 filters per layer and identity residual connections between layers on Speech Commands; otherwise identical to Table 1(a).

	Raw audio			MFCC		
Depth	Params	Prospective	Non-pros.	Params	Prospective	Non-pros.
2	52.3k	94.84 ± 0.25	94.68 ± 0.07	53.5k	96.42 ± 0.16	96.30 ± 0.27
4	68.9k	96.08 ± 0.18	95.84 ± 0.25	70.2k	96.55 ± 0.32	96.49 ± 0.23
6	85.6k	96.27 ± 0.24	96.01 ± 0.14	86.8k	96.69 ± 0.18	96.54 ± 0.11

D.3 Direct measurements of the spatial gradient path

These measurements test how much of the error signal survives as it propagates backward through the layers. We use a six-layer, 64-channel raw-audio RQF under spatial-only backpropagation at the paper's operating point, $h/\tau = 0.2$. Table 9 reports the weight-gradient norms: the two gradients are similar at layer 6, which is closest to the loss, but the instantaneous gradient decays much faster toward earlier layers. At layer 1, the prospective gradient is $10{,}290\times$ larger.

Table 9: **Direct weight-gradient norms.** Frobenius norms $\|\nabla^{\mathrm{pack}}_{\mathbf{W}^{(\ell)}}\mathcal{L}\|_F$ at $h/\tau = 0.2$.

Layer ℓ	Prospective	Instantaneous	Ratio
1	7.69×10^{-7}	7.47×10^{-11}	$10{,}290\times$
2	5.71×10^{-6}	6.01×10^{-9}	$950\times$
3	1.03×10^{-5}	5.61×10^{-8}	$185\times$
4	1.90×10^{-5}	5.62×10^{-7}	$34\times$
5	3.08×10^{-5}	5.04×10^{-6}	$6.1\times$
6	5.42×10^{-5}	4.87×10^{-5}	$1.1\times$

Table 10 sweeps the step size. We measure the per-hop error gain g as follows: with q_ℓ the RMS norm of $\partial\mathcal{L}/\partial\mathbf{x}^{(\ell)}$, we fit $g = (q_2/q_6)^{1/4}$ over the four hops from layer 6 to layer 2. The table compares the ratio $g_{\mathrm{pros}}/g_{\mathrm{inst}}$ with the exact-ZOH current-tap prediction $|\bar{\mathbf{B}}_h + \boldsymbol{\gamma} \odot \bar{\mathbf{A}}_h|/|\bar{\mathbf{B}}_h|$ for the same filter bank. As h/τ decreases, g_{pros} remains between 0.585 and 0.624, whereas g_{inst} falls approximately in proportion to h/τ. The measured ratio follows the predicted increase across the sweep.

Table 10: **Per-hop error gain versus step size.** g_{pros} and g_{inst} are the measured per-hop error gains for prospective and instantaneous input; prediction and measurement use the same six-layer, 64-channel filter bank.

h/τ	Predicted ratio	g_{pros}	g_{inst}	Measured ratio
1.00	1.96	0.619	0.361	1.72
0.50	2.95	0.594	0.220	2.70
0.20	5.93	0.585	0.104	5.65
0.10	10.91	0.598	0.0565	10.59
0.05	20.89	0.624	0.0305	20.43

E Architecture and hyperparameter details

E.1 RQF architecture

A bias-free real projection maps the input dimension $d_{\mathrm{in}} \in \{1, 20\}$ to width $n \in \{32, 64\}$. This is followed by $L \in \{2, 4, 6\}$ square complex RQF layers of width n, with split-ReLU between layers and no normalization, gating, dropout, bidirectionality, or residual connections. At every step, the classifier concatenates the real and imaginary parts of the final state and applies a real MLP $2n \to 256 \to 10$ with ReLU. Predictions use logits averaged across time. Parameter counts refer to real-valued trainable degrees of freedom, so a complex parameter counts twice. With $C = 10$ classes, the count is

$$N_{\mathrm{params}} = d_{\mathrm{in}}n + 2Ln^2 + 2Ln + 512n + 2826. \tag{E.1}$$

Raw waveforms have shape $(16{,}000, 1)$ and use $h = 1/16{,}000$ with fixed $\tau = 5h = 3.125 \times 10^{-4}$. MFCC inputs have shape $(161, 20)$ and use their effective 160 Hz frame clock, $h = 1/160$ and $\tau = 5h = 0.03125$. Path-X uses $h = 1/16{,}384$ and $\tau = 5h$. No separate look-ahead scale is learned. Each RQF channel has complex state s_i, dimensionless frequency $\theta_i = \tau\omega_i$, bandwidth parameter γ_i, and continuous-time dynamics

$$\frac{\mathrm{d}s_i}{\mathrm{d}t} \;=\; \lambda_i s_i + \frac{\gamma_i}{\tau}u_i(t), \qquad \lambda_i \;=\; -\frac{\gamma_i}{\tau} + \iota\,\omega_i(1-\gamma_i), \tag{E.2}$$

where $u_i(t)$ is the post-weight drive; s_i and u_i are the per-channel entries of $\mathbf{y}^{(\ell)}$ and $\mathbf{z}^{(\ell)}$ in the main text. All runs use the ZOH recurrence in Eq. (E.3); prospective input replaces its single input coefficient with the two-tap coefficients in Eq. (E.4), with the delayed input initialized to zero.

The scalar instantaneous recurrence is

$$s_{i,n+1} \;=\; \bar{A}_{h,i} s_{i,n} + \bar{B}_{h,i} u_{i,n+1}, \qquad \bar{A}_{h,i} = e^{\lambda_i h}, \qquad \bar{B}_{h,i} = \frac{\gamma_i}{\tau\lambda_i}\left(\bar{A}_{h,i} - 1\right). \tag{E.3}$$

The prospective-input recurrence keeps the same state coefficient and replaces the input contribution by the two-tap drive

$$s_{i,n+1} \;=\; \bar{A}_{h,i} s_{i,n} + \left(\bar{B}_{h,i} + \gamma_i \bar{A}_{h,i}\right) u_{i,n+1} - \gamma_i \bar{A}_{h,i} u_{i,n}, \tag{E.4}$$

with the previous input buffer initialized to zero. This is the scalar-channel version of Eq. (17).

RQF parameterization and initialization. The learned leaves are $\log\theta_i$ and $\log\gamma_i$:

$$\theta_i = \tau\omega_i = e^{\log\theta_i}, \qquad \gamma_i = e^{\log\gamma_i}. \tag{E.5}$$

For a sequence with model-coordinate duration T_{seq}, initialization is

$$\log\theta_i \sim \mathcal{U}(\log\theta_{\min}, \log\theta_{\max}), \quad \theta_{\min} = \frac{2\pi\tau}{T_{\mathrm{seq}}}, \quad \theta_{\max} = 2\pi F, \quad \gamma_i = \frac{\theta_i}{1+\theta_i}. \tag{E.6}$$

All runs use $F = 1$. Speech Commands uses $T_{\mathrm{seq}} = 1$ in the model clock for each input representation, while Path-X uses $T_{\mathrm{seq}} = 2\pi$. After every optimizer step, θ_i is clamped to $[\theta_{\min}, 2\pi]$ and γ_i to $[10^{-6}, 1-10^{-6}]$.

Weight and readout initialization. Let n_{in} be the complex fan-in, $G_f = 1/2$ the nominal RQF filter power gain, and G_ρ the input activation's second-moment gain. Each real and imaginary component of a complex weight is sampled independently as follows:

$$\mathrm{Re}(W_{ij}), \mathrm{Im}(W_{ij}) \sim \mathcal{N}(0, \sigma_W^2), \qquad \sigma_W = \left(2 n_{\mathrm{in}} G_f G_\rho\right)^{-1/2}. \tag{E.7}$$

Thus, $G_\rho = 1$ for a direct standardized input and $G_\rho = 1/2$ after split-ReLU, giving $\sigma_W = 1/\sqrt{n_{\mathrm{in}}}$ and $\sqrt{2/n_{\mathrm{in}}}$, respectively. The separate real input projection retains the default PyTorch linear initialization, uniform on $[-1/\sqrt{d_{\mathrm{in}}}, 1/\sqrt{d_{\mathrm{in}}}]$, and its bias is removed. The first classifier layer likewise retains its default initialization, with weights and biases uniform on $[-1/\sqrt{2n}, 1/\sqrt{2n}]$. The $256 \to C$ logit weights are normal with standard deviation 10^{-3} and zero bias. The reference implementation evaluates the recurrence in complex64 with autocasting disabled; optimized scan kernels preserve the same coefficients.

E.2 ORGaNICs architecture

The discrete ORGaNICs control follows the sequential model of Rawat et al. [34]. We fix the recurrent matrix in every layer to $\mathbf{W}_r = \mathbf{I}$, learn the intrinsic time constants, retain the gain and normalization dynamics, and stack $L \in \{2, 4, 6\}$ layers of width 64. The reported parameter counts include the learned intrinsic time constants. Shape-matched layers use identity residual additions; the first $20 \to 64$ MFCC layer cannot use one. Layer normalization is disabled.

For layer ℓ, let $\mathbf{x}^{(\ell)}(t) \in \mathbb{R}^{d_\ell}$ denote its input, with $\mathbf{x}^{(1)}$ equal to the 20-dimensional MFCC sequence. Each layer has 64-dimensional excitatory state $\mathbf{y}^{(\ell)}$, inhibitory normalization state $\mathbf{a}^{(\ell)}$, input gain $\mathbf{b}^{(\ell)}$, and semisaturation gain $\mathbf{b}_0^{(\ell)}$. We use the rectified-input variant, defining $[\mathbf{u}]_+ = \max(\mathbf{u}, \mathbf{0})$ element-wise and

$$\mathbf{z}^{(\ell)}(t) \;=\; \left[\mathbf{W}_{zx}^{(\ell)} \mathbf{x}^{(\ell)}(t)\right]_+, \qquad \mathbf{z}_\chi^{(\ell)}(t) \;=\; \mathbf{z}^{(\ell)}(t) + \delta_\chi \, \boldsymbol{\tau}_y^{(\ell)} \odot \dot{\mathbf{z}}^{(\ell)}(t). \tag{E.8}$$

Here, $\mathbf{z}^{(\ell)}(t)$ is the input drive and $\mathbf{z}_\chi^{(\ell)}(t)$ is the drive passed to the ORGaNICs input-gain pathway. The binary variable δ_χ selects the input-drive condition: $\delta_\chi = 0$ gives the instantaneous-input baseline, whereas $\delta_\chi = 1$ gives the prospective-input variant $\chi_{\boldsymbol{\tau}_y^{(\ell)}}(\mathbf{z}^{(\ell)}(t))$ with look-ahead scale $\boldsymbol{\tau}_y^{(\ell)}$. The gain modulators evolve dynamically toward the following instantaneous targets:

$$\begin{aligned} \mathbf{B}^{(\ell)}(t) &= \sigma_{\mathrm{sig}}\Big(\mathbf{W}_{xb}^{(\ell)} \mathbf{x}^{(\ell)}(t) + \mathbf{W}_{yb}^{(\ell)} \mathbf{y}^{(\ell)}(t) + \mathbf{W}_{ab}^{(\ell)} \mathbf{a}^{(\ell)}(t)\Big), \\ \mathbf{B}_0^{(\ell)}(t) &= \sigma_{\mathrm{sig}}\Big(\mathbf{W}_{x0}^{(\ell)} \mathbf{x}^{(\ell)}(t) + \mathbf{W}_{y0}^{(\ell)} \mathbf{y}^{(\ell)}(t) + \mathbf{W}_{a0}^{(\ell)} \mathbf{a}^{(\ell)}(t)\Big), \end{aligned} \tag{E.9}$$

where σ_{sig} is the logistic sigmoid; $\mathbf{B}^{(\ell)}$ is the target for the input gain $\mathbf{b}^{(\ell)}$; and $\mathbf{B}_0^{(\ell)}$ is the target for the semisaturation gain $\mathbf{b}_0^{(\ell)}$.

For the rectified-recurrence variant, define the recurrent prediction $\widehat{\mathbf{y}}^{(\ell)} = [\mathbf{W}_r \mathbf{y}^{(\ell)}]_+ = [\mathbf{y}^{(\ell)}]_+$. The continuous-time ORGaNICs layer is then described by

$$\begin{aligned}
\boldsymbol{\tau}_y^{(\ell)} \odot \dot{\mathbf{y}}^{(\ell)} &= -\mathbf{y}^{(\ell)} + \mathbf{b}^{(\ell)} \odot \mathbf{z}_\chi^{(\ell)} + \left(\mathbf{1} - \sqrt{\left[\mathbf{a}^{(\ell)}\right]_+}\right) \odot \widehat{\mathbf{y}}^{(\ell)}, \\
\boldsymbol{\tau}_a^{(\ell)} \odot \dot{\mathbf{a}}^{(\ell)} &= -\mathbf{a}^{(\ell)} + \sigma_0^2 \left(\mathbf{b}_0^{(\ell)}\right)^2 + \mathbf{W}_{ay}^{(\ell)} \left(\left[\mathbf{y}^{(\ell)}\right]_+^2 \odot \left[\mathbf{a}^{(\ell)}\right]_+ \right), \\
\boldsymbol{\tau}_b^{(\ell)} \odot \dot{\mathbf{b}}^{(\ell)} &= -\mathbf{b}^{(\ell)} + \mathbf{B}^{(\ell)}, \\
\boldsymbol{\tau}_{b0}^{(\ell)} \odot \dot{\mathbf{b}}_0^{(\ell)} &= -\mathbf{b}_0^{(\ell)} + \mathbf{B}_0^{(\ell)}.
\end{aligned} \tag{E.10}$$

All products, powers, divisions by time constants, and square roots in Eq. (E.10) are element-wise except for multiplication by learned matrices. In the implementation, Eq. (E.10) is integrated with an ordered forward Euler step: the gain states are updated first, and those updated gains enter the principal and normalization updates. With $\dot{\mathbf{z}}$ approximated by a backward difference, the Euler step multiplies the prospective term $\mathbf{b} \odot \boldsymbol{\tau}_y \odot (\mathbf{z}_t - \mathbf{z}_{t-1})/\Delta t$ by $\boldsymbol{\eta}_y = \Delta t/\boldsymbol{\tau}_y$, so the two-tap prospective correction reduces to $\mathbf{b} \odot (\mathbf{z}_t - \mathbf{z}_{t-1})$ in addition to the ordinary Euler drive, with the previous input-drive buffer initialized as $\mathbf{z}_{-1}^{(\ell)} = \mathbf{0}$. The gain dynamics for $\mathbf{b}_0$ and $\mathbf{b}$ are otherwise identical in the prospective and instantaneous variants.

The normalization matrix $\mathbf{W}_{ay}^{(\ell)}$ is a learned full-rank positive matrix represented in log space; its initialization is 50% sparse off-diagonal with unit diagonal. $\mathbf{W}_{zx}$ and the six gain-target matrices in Eq. (E.9) use Kaiming-uniform initialization with $a = \sqrt{5}$. The semisaturation parameter is fixed to $\sigma_0 = 1$. The four time-constant vectors are learned through bounded positive sigmoid maps; their unconstrained leaves are standard normal at initialization, with $0 < \Delta t/\tau_y, \Delta t/\tau_a < 0.05$ and $0 < \Delta t/\tau_b, \Delta t/\tau_{b0} < 0.1$. The principal state $\mathbf{y}^{(\ell)}(0)$ is a normalized random positive vector; $\mathbf{a}^{(\ell)}(0) \sim \mathcal{U}(0,1)$ is not normalized, $\mathbf{b}_0^{(\ell)}(0) = \mathbf{1}$, and $\mathbf{b}^{(\ell)}(0) \sim \mathcal{U}(0,1)$. A learned linear $64 \to 10$ readout produces per-step logits, which are averaged for evaluation. The input correction is the only difference between the members of each prospective/non-prospective pair.

E.3 αP-S5 architecture and parameterization

The S5 controls use the native forward-only residual S5 architecture [28]. A real input projection maps $d_{\mathrm{in}} \in \{1, 20\}$ to $n \in \{32, 64\}$ units per layer. The $L \in \{2, 4, 6\}$ pre-normalized residual blocks use this value of n, conjugate symmetry, eight HiPPO blocks, ZOH discretization, and a half-GLU pointwise map. The readout normalizes the final n-dimensional block state and projects it to 64 dimensions; that representation is mean-pooled and classified by a $64 \to 256 \to 10$ GELU MLP with dropout 0.1. The reported runs use batch pre-normalization and are unidirectional.

Let $\boldsymbol{\lambda}$ collect the diagonal poles in S5's native clock, let $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$, and let $\boldsymbol{\Delta}$ collect the learned per-mode steps. For a physical input interval h, mode p has effective continuous-time pole $\lambda_{\mathrm{eff},p} = (\Delta_p/h)\lambda_p$. We define $\alpha_p = -\mathrm{Re}(\lambda_p) > 0$ and set $\mathbf{B}_c = \mathrm{diag}(\boldsymbol{\alpha})\widetilde{\mathbf{B}}$ in the native clock. The corresponding physical input row is $\mathbf{B}_{\mathrm{eff},p:} = (\Delta_p/h)\mathbf{B}_{c,p:} = -\mathrm{Re}(\lambda_{\mathrm{eff},p})\widetilde{\mathbf{B}}_{p:}$, so the α gain remains tied to the physical pole decay rate and is not learned separately.

Prospective input uses the same shared physical horizon as the RQF experiments, $\tau = 5h$. Substituting $\mathbf{B}_{\mathrm{eff},p:} = (\Delta_p/h)\mathbf{B}_{c,p:}$ into the impulse-ZOH correction of Eq. (B.6) gives $\tau \bar{A}_p \mathbf{B}_{\mathrm{eff},p:} = 5\Delta_p \bar{A}_p \mathbf{B}_{c,p:}$. Thus, the factor h cancels when the update is expressed in S5's native clock, and the implementation computes the following coefficients:

$$\begin{aligned}
\bar{\mathbf{A}} &= \mathrm{diag}\left(e^{\boldsymbol{\lambda} \odot \boldsymbol{\Delta}}\right), \\
\bar{\mathbf{B}} &= \boldsymbol{\Lambda}^{-1}(\bar{\mathbf{A}} - \mathbf{I})\mathbf{B}_c, \\
\bar{\mathbf{B}}_+ &= \bar{\mathbf{B}} + 5\,\mathrm{diag}(\boldsymbol{\Delta})\bar{\mathbf{A}}\mathbf{B}_c, \\
\bar{\mathbf{B}}_- &= -5\,\mathrm{diag}(\boldsymbol{\Delta})\bar{\mathbf{A}}\mathbf{B}_c.
\end{aligned} \tag{E.11}$$

It then advances $\mathbf{h}_{t+1} = \bar{\mathbf{A}}\mathbf{h}_t + \bar{\mathbf{B}}_+ \mathbf{x}_{t+1} + \bar{\mathbf{B}}_- \mathbf{x}_t$. The two taps span adjacent input samples; $5\Delta_p$ is the native-clock impulse coefficient produced by the shared $5h$ horizon, not a learned or mode-

dependent physical look-ahead. The delayed input starts at zero, and the direct $\mathbf{D} \odot \mathbf{x}_t$ feedthrough remains instantaneous. Within αP-S5, the two-tap operation adds no trainable parameters and leaves the recurrent transition, residual path, output projection, and scan structure unchanged.

The poles and basis transformation use the standard HiPPO-DPLR initialization. Before transformation, entries of $\widetilde{\mathbf{B}}$ and $\widetilde{\mathbf{C}}$ are LeCun-normal with standard deviation $1/\sqrt{P_{\mathrm{local}}}$, and $D_i \sim \mathcal{N}(0, 1)$. Log steps are initialized uniformly so that $\Delta_p \in [10^{-3}, 10^{-1}]$. In the reference αP-S5 configuration, real pole parts are clipped to at most -10^{-4} before forming $\boldsymbol{\alpha}$, whereas the native one-tap S5 configuration does not apply this clipping and uses the unscaled learned input matrix. Tables 2 and 7 therefore compare the complete αP-S5 construction with native S5; they do not isolate the second tap while holding the continuous-time input map and pole clipping fixed.

E.4 Optimizer and training

All Speech Commands runs use the 10-word subset of v0.02 with a stratified $70/15/15$ split (split seed 0) and feature-wise standardization from the training split. MFCC extraction uses 20 coefficients, a 200-sample FFT window, 64 mel bands, and a 100-sample hop, producing 161 frames. A checkpoint is saved each epoch, and test accuracy is read from the checkpoint with the highest validation accuracy. Prospective/non-prospective pairs use identical training settings and seeds. Neither Speech Commands nor Path-X uses data augmentation. The Speech Commands v0.02 archive includes a CC BY 4.0 license. Path-X was obtained from the official LRA release; the accompanying code repository is Apache 2.0, but the downloaded data archive does not include separate dataset-license terms.

RQF. RQF Speech Commands models use AdamW with a batch size of 32; the reported runs use seeds 42–46. Complex mixing weights, the input projection, and the readout use a learning rate of 10^{-3} and weight decay 10^{-4}; RQF pole parameters use a learning rate of 10^{-4} and zero weight decay. The global gradient norm is clipped at 1.0. Training runs for at most 300 epochs with cosine annealing to 10^{-6} and stops after 20 epochs without improved validation accuracy. Full BPTT retains the complete temporal graph and applies cross entropy with label smoothing 0.1 to the mean-pooled logits. Spatial-only backpropagation detaches the recurrent state after every step and applies the same smoothed cross entropy per time step; validation and test predictions average logits across time.

ORGaNICs. ORGaNICs models use AdamW with a learning rate of 10^{-3} and weight decay 10^{-4}, cross entropy with label smoothing 0.1, and gradient clipping at 1.0. The learning rate follows cosine annealing to 10^{-6} over a maximum of 300 epochs, with early stopping after 20 epochs without improved validation accuracy. Full BPTT runs use the mean-pooled prediction loss; spatial-only backpropagation detaches every recurrent state after each step and applies the loss at each time step.

αP-S5. S5 uses the same AdamW optimizer, base batch size, label smoothing, clipping, cosine schedule, early stopping, and checkpoint selection as the RQF Speech Commands runs; the reported runs use seeds 0–4. All S5 parameters, including $\boldsymbol{\Lambda}$, $\widetilde{\mathbf{B}}$, $\widetilde{\mathbf{C}}$, and the log steps, use the base 10^{-3} learning rate and 10^{-4} weight decay. The native-clock gain $\boldsymbol{\alpha} = -\mathrm{Re}(\mathrm{diag}(\boldsymbol{\Lambda}))$ introduces no optimizer state because it is derived from the poles.

Path-X. We use the $200{,}000$-example `pathfinder128/curv_contour_length_14` task with a seeded $80/10/10$ split (split seed 42). Pixels are divided by 255 and standardized by the training-split scalar mean and standard deviation. The six-layer, width-64 RQFs and sRQFs use Adam with $\epsilon = 10^{-4}$, zero weight decay, a learning rate of 3×10^{-3} for all parameter groups, a batch size of 64, cross entropy at each time step, and gradient clipping at 1.0. Training runs for 300 epochs with cosine annealing to zero; evaluation averages logits over time and reports the test result at the best-validation epoch. The five runs use seeds 0–4 for initialization and data-shuffling order while retaining the same data split. Path-X uses the same recurrence, log-space parameterization, split-ReLU, and real–imaginary MLP readout as the Speech Commands models. The optimizer schedule and the initialization duration T_{seq} were chosen for Path-X rather than carried over from the Speech Commands runs. The sRQFs differ from the RQFs only by the input gain of Appendix F; their γ and ω are not input-dependent.

F Selective RQFs (sRQFs)

An sRQF layer (Eq. (15)) has the same drive $\mathbf{z}_n^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{x}_n^{(\ell)}$ as an RQF layer and adds one input-dependent gain per filter, sampled at each step as

$$\boldsymbol{\kappa}_n^{(\ell)} \;=\; \phi\left(\mathbf{W}_\kappa^{(\ell)} \begin{bmatrix}\mathrm{Re}(\mathbf{y}_n^{(\ell-1)}) \\ \mathrm{Im}(\mathbf{y}_n^{(\ell-1)})\end{bmatrix}\right), \tag{F.1}$$

where $\mathbf{W}_\kappa^{(\ell)} \in \mathbb{R}^{n_\ell \times 2n_{\ell-1}}$ is a learned, bias-free matrix and ϕ is a positive function applied element-wise. Our experiments use the example $\phi(u) = 2\ \mathrm{sigmoid}(u)$, which lies in $(0, 2)$ and equals 1 at $u = 0$. The gain reads the preceding layer's output before the activation ρ, so it retains the sign information that the rectified layer input $\mathbf{x}_n^{(\ell)} = \rho(\mathbf{y}_n^{(\ell-1)})$ discards; at the first layer, it reads the network input. Because the gain matrix is initialized at zero and this ϕ satisfies $\phi(0) = 1$, $\boldsymbol{\kappa}_n^{(\ell)} = \mathbf{1}$ at initialization and each sRQF starts as the corresponding RQF.

When $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ are also selective, they are computed from the same features at each step:

$$\begin{aligned}
\boldsymbol{\gamma}_n^{(\ell)} &= \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})\ \mathrm{sigmoid}\left(\boldsymbol{\beta}^{(\ell)} + \mathbf{W}_\gamma^{(\ell)} \begin{bmatrix}\mathrm{Re}(\mathbf{y}_n^{(\ell-1)}) \\ \mathrm{Im}(\mathbf{y}_n^{(\ell-1)})\end{bmatrix}\right), \\
\boldsymbol{\omega}_n^{(\ell)} &= \boldsymbol{\omega}^{(\ell)} \odot \exp\left(c\ \tanh\left(\mathbf{W}_\omega^{(\ell)} \begin{bmatrix}\mathrm{Re}(\mathbf{y}_n^{(\ell-1)}) \\ \mathrm{Im}(\mathbf{y}_n^{(\ell-1)})\end{bmatrix}\right)\right),
\end{aligned} \tag{F.2}$$

where $\mathbf{W}_\gamma^{(\ell)}, \mathbf{W}_\omega^{(\ell)} \in \mathbb{R}^{n_\ell \times 2n_{\ell-1}}$ are initialized at zero and $\boldsymbol{\beta}^{(\ell)}$ is the element-wise logit of $(\boldsymbol{\gamma}^{(\ell)} - \gamma_{\min})/(\gamma_{\max} - \gamma_{\min})$, so that each sRQF starts as the corresponding RQF; $c > 0$ bounds each frequency multiplier to $[e^{-c}, e^c]$. The Selective Copying sRQFs use $\gamma_{\min} = 10^{-6}$, $\gamma_{\max} = 1 - 10^{-6}$, and $c = \ln 4$, and split the filters of each layer by initial frequency into four equal groups that share one frequency multiplier, so $\mathbf{W}_\omega^{(\ell)}$ has four distinct rows.

The selective quantities are held at $\boldsymbol{\kappa}_{n+1}^{(\ell)}$, $\boldsymbol{\gamma}_{n+1}^{(\ell)}$, and $\boldsymbol{\omega}_{n+1}^{(\ell)}$ over each interval $[t_n, t_{n+1}]$, the same right-endpoint convention as the ZOH drive. The gain then scales the input matrix of Eq. (14) to $\mathrm{diag}(\boldsymbol{\kappa}_{n+1}^{(\ell)})\,\mathrm{diag}(\boldsymbol{\alpha}_{n+1}^{(\ell)})\mathbf{W}^{(\ell)}$, and each interval is a fixed-parameter problem, so the derivations of Appendices B.1 and B.2 apply with the pole $\boldsymbol{\lambda}_{n+1} = -\boldsymbol{\alpha}_{n+1} + \iota(\mathbf{1} - \boldsymbol{\gamma}_{n+1}) \odot \boldsymbol{\omega}_{n+1}$ and $\boldsymbol{\alpha}_{n+1} = \boldsymbol{\gamma}_{n+1}/\tau$. Suppressing the layer index, the instantaneous and prospective ZOH updates of Eqs. (9) and (17) become

$$\mathbf{y}_{n+1} = \bar{\mathbf{A}}_{n+1} \odot \mathbf{y}_n + \boldsymbol{\kappa}_{n+1} \odot \bar{\mathbf{B}}_{n+1} \odot \mathbf{z}_{n+1}, \tag{F.3}$$

$$\mathbf{y}_{n+1} = \bar{\mathbf{A}}_{n+1} \odot \mathbf{y}_n + \boldsymbol{\kappa}_{n+1} \odot \left[\left(\bar{\mathbf{B}}_{n+1} + \boldsymbol{\gamma}_{n+1} \odot \bar{\mathbf{A}}_{n+1}\right) \odot \mathbf{z}_{n+1} - \boldsymbol{\gamma}_{n+1} \odot \bar{\mathbf{A}}_{n+1} \odot \mathbf{z}_n\right], \tag{F.4}$$

respectively, where $\bar{\mathbf{A}}_{n+1} = e^{\boldsymbol{\lambda}_{n+1} h}$ and $\bar{\mathbf{B}}_{n+1} = (\boldsymbol{\alpha}_{n+1}/\boldsymbol{\lambda}_{n+1}) \odot (\bar{\mathbf{A}}_{n+1} - \mathbf{1})$. With input-independent $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$, as on Path-X, $\bar{\mathbf{A}}_{n+1} = \bar{\mathbf{A}}_h$ and $\bar{\mathbf{B}}_{n+1} = \bar{\mathbf{B}}_h$; setting $\boldsymbol{\kappa}_n = \mathbf{1}$ as well recovers Eqs. (9) and (17). The current gain multiplies both prospective taps: the look-ahead acts on the drive, and the gain scales the resulting input contribution, so the lagged drive $\mathbf{z}_n$ is not multiplied by $\boldsymbol{\kappa}_n$; likewise, both taps use the current $\boldsymbol{\gamma}_{n+1}$ and $\bar{\mathbf{A}}_{n+1}$. The gain alone changes how strongly each input is written into the state but not the pole, so a gain-only sRQF keeps the tuning frequency and bandwidth analyzed in Appendix A.2; selective $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ instead move the pole at every step.

Parallel scan. The selective quantities depend only on the preceding layer's output and not on the layer's own state, so once that output is available, the transitions $\mathbf{a}_{n+1} = \bar{\mathbf{A}}_{n+1}$ and the input terms $\mathbf{d}_{n+1}$ of Eqs. (F.3) and (F.4) can be computed for all samples in parallel. Both updates then take the form $\mathbf{y}_{n+1} = \mathbf{a}_{n+1} \odot \mathbf{y}_n + \mathbf{d}_{n+1}$, a diagonal linear recurrence whose coefficients may vary with n. Pairs $(\mathbf{a}, \mathbf{d})$ compose associatively as $(\mathbf{a}_2, \mathbf{d}_2) \circ (\mathbf{a}_1, \mathbf{d}_1) = (\mathbf{a}_2 \odot \mathbf{a}_1,\ \mathbf{d}_2 + \mathbf{a}_2 \odot \mathbf{d}_1)$ whether or not $\mathbf{a}$ varies with n, so the parallel scan for diagonal linear recurrences applies, with $O(Tn_\ell)$ work and $O(\log T)$ parallel depth [26–28]. This is the same property that lets Mamba compute its time-varying selective SSMs with a parallel scan [23]. As in Mamba, however, the time-varying transition removes the equivalence to a convolution [23]: FFT convolution remains available only for gain-only sRQFs, whose fixed $\bar{\mathbf{A}}_h$ makes $\mathbf{y}$ a fixed-kernel causal convolution of $\mathbf{d}$. Each selective projection adds at most $O(Tn_\ell n_{\ell-1})$ work, the same order as the dense drive; $\mathbf{W}_\kappa^{(\ell)}$ and $\mathbf{W}_\gamma^{(\ell)}$ each add $2n_\ell n_{\ell-1}$ real

parameters per layer, and the grouped $\mathbf{W}_{\omega}^{(\ell)}$ adds $8n_{\ell-1}$. All sRQF results use full BPTT, which differentiates through the selective quantities; the spatial-only analysis of Appendix C does not cover them.

F.1 Selective Copying protocol

The sRQFs make $\boldsymbol{\gamma}$, $\boldsymbol{\omega}$, and the input gain input-dependent (Appendix F). Each sequence has $4{,}096$ one-hot inputs over a vocabulary of 16 symbols: blank token 0, data tokens 1–14, and recall marker 15. Sixteen distinct positions among the first $4{,}080$ are drawn uniformly and sorted, and each receives an independent uniformly drawn data token, so token values can repeat; all other context positions are blank. The final 16 inputs are recall markers, and their targets are the data tokens in input order. This follows the sequence length, vocabulary size, and number of data tokens of Gu and Dao [23]. Each epoch draws $64{,}000$ fresh training sequences; validation and test use separate fixed sets of $1{,}024$ sequences. The loss is the cross entropy averaged over the 16 recall positions, and the reported metric is recall-token accuracy, the fraction of recall tokens predicted correctly.

The RQFs and sRQFs both have four layers of 32 complex filters and no input projection: the first complex matrix maps the one-hot input directly to the filter bank. Between layers, the activation applies split-ReLU followed by divisive normalization across the filters of the layer:

$$[\rho(\mathbf{y})]_i = \frac{\rho_0(y_i)}{\sqrt{\sigma^2 + \frac{1}{n}\sum_{j=1}^{n} |\rho_0(y_j)|^2}},$$

where ρ_0 is the split-ReLU of Eq. (8); $n = 32$ is the layer width; and $\sigma = 0.3$. There are no residual connections or dropout. At each recall position, the readout concatenates the real and imaginary parts of the last layer's state and applies a $64 \to 128 \to 16$ MLP with a ReLU hidden layer. The RQF parameterization and initialization follow Eqs. (E.5)–(E.7) with $h = 1$, $\tau = 5h$, $T_{\mathrm{seq}} = 4{,}096h$, $F = 1$, and $G_\rho = 1$ both for the one-hot input and after divisive normalization. The readout's hidden layer is He-normal with zero bias, and its logit layer is normal with standard deviation 10^{-3} and zero bias. With complex parameters counted twice, the RQF and sRQF have $17{,}808$ and $33{,}040$ parameters, respectively.

All models are trained with full BPTT using Adam with a learning rate of 10^{-3} for all parameters, $\epsilon = 10^{-4}$, and no weight decay; a batch size of 64; global gradient-norm clipping at 1.0; and no label smoothing, learning-rate schedule, or early stopping. Training runs for 600 epochs of 1,000 updates. Validation accuracy is computed after every epoch, and test accuracy is read from the checkpoint with the highest validation accuracy. Each configuration is trained with five seeds, and the seed sets both the initialization and the data generation. Matched runs differ only in the input rule and in whether the gain, $\boldsymbol{\gamma}$, and $\boldsymbol{\omega}$ are input-dependent.