Spectral Quantum Memory for Implicit Neural Representations

Daniel A. Moroz-Broitman
Department of Computer Science
Ben Gurion University
morozd@post.bgu.ac.il

Eliya Nachmani
Department of Electrical Engineering
Ben Gurion University
eliyanac@bgu.ac.il

Abstract

Implicit neural representations (INRs) model signals as continuous functions from coordinates to values, but their performance depends strongly on how spectral capacity is allocated. Yet most Quantum INR (QINR) architectures treat spectral components as independent outputs of separate quantum blocks. We introduce the *Spectral Quantum Memory Model* (SQMM), a memory coupled quantum INR in which sequential hybrid data reuploading layers (bands) communicate through a shared coherent quantum memory register. Each band reads the memory, computes on its own coordinate upload, and writes part of its state back, so later bands can build on the computation of earlier ones. When memory read and write layers are input independent, we prove that the memory state forms an operator valued Fourier series whose support is contained in the Minkowski sum of the initial memory spectrum and the local band spectra. Downstream bands then condition their Fourier coefficients on the spectral content stored by earlier bands, and the memory coupling creates no new primitive frequencies. This theory motivates a *spectral memory alignment* loss that encourages memory to encode the spectrum expressed by the deployed prefix of the model. Across audio and image reconstruction benchmarks, SQMM improves over compact QINR and INR baselines. Code is provided in: https://github.com/Kiiwii10/SQMM

1 Introduction

Implicit neural representations (INRs) model signals as continuous coordinate to value functions, enabling compact representations for images, audio (Figure 1), video, geometry, radiance fields, and physical signals [81, 51, 46, 67, 14, 70]. Their main bottleneck is spectral. Standard MLPs learn low frequency structure before high frequency detail [54, 82], so INR performance depends on how frequency content is exposed, selected, and composed [6]. This has motivated Fourier features [70], periodic activations [67], multiplicative filters [16], band limited and multiscale coordinate networks [39, 61], wavelet INRs [57], and adaptive periodic bases [40, 44]. These works show that spectral organization is central to INR capacity.

Quantum implicit neural representations (QINRs) make this spectral structure explicit because data reuploading circuits compute finite Fourier series. Data reuploading repeatedly injects classical inputs into variational quantum circuits [47, 53, 22]. Their accessible frequencies are determined by encoding gates, while trainable circuit parameters determine the coefficients [60, 9, 83, 79, 3, 45]. QIREN formulates QINRs as quantum generalizations of Fourier neural networks [85], and QVF extends QINRs toward quantum centric visual fields through neural amplitude encoding and projective measurement [77]. Quantum recurrent models and quantum LSTMs use hidden quantum states or variational cells as memory mechanisms [5, 12]. Existing QINRs, however, do not characterize how memory qubits share spectral information across coordinate frequency modules.

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

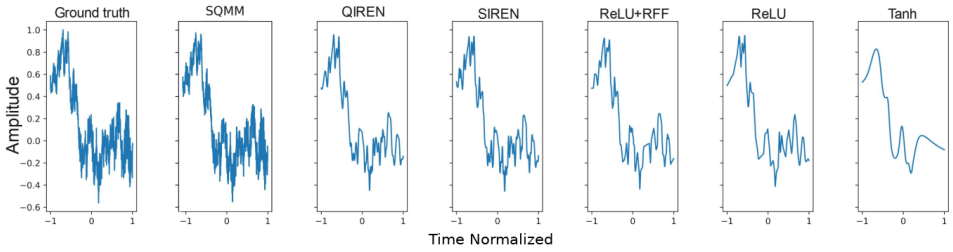

Figure 1: Sound representation results adapted from QIREN [85], showing normalized amplitude over normalized time.

In existing multiband QINRs, the bands operate largely independently, and what one band computes reaches the others only through the classical decoder. A shared memory lets information produced by earlier bands influence later ones. This paper introduces the Spectral Quantum Memory Model (SQMM), a memory coupled QINR built from sequential data reuploading circuits organized into hybrid quantum and classical bands. Each band reads a shared unmeasured memory register, performs local spectral computation, and writes back before the next band. SQMM therefore propagates coherent spectral context across bands, allowing later bands to condition coefficient allocation on earlier spectral computation instead of relearning each band independently.

When memory read and write layers are independent of the input coordinate, each memory coupled band remains in the data reuploading model class. The reduced memory state is an operator valued Fourier series whose support is contained in the Minkowski sum of the initial memory spectrum and the band spectra. Downstream readout coefficients arise from convolutions between the memory spectrum and local band spectra. Thus memory does not introduce arbitrary primitive frequencies. It preserves the Fourier properties of QINRs while expanding spectral context between bands.

This view motivates a spectral memory alignment loss. The loss aligns memory with an estimate of the accumulated spectral occupancy of the deployed prefix contribution. The supervised loss trains the predictor, while the alignment loss calibrates memory as a readable spectral summary of the computation so far. Downstream bands can still reuse, refine, or correct upstream frequencies.

The contributions are fourfold. First, SQMM introduces a memory coupled QINR that propagates coherent spectral context across sequential data reuploading bands. Second, the theory proves that memory preserves the Fourier properties of data reuploading circuits, since the memory state remains an operator valued Fourier series whose support is contained in the Minkowski sum of the band spectra. Third, a spectral memory alignment loss trains memory to summarize the deployed prefix spectrum. Fourth, SQMM improves on the reconstruction results of quantum and classical INR baselines on audio and image representation tasks under small parameter budgets.

2 Related Work

Implicit neural representations and spectral coordinate networks. Coordinate based neural representations originate in compositional pattern producing networks [69] and now support continuous representation of signals, shapes, scenes, and fields [81], including shape templates [21], continuous shape distributions [86], signed distance functions [51], scene representation networks [66], local implicit grids [29], neural radiance fields [46], image generation and compression [68, 2, 14, 10, 49], time indexed modeling [80], and scientific modeling [28]. Their central limitation is spectral bias, since standard networks tend to fit low frequencies before high frequency detail [54, 82, 4, 70]. This has motivated spectral INR designs based on Fourier features [70], sinusoidal and nonstandard activations [50, 67, 55], multiplicative filters [16], band limited and multiscale networks [39, 61], wavelet INRs [57], adaptive periodic bases [40, 44], Fourier reparameterized training [62], and, more recently, content aware Fourier Chebyshev encodings (CAFE) [30], multi scale sine activations (MSA-INR) [24], frequency and locality aware representations (FLAIR) [33], robust frequency tuning for sinusoidal networks (TUNER) [48], variance informed initialization (VI3NR) [34], and inductive gradient adjustment against spectral bias (IGA) [63]. These methods shape the spectrum of a single classical network through its encoding, activation, initialization, or gradient, and none of them passes a state between separately computed frequency modules. SQMM follows this spectral view, but

studies spectral sharing in quantum data reuploading circuits where the finite Fourier structure can be analyzed explicitly.

Parameterized quantum models and data reuploading. Parameterized quantum circuits are trainable models for supervised learning, quantum feature maps, generative modeling, reinforcement learning, and hybrid quantum classical computation [43, 47, 15, 13, 25, 7, 65, 58, 8]. Hybrid quantum neural networks combine classical processing with quantum circuits that implement trainable feature spaces, kernels, or intermediate representations [59, 23], with applications in sequence modeling, classification, reinforcement learning, generative modeling, and architecture search [12, 38, 11, 42, 20, 76]. Data reuploading repeatedly injects classical variables into variational quantum circuits and improves expressivity even with small registers [53, 22]. Angle encoded quantum models compute finite Fourier series whose accessible frequencies are determined by encoding generator spectra, while trainable parameters determine the coefficients [60]. This Fourier view has been extended to single qubit native models [83], multidimensional Fourier series [9], universal approximation in quantum enhanced feature spaces [23], structured and exponential encodings [64, 75], exact realized spectra and coefficients [79], coefficient constraints from encoding redundancy [45], and gradient profiles in deep reuploading circuits [3]. SQMM extends this Fourier framework from isolated reuploading circuits to stacked memory coupled bands.

Quantum implicit neural representations. The closest prior work is QIREN, which connects data reuploading circuits to INRs by treating parameterized quantum circuits as quantum analogues of Fourier feature networks [85]. QIREN uses repeated coordinate encodings to produce quantum Fourier features decoded by classical layers for signal representation, image super resolution, and image generation. QVF moves toward quantum centric visual fields using neural amplitude encoding and projective measurement over entangled ansatzes for 2D and 3D signals [77]. SQMM is complementary to both models. It retains the data reuploading Fourier view of QIREN but couples its bands through a shared coherent memory, and it differs from QVF by emphasizing spectral communication across reuploading bands rather than amplitude encoding or direct visual field measurement. Prior Fourier analyses of data reuploading circuits characterize circuits with fixed incoming quantum states [60, 83, 9]. QIREN connects that theory to INRs, but it measures each quantum block and passes a classical feature vector to the next hybrid layer, so its analysis characterizes the Fourier form of an individual circuit in its immediate input. SQMM instead keeps a coherent memory state across sequential bands, and our analysis describes how the operator valued spectrum of that state conditions downstream quantum features throughout the model.

Memory, recurrence, and auxiliary spectral supervision. Memory and recurrence have appeared in quantum learning through recurrent quantum neural networks, quantum LSTM style architectures, and quantum reservoir computing [5, 12, 19]. Classical models also use inter stage coupling through residual or stagewise refinement, where later modules correct earlier computations [26, 18]. In INRs, band limited and multiscale coordinate networks explicitly organize frequency bands [39, 61], while spectral regularization and auxiliary supervision shape learned representations during training [62, 37]. Quantum recurrent models use hidden quantum states as memory, but they do not characterize the coordinate frequency content of that state or its effect on downstream QINR coefficients. SQMM differs from these classical mechanisms because the coupled state is not a residual stream, recurrent hidden vector, or auxiliary prediction head. It is a quantum density matrix with operator valued Fourier coefficients that remains readable by downstream quantum bands.

3 Background

We consider coordinate based signal representation. Coordinates are $x \in \mathbb{R}^d$, targets are $y \in \mathbb{R}^{d_y}$, and the model learns a continuous map $x \mapsto y$. SQMM first maps x to a band specific uploaded variable

$$\gamma_i(x) = A_i \mathrm{PE}(x) + b_i \in \mathbb{R}^{g_i}, \qquad A_i \in \mathbb{R}^{g_i \times p}, \qquad b_i \in \mathbb{R}^{g_i}. \tag{1}$$

Here $\mathrm{PE} : \mathbb{R}^d \to \mathbb{R}^p$ is a fixed positional or Fourier feature encoding [73, 70, 46]. SQMM uses quantum data reuploading circuits inside this coordinate model. A register is a group of qubits treated as one subsystem. Band i has a local data register D_i with q_i qubits, and all bands share a memory register M with m qubits. The fixed new data state is η_{D_i}, and the memory state after band i is a density matrix $\sigma_M^{(i)}(x)$, namely a positive semidefinite matrix with trace one. A quantum operation

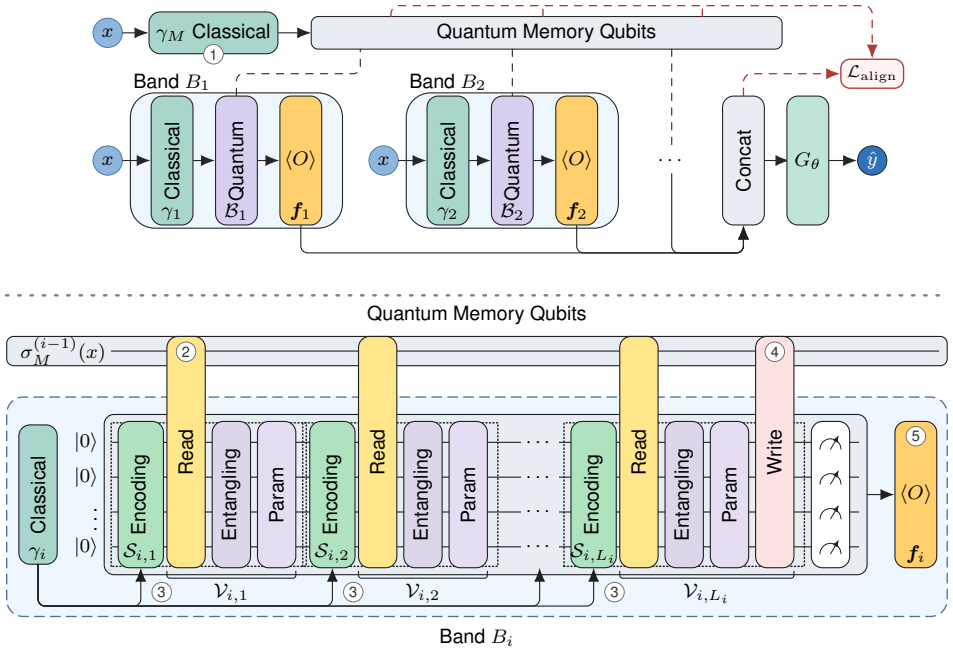

Figure 2: SQMM architecture. Coordinates are mapped to band uploads $\gamma_i(x)$ and to the memory upload $\gamma_M(x)$, both of the form of Eq. (1). Sequential quantum bands share the memory register M and emit Pauli expectation features, which a classical decoder combines. Dashed red arrows mark the training only spectral alignment loss. Numbers follow the Intuition paragraph.

is written as a channel. The band channel $\mathcal{B}_i(\gamma_i(x))$ acts on the joint register $D_i \otimes M$. Table 7 in Appendix 8 collects the notation.

We use $\mathrm{Tr}[\cdot]$ for trace and $\mathrm{Tr}_{D_i}[\cdot]$ for partial trace over the local data register. An observable O is a Hermitian matrix that represents a measurable quantity, and $\mathrm{Tr}[O\rho]$ is its expectation value in state ρ. In SQMM these expectations are used as classical features.

A finite data reuploading circuit has a finite Fourier expansion in the uploaded variable because input dependent gates are generated by finite dimensional Hermitian operators [60, 9, 83]. For a generic uploaded variable $z \in \mathbb{R}^{g_i}$, the local upload support of band i is denoted by $\Omega^{\mathrm{loc}}_{i,\gamma}$, with terms $e^{\mathrm{i}\langle \omega_\gamma, z\rangle}$ for $\omega_\gamma \in \Omega^{\mathrm{loc}}_{i,\gamma}$. After substituting $z = \gamma_i(x)$, the analysis uses the induced coordinate support $\Omega^{\mathrm{loc}}_{i,x}$. This support is exact when the upload is affine in the raw coordinate. With fixed sinusoidal positional encodings or nonlinear angle maps, as commonly used in coordinate networks and QINRs [70, 46, 85], the composed coordinate dependence is analyzed either in the lifted encoding coordinates or through the finite grid projection used by the spectral alignment loss. Thus the theory applies to an exact finite coordinate expansion when available, and otherwise to the chosen finite truncation. For sets $U, V \subset \mathbb{R}^d$, let $U + V = \{u + v \mid u \in U,\ v \in V\}$ denote their Minkowski sum. Since memory is a density matrix, its Fourier coefficients are matrices rather than scalar hidden features. Additional quantum notation and the upload to coordinate support conversion are given in Appendix 9.

4 Spectral Quantum Memory Model

SQMM (Figure 2) is a hybrid quantum and classical INR for learning $x \mapsto y$. It consists of n sequential data reuploading bands connected by a shared memory register. Each band emits expectation valued classical features, while the memory state is updated and passed to later bands. The memory read and write layers are input independent, so they change Fourier coefficients but do not create new primitive coordinate frequencies.

Intuition. Without memory, each band of a multiband QINR sees only the coordinate, and its features meet those of other bands only in the classical decoder. SQMM adds a small register of m

unmeasured memory qubits that is carried from band to band. For a coordinate x, the memory is first prepared from the coordinate by a one time upload $\gamma_M(x)$ (1). Each band then reads the memory through input independent couplings (2), uploads its own coordinate encoding $\gamma_i(x)$ several times (3), and writes part of its local state back into memory (4) before its local qubits are discarded. The band emits Pauli expectation features, and the updated memory is passed to the next band (5). (1)-(5) refer to marks in Figure 2. At inference only the band features are estimated, and the memory is never measured. Section 4.2 shows what this pathway can carry, and Section 4.3 introduces a loss that trains the memory to resemble the spectrum accumulated by the preceding bands.

4.1 Architecture

Band i starts with a new data state η_{D_i} and incoming memory $\sigma_M^{(i-1)}(x)$. Its uploaded variable is $\gamma_i(x)$ from Eq. (1). The band channel is

$$\mathcal{B}_i(\gamma_i(x)) = \mathcal{V}_{i,L_i+1} \circ \mathcal{S}_{i,L_i}(\gamma_i(x)) \circ \mathcal{V}_{i,L_i} \circ \cdots \circ \mathcal{S}_{i,1}(\gamma_i(x)) \circ \mathcal{V}_{i,1}. \tag{2}$$

The layers $\mathcal{S}_{i,\ell}$ are the only input dependent layers. The layers $\mathcal{V}_{i,\ell}$ are trainable and input independent, and include local gates, entangling gates, memory read couplings, memory write couplings, and memory only processing. In the deployed circuit, $\mathcal{V}_{i,1}$ holds the memory read couplings, each upload is followed by a local entangling layer, and the closing block $\mathcal{V}_{i,L_i+1}$ holds the last local entangling layer together with the memory write couplings. Therefore memory coupling preserves the data reuploading form on $D_i \otimes M$. The joint post band state is

$$\xi_i(x) = \mathcal{B}_i(\gamma_i(x)) \left(\eta_{D_i} \otimes \sigma_M^{(i-1)}(x) \right). \tag{3}$$

The updated memory is obtained by discarding the local data register,

$$\sigma_M^{(i)}(x) = \mathrm{Tr}_{D_i} \left[\xi_i(x) \right]. \tag{4}$$

Band i emits Pauli expectation features

$$\boldsymbol{f}_i(x) = \left(\mathrm{Tr}[(O_{i,1} \otimes I_M)\xi_i(x)], \ldots, \mathrm{Tr}[(O_{i,d_{f,i}} \otimes I_M)\xi_i(x)] \right) \in \mathbb{R}^{d_{f,i}}. \tag{5}$$

These features are ensemble expectations. The downstream memory is not conditioned on local measurement outcomes. Hardware estimation uses repeated circuit executions, and the readout semantics are formalized in Appendix 11. Let $d_F = \sum_{i=1}^{n} d_{f,i}$, let

$$\boldsymbol{f}(x) = [\boldsymbol{f}_1(x)^\top, \ldots, \boldsymbol{f}_n(x)^\top]^\top \in \mathbb{R}^{d_F}, \tag{6}$$

and let $G_\theta : \mathbb{R}^{d_F} \to \mathbb{R}^{d_y}$ be a classical decoder. The prediction is

$$\hat{y}(x) = G_\theta(\boldsymbol{f}(x)). \tag{7}$$

4.2 Spectral Interpretation

A data reuploading circuit computes a finite Fourier series whose frequencies are fixed by its encoding gates. The result below states that this remains true for a band that receives a coordinate dependent memory state. Every frequency that leaves a band, in its features or in the memory passed on, is the sum of a frequency already present in the incoming memory and a frequency introduced by the band's own uploads. The input independent layers of the band decide how strongly each such frequency appears. The memory is therefore a channel through which later bands can reuse and reweight the spectral content of earlier bands.

The spectral analysis expands each band on the coordinate support $\Omega_{i,x}^{\mathrm{loc}}$. For a fixed band i, assume the incoming memory has the operator valued Fourier expansion

$$\sigma_M^{(i-1)}(x) = \sum_{\mu \in \Omega_{M,x}^{(i-1)}} e^{\mathrm{i}\langle \mu, x \rangle} \tau_\mu^{(i-1)}, \qquad \tau_\mu^{(i-1)} \in \mathbb{C}^{2^m \times 2^m}. \tag{8}$$

Here $\tau_\mu^{(i-1)}$ is the matrix coefficient of the memory state at coordinate frequency μ. Write the coordinate expansion of the band channel as

$$\mathcal{B}_i(\gamma_i(x)) = \sum_{\nu \in \Omega_{i,x}^{\mathrm{loc}}} e^{\mathrm{i}\langle \nu, x \rangle} \mathcal{B}_{i,\nu}. \tag{9}$$

Lemma 1 (Spectral propagation through memory)**.** *Under Eqs.* (8) *and* (9)*, the outgoing memory and emitted features of band i have no coordinate frequencies outside $\Omega_{M,x}^{(i-1)} + \Omega_{i,x}^{\mathrm{loc}}$. In particular,*

$$\Omega_{M,x}^{(i)} \subseteq \Omega_{M,x}^{(i-1)} + \Omega_{i,x}^{\mathrm{loc}}, \qquad \Omega_{f_i,x} \subseteq \Omega_{M,x}^{(i-1)} + \Omega_{i,x}^{\mathrm{loc}}. \tag{10}$$

Input independent read, write, and memory only layers can change Fourier coefficients but cannot enlarge these supports (Proposition 1 and Lemma 2).

Proof. The main point is the product rule for Fourier phases. Substituting Eqs. (8) and (9) into Eq. (4) gives

$$\sigma_M^{(i)}(x) = \sum_{\omega} e^{\mathrm{i}\langle \omega, x\rangle} \tau_\omega^{(i)}, \qquad \tau_\omega^{(i)} = \sum_{\mu+\nu=\omega} \mathrm{Tr}_{D_i}\left[\mathcal{B}_{i,\nu}\left(\eta_{D_i} \otimes \tau_\mu^{(i-1)}\right)\right]. \tag{11}$$

Thus an outgoing memory frequency ω can appear only when $\omega = \mu + \nu$, where μ is already present in memory and ν is introduced by the current band. The same argument gives the emitted feature spectrum. For feature component a, define

$$\ell_{i,a,\nu}(X) = \mathrm{Tr}\left[(O_{i,a} \otimes I_M)\mathcal{B}_{i,\nu}(\eta_{D_i} \otimes X)\right]. \tag{12}$$

This map is linear in the memory side matrix X. Applying it to Eq. (8) gives

$$f_{i,a}(x) = \sum_{\omega \in \Omega_{M,x}^{(i-1)} + \Omega_{i,x}^{\mathrm{loc}}} c_{\omega,a}^{(i)} e^{\mathrm{i}\langle \omega, x\rangle}, \qquad c_{\omega,a}^{(i)} = \sum_{\mu+\nu=\omega} \ell_{i,a,\nu}\left(\tau_\mu^{(i-1)}\right). \tag{13}$$

This proves the support claim. □

The lemma shows that memory does not introduce arbitrary primitive frequencies. It lets downstream bands allocate coefficients conditional on an operator valued spectral summary of upstream computation. The one time memory upload $\gamma_M(x)$ before the first band (Figure 2) gives the base support $\Omega_{M,x}^{(0)}$, which is $\{0\}$ when memory is initialized independently of x. Applying Eq. (10) recursively bounds the memory support after band i by $\Omega_{M,x}^{(0)} + \sum_{j=1}^{i} \Omega_{j,x}^{\mathrm{loc}}$ (Proposition 2 and Corollary 2 in Appendix 10).

4.3 Spectral Memory Alignment Loss

Lemma 1 says that memory can carry a spectral summary of the prefix computation. The alignment loss makes this summary readable. It compares two spectra on a finite grid,

$$\Omega_{\mathrm{grid}} = \{\bar{\omega}_1, \ldots, \bar{\omega}_{N_\omega}\} \subset \mathbb{R}^d. \tag{14}$$

Bars denote grid frequencies. Hats denote coefficients estimated from sampled coordinates.

Feature spectrum. For sampled coordinates $x_1, \ldots, x_T$, define

$$\Phi_{th} = e^{\mathrm{i}\langle \bar{\omega}_h, x_t\rangle}, \qquad \widehat{a}(u) = (\Phi^\dagger \Phi + \varepsilon I_{N_\omega})^{-1} \Phi^\dagger u. \tag{15}$$

All signals in this paper are sampled on regular grids. We therefore take Ω_{grid} from the discrete Fourier frequencies of the sampling grid and set $\varepsilon = 0$, so Eq. (15) reduces to the discrete Fourier transform scaled by $1/T$ and is computed with an FFT (Appendix 12). Applying this estimator to each feature component of band j gives $\widehat{\boldsymbol{c}}_{j,h}^F = (\widehat{c}_{j,1,h}^F, \ldots, \widehat{c}_{j,d_{f,j},h}^F) \in \mathbb{C}^{d_{f,j}}$, and the prefix feature occupancy is

$$P_i^F(\bar{\omega}_h) = \sum_{j=1}^{i} \|\widehat{\boldsymbol{c}}_{j,h}^F\|_2^2. \tag{16}$$

This records which grid frequencies have been expressed by the emitted features of bands $1, \ldots, i$, before the final decoder.

Memory spectrum. The memory is probed with the unnormalized m qubit Pauli string basis $\mathcal{Q}_m = \{I, X, Y, Z\}^{\otimes m}$, with $|\mathcal{Q}_m| = 4^m$ and $\mathrm{Tr}(Q_s Q_{s'}) = 2^m \mathbf{1}\{s = s'\}$. For each $Q_s \in \mathcal{Q}_m$, the

scalar memory probe is $\chi^M_{i,s}(x) = \mathrm{Tr}[Q_s \sigma^{(i)}_M(x)]$. Estimating its Fourier coefficients with Eq. (15) and collecting them at $\bar{\omega}_h$ gives $\widehat{\boldsymbol{c}}^M_{i,h} = (\widehat{c}^M_{i,s,h})_{Q_s \in \mathcal{Q}_m} \in \mathbb{C}^{4^m}$. The memory occupancy spectrum is

$$P^M_i(\bar{\omega}_h) = 2^{-m} \sum_{Q_s \in \mathcal{Q}_m} |\widehat{c}^M_{i,s,h}|^2. \tag{17}$$

The factor 2^{-m} follows from the Pauli trace normalization and cancels under the normalization of Eq. (18) up to a rescaling of δ. If the finite expansion is exact and the full Pauli basis is used, then $P^M_i(\bar{\omega}_h) = \|\tau^{(i)}_{\bar{\omega}_h}\|^2_F$ up to finite grid estimation error. The derivation is given in Appendix 12.

Alignment objective. For any nonnegative spectrum $R : \Omega_{\mathrm{grid}} \to \mathbb{R}_{\geq 0}$, define

$$\mathcal{N}_\delta R(\bar{\omega}_h) = \frac{R(\bar{\omega}_h) + \delta}{\sum_{g=1}^{N_\omega} R(\bar{\omega}_g) + N_\omega \delta}. \tag{18}$$

With stop gradient sg, the alignment loss is

$$\mathcal{L}_{\mathrm{align}} = \sum_{i=1}^{n-1} \sum_{h=1}^{N_\omega} \left(\mathcal{N}_\delta P^M_i(\bar{\omega}_h) - \mathrm{sg}\left[\mathcal{N}_\delta P^F_i(\bar{\omega}_h) \right] \right)^2. \tag{19}$$

The sum stops at $n-1$, since the final memory state is not used by a downstream band. The stop gradient keeps the prefix feature spectrum as a target and trains the memory state to match it. For supervised loss ℓ_{sup}, task loss $\mathcal{L}_{\mathrm{task}} = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \ell_{\mathrm{sup}}(\hat{y}(x), y)$, the full training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{align}} \mathcal{L}_{\mathrm{align}}. \tag{20}$$

The supervised term trains the predictor. The auxiliary term encourages the memory after each deployed prefix to store the resolved spectrum of that prefix. Frequencies outside Ω_{grid} are not matched directly and are treated only through the finite grid projection. In our experiments the loss is averaged over boundaries and grid frequencies, weighted by λ_{align}, and applied only while bands are being added (Appendix 12).

5 Experiments

We evaluate SQMM as a compact coordinate based reconstruction model for audio and images. The comparison targets the low parameter regime used by QIREN [85], which is the closest quantum INR baseline. All experiments were run on internal CPU cluster nodes with a mixed state simulator using PennyLane [8] and PyTorch [52]. Unless stated otherwise, models were trained with Adam [31] for 600 epochs using learning rate 0.03, betas $(0.9, 0.999)$, epsilon 10^{-8}, and supervised MSE loss.

Datasets. The datasets cover single signal reconstruction, dataset level audio reconstruction, and image reconstruction. The classic INR setting follows QIREN [85] and uses a Bach cello excerpt together with Astronaut, Camera, and Coffee images from scikit image [72]. Dataset level audio uses GTZAN [71], UrbanSound8K [56], and VCTK [74]. To control simulation cost, each selected audio dataset uses 10 samples per class. Since only VCTK has more than 10 speakers (classes), the first 10 are chosen. CIFAR10 [35] is used for grayscale image reconstruction on 100 images. For larger images we use all 24 Kodak images [17] and 10 DIV2K images [1], in RGB.

Baselines. Baselines include classical coordinate networks and quantum INR models. We compare against ReLU, Tanh, ReLU with random Fourier features (R+RFF) [70], SIREN [67], FINER [40], WIRE [57], KAN [41], NeRF [46], QIREN [85], and QVF [77], and against the recent INRs CAFE [30], MSA-INR [24], FLAIR [33], TUNER [48], VI3NR [34], and IGA [63]. Every classical baseline has one configuration for audio and one for images. Each uses the default settings of its official implementation, with network width and depth selected inside a fixed parameter budget and the learning rate selected from $[3 \cdot 10^{-2}, 10^{-3}]$. Configurations and the two baselines with adjusted settings, MSA-INR and the encoder scaled CAFE (enc.), are given in Appendix 14.5.

Metrics. Metrics are reported as reconstruction error unless otherwise stated. Audio and classic INR tables report MSE with the scaling shown in each caption. Audio analysis also reports log spectral distance. CIFAR10 reports MSE, SSIM [78], and LPIPS [84]. Params denotes the number of model parameters. All reconstruction experiments are per instance INR fitting experiments.

Table 1: MSE↓ ($\times 10^{-3}$) on sound and image representation tasks, mean ± SD. * marks our model and its subscript the number of shots. QVF has no audio configuration. Best runs of all models are in Table 15. Best in **bold**.

	SQMM*	SQMM$^\star_{2048}$	QIREN	QVF	IGA	FINER	SIREN	MSA-INR	CAFE (enc.)
Params	642	642	649		673	691	673	673	703
Cello	**0.48±0.08**	0.76±0.06	6.79±0.35		4.87±0.01	3.73±0.44	5.02±0.08	7.04±1.23	3.60±0.60
Params	654	654	657	671	687	697	687	697	709
Astronaut	**1.12±0.10**	2.86±0.53	5.90±0.49	12.61±1.65	2.01±0.53	2.80±0.31	2.15±0.22	2.03±0.28	10.46±5.92
Camera	**0.38±0.08**	1.27±0.16	2.19±0.31	4.43±1.38	1.07±0.46	1.96±0.62	0.76±0.47	1.14±0.14	3.36±1.41
Coffee	**0.78±0.05**	1.79±0.16	2.52±0.67	14.38±6.80	1.63±0.42	1.85±0.31	2.26±0.40	1.37±0.19	5.17±1.06

Configuration. SQMM is kept near the QIREN parameter scale and uses five bands with $[2, 3, 4, 4, 5]$ reuploads and two data qubits per band. For audio, SQMM encodes the raw coordinate with 16 dyadic sinusoidal frequencies uploaded through R_Y rotations. For images, SQMM uses the same circuit with 8 consecutive frequencies per axis uploaded through R_Y and R_X rotations, and RGB images only widen the output layer. All bands share a two qubit memory register, initialized from the same encoding, with controlled R_Y reads and XY writes. Every band is read out through its Pauli Z and ZZ expectations, and the decoder is an MLP with one hidden layer of width 5 and GELU activation. Full configurations and training details are given in Appendix 14.5.

5.1 Base Benchmark

The classic INR benchmark follows the QIREN protocol. The cello signal is sampled at 1000 points with stride 5, normalized in amplitude, and indexed by timestamps evenly spaced in $[-1, 1]$. The image task uses Astronaut, Camera, and Coffee images cropped and downsampled to 32×32. This setting tests whether memory coupling improves the small signal regime used in prior quantum INR comparisons. For this benchmark only, models trained for 300 epochs for Cello, Camera and Coffee to match QIREN's benchmark. The column SQMM$_{2048}$ estimates every expectation from 2048 measurement shots in training and evaluation (Appendix 14.4). Figure 1 shows the Cello reconstruction.

SQMM obtains the best reconstruction error on each sound and image representation task in Table 1. With 2048 shots SQMM stays best on the audio sample, Cello. The comparison is most informative against QIREN, FINER, and QVF because they represent the closest quantum INR baseline, a strong adaptive classical spectral baseline, and a recent quantum visual field baseline.

5.2 Model analysis

The mechanism ablation isolates the contribution of memory access and spectral alignment. Table 2 compares SQMM on Cello with a no memory control that gives every band two further qubits that act locally and are not shared between bands, which matches the qubit count and parameter budget of SQMM. The no memory control has a higher error than SQMM. On the full classic benchmark (Table 9), adding memory with read write substantially improves reconstruction. Directly supervising the final output spectrum with a control loss $\mathcal{L}_{\text{out}}$ (Appendix 14.3) rather than the memory spectrum gives a similar error, while aligning the memory spectrum gives the best result at the same parameter count.

Table 2: Mechanism ablation on Cello, MSE ($\times 10^{-3}$), mean ± SD. No memory, local qubits replaces the shared memory with two local qubits in every band and shares nothing between bands. All samples are in Table 9. Best in **bold**.

Setting	Params	MSE↓ ($\times 10^{-3}$)
No memory, local qubits	650	0.53±0.06
Memory	642	0.52±0.08
$\mathcal{L}_{\text{out}}$	642	0.54±0.06
$\mathcal{L}_{\text{align}}$ (Ours)	642	**0.48±0.08**

Table 3 shows the same effect on every sample of the base benchmark. Spectral alignment improves MSE, PSNR, and log spectral distance on Cello, and MSE, PSNR, SSIM, and LPIPS on the images. The consistency across sound and images is more important than the absolute size of any single gain. Additional analysis in Appendix 14.

Table 3: SQMM without and with $\mathcal{L}_{\text{align}}$ on the base benchmark, mean $\pm$ SD. Cello reports LSD, the images SSIM and LPIPS. Best in **bold**.

	Cello		Astronaut		Camera		Coffee	
Metric	No loss	$\mathcal{L}_{\text{align}}$	No loss	$\mathcal{L}_{\text{align}}$	No loss	$\mathcal{L}_{\text{align}}$	No loss	$\mathcal{L}_{\text{align}}$
MSE↓ ($\times 10^{-3}$)	0.52±0.08	**0.48±0.08**	1.23±0.21	**1.12±0.10**	0.42±0.06	**0.38±0.08**	0.88±0.11	**0.78±0.05**
PSNR↑ (dB)	32.85±0.67	**33.27±0.72**	35.34±0.71	**35.66±0.39**	39.83±0.66	**40.28±0.94**	36.60±0.54	**37.12±0.28**
LSD↓ (1D)	4.70±0.25	**4.45±0.14**						
SSIM↑ ($\times 10^{-2}$) (2D)			99.23±0.10	**99.32±0.07**	98.21±0.26	**98.45±0.32**	98.93±0.10	**99.06±0.04**
LPIPS↓ ($\times 10^{-4}$) (2D)			3.54±0.84	**2.64±0.66**	3.14±1.26	**2.82±0.90**	3.55±0.99	**3.40±0.92**

Table 4: Audio MSE↓ ($\times 10^{-3}$), mean $\pm$ SD. * marks our model. All models are in Table 16. Best in **bold**.

	SQMM*	QIREN	FINER	SIREN	NeRF	MSA-INR	IGA	CAFE (enc.)
Params	642	649	691	673	698	673	673	703
GTZAN	**4.71±0.17**	44.19±1.88	5.14±1.06	8.65±0.61	8.89±0.44	42.11±3.39	8.45±0.56	6.42±1.34
US8K	**6.11±0.27**	51.83±0.86	8.56±2.35	15.55±1.49	13.96±0.72	52.21±1.55	15.47±1.22	10.83±3.75
VCTK	**1.00±0.03**	10.70±0.10	1.66±0.21	2.98±0.15	2.59±0.15	8.51±0.90	2.69±0.13	2.25±0.34

5.3 Audio

Audio reconstruction evaluates whether the single signal improvement transfers to varied waveform structure. GTZAN provides music, UrbanSound8K (US8K) provides environmental sounds, and VCTK provides speech. Clips are converted to mono. GTZAN uses 22.05 kHz audio, UrbanSound8K is resampled to 22.05 kHz, and VCTK is resampled to 24 kHz. A stride of 5 keeps 1024 coordinates beginning at 0.5 seconds, using the same coordinate normalization as the cello task.

Table 4 shows that SQMM remains stable across the three audio domains. The strongest competing baseline is FINER, which is consistent with the importance of adaptive spectral structure in audio INRs. SQMM improves over FINER on all datasets with fewer parameters.

5.4 Images

For image reconstruction, the preprocessing follows Section 5.1. We evaluate grayscale CIFAR10 using 100 images. This setting tests whether the SQMM gain extends beyond the three scikit image examples to a larger image collection. To test larger images, SQMM is fitted to the Kodak and DIV2K images from 32^2 to 256^2 pixels and at full size (Appendix 14.8).

Table 5 shows that SQMM also improves reconstruction on grayscale CIFAR10, where QIREN+ adds one spectrum layer to QIREN. SQMM achieves the lowest MSE and LPIPS and the highest SSIM with the fewest parameters of the compared models. This suggests that the memory coupled representation is not limited to the scikit image setting and transfers to a larger image collection. On Kodak and DIV2K, SQMM obtains the lowest MSE of all models at every resolution, including NeRF which holds $\times 1.54$ the parameters of SQMM (Table 6), and Figure 3 shows example reconstructions.

6 Discussion

SQMM suggests that memory coupling is useful not only as an architectural addition, but as a mechanism for organizing spectral computation across quantum INR bands. In prior QINR models, Fourier features are typically produced by independent quantum feature blocks or readout circuits [85, 77]. In SQMM, the shared memory provides an intermediate quantum state through which later bands can condition their coefficients on spectra expressed by earlier bands. This connects the architectural coupling of Section 4.1 to the Fourier view of data reuploading circuits (Section 4.2 and Appendix 10) [60, 9, 83].

The alignment objective further indicates that the memory should be treated as a trainable spectral interface rather than an opaque hidden state. The ablations in Section 5.2 support this interpretation.

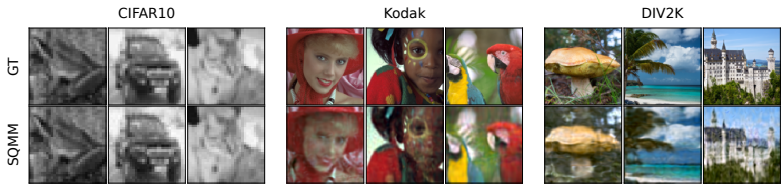

Figure 3: Ground truth (top) and SQMM (bottom), CIFAR10 at 32×32, Kodak and DIV2K at 128×128.

Table 5: Grayscale CIFAR10, mean $\pm$ SD. * marks our model. Table 21 has all models. Best in **bold**.

	SQMM*	QIREN+	QVF	FINER	SIREN	IGA	MSA-INR
Params	654	801	671	697	687	687	697
SSIM↑ ($\times 10^{-2}$)	**99.21±0.03**	97.69±0.19	92.90±0.23	97.43±0.39	98.37±0.32	98.56±0.27	98.59±0.14
LPIPS↓ ($\times 10^{-4}$)	**3.44±0.25**	30.41±2.98	104.84±8.38	14.17±3.69	8.22±1.89	7.64±1.70	10.56±1.93
MSE↓ ($\times 10^{-3}$)	**0.46±0.02**	1.67±0.13	5.12±0.17	1.45±0.21	0.98±0.21	0.98±0.18	0.90±0.11

Table 6: Kodak and DIV2K, MSE↓ ($\times 10^{-3}$), mean $\pm$ SD. Kodak without a size is the native 768×512. * marks our model. All models are in Tables 19 and 20. Best in **bold**.

	SQMM*	NeRF	MSA-INR	SIREN	IGA	R+RFF	FLAIR
Params	666	1023	747	717	717	723	723
Kodak 32^2	**1.58±0.10**	3.80±0.42	1.85±0.06	2.70±0.33	3.11±0.34	3.35±0.07	3.48±0.08
Kodak 64^2	**5.43±0.03**	8.76±0.68	6.84±0.06	6.89±0.26	7.06±0.25	9.47±0.18	9.57±0.46
Kodak 128^2	**11.54±0.05**	14.94±0.26	14.02±0.05	13.18±0.47	13.65±0.36	16.89±0.29	17.57±0.28
Kodak 256^2	**18.52±0.04**	21.57±0.23	21.54±0.04	20.57±0.55	20.94±0.54	24.53±0.34	25.37±0.14
Kodak	**32.08±0.15**	34.79±0.40	35.64±0.26	34.44±0.61	34.04±0.56	38.54±0.64	39.04±0.39
DIV2K 32^2	**3.13±0.08**	6.91±0.20	3.58±0.09	5.22±0.74	5.34±0.55	6.12±0.22	7.06±0.13
DIV2K 64^2	**11.13±0.06**	18.57±0.66	13.66±0.11	14.37±0.75	15.22±0.60	18.42±0.41	19.83±0.28
DIV2K 128^2	**23.16±0.34**	30.87±0.38	27.35±0.41	27.07±1.02	28.11±0.98	32.63±0.76	35.18±0.30
DIV2K 256^2	**35.15±0.43**	42.68±1.26	39.84±0.31	39.34±0.74	40.02±0.40	45.80±0.94	48.18±0.70
DIV2K 512^2	**54.13±0.44**	61.08±0.53	59.10±0.46	58.41±1.14	58.40±0.39	65.14±0.85	68.11±1.31

Memory access improves reconstruction, and memory alignment gives the strongest result among the compared variants in Table 9.

These results position spectral memory as a practical design principle for compact QINRs, which could extend beyond per instance reconstruction to coordinate based settings with central spectral structure, such as higher dimensional signals, time varying representations, and physical fields [46, 80, 28]. SQMM inherits the practical constraints of parameterized QML and recurrent quantum models, such as noise sensitivity, simulator cost, and hardware dependent scaling with circuit depth and memory size [43, 7, 65, 5, 12]. No experiment was run on quantum hardware, and only finite shot estimation and coherent gate perturbations are simulated. At inference the band observables need $\mathcal{O}(d_F T/\mathrm{e}^2)$ shots at standard error e, while full Pauli memory probing needs $\mathcal{O}(4^m T(n-1)/\mathrm{e}^2)$ shots per training update (Appendix 13).

7 Conclusion

SQMM introduces a memory coupled quantum INR where sequential data reuploading bands share a coherent quantum memory. The analysis shows that input independent memory read and write operations preserve the finite Fourier structure of data reuploading circuits while allowing later bands to condition on an operator valued spectral summary of earlier bands. SQMM achieves the best MSE among the evaluated compact INR and QINR baselines on the considered reconstruction benchmarks. Questions of quantum computational or hardware advantage remain outside the scope of this work. Overall, the results support coherent spectral memory as a useful mechanism for compact quantum implicit representations, with simulation cost and hardware measurement overhead as the main limitations.

Acknowledgments and Disclosure of Funding

This work was supported by the D.E. Koshland Jr. Family Career Development Chair in Advanced Technologies in Electrical and Computer Engineering. We declare no competing interests.

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[2] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021.

[3] Alice Barthe and Adrián Pérez-Salinas. Gradients and frequency profiles of quantum re-uploading models. *Quantum*, 8:1523, November 2024. doi: 10.22331/q-2024-11-14-1523. URL `https://doi.org/10.22331/q-2024-11-14-1523`.

[4] Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 685–694. PMLR, 2020.

[5] Johannes Bausch. Recurrent quantum neural networks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33*. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/hash/0ec96be397dd6d3cf2fecb4a2d627c1c-Abstract.html`.

[6] Nuri Benbarka, Timon Höfer, Andreas Zell, et al. Seeing implicit neural representations as fourier series. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2041–2050, 2022.

[7] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

[8] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, B AkashNarayanan, Ali Asadi, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[9] Berta Casas and Alba Cervera-Lierta. Multidimensional fourier series with quantum circuits. *Physical Review A*, 107(6):062612, 2023. doi: 10.1103/PhysRevA.107.062612.

[10] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021.

[11] Samuel Yen-Chi Chen, Tzu-Chieh Wei, Chao Zhang, Haiwang Yu, and Shinjae Yoo. Quantum convolutional neural networks for high energy physics data analysis. *Physical Review Research*, 4(1):013231, 2022.

[12] Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang. Quantum long short-term memory. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8622–8626. IEEE, 2022.

[13] Pierre-Luc Dallaire-Demers and Nathan Killoran. Quantum generative adversarial networks. *Physical Review A*, 98(1):012324, 2018.

[14] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021.

[15] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[16] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=OmtmcPkkhT`.

[17] Rich Franzen. Kodak lossless true color image suite. `http://r0k.us/graphics/kodak/`, 1999.

[18] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451. URL https://doi.org/10.1214/aos/1013203451.

[19] Keisuke Fujii and Kohei Nakajima. Harnessing disordered-ensemble quantum dynamics for machine learning. *Physical Review Applied*, 8(2):024030, 2017.

[20] Xun Gao, Eric R Anschuetz, Sheng-Tao Wang, J Ignacio Cirac, and Mikhail D Lukin. Enhancing generative models via quantum correlations. *Physical Review X*, 12(2):021037, 2022.

[21] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019.

[22] Francisco Javier Gil Vidal and Dirk Oliver Theis. Input redundancy for parameterized quantum circuits. *Frontiers in Physics*, 8:297, 2020. doi: 10.3389/fphy.2020.00297. URL https://doi.org/10.3389/fphy.2020.00297.

[23] Takahiro Goto, Quoc Hoan Tran, and Kohei Nakajima. Universal approximation property of quantum machine learning models in quantum-enhanced feature spaces. *Physical Review Letters*, 127(9):090506, 2021. doi: 10.1103/PhysRevLett.127.090506.

[24] Jufeng Han, Shu Wei, Min Wu, Lina Yu, Weijun Li, Linjun Sun, Hong Qin, and Yan Pang. Implicit neural representation with multi-scale sine activation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 21567–21575, 2026. doi: 10.1609/aaai.v40i26.39305.

[25] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567 (7747):209–212, 2019.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

[27] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16:1050–1057, 2020. doi: 10.1038/s41567-020-0932-7.

[28] Xiang Huang, Hongsheng Liu, Beiji Shi, Zidong Wang, Kang Yang, Yang Li, Min Wang, Haotian Chu, Jing Zhou, Fan Yu, et al. A universal pinns method for solving partial differential equations with a point source. In *IJCAI*, pages 3839–3846, 2022.

[29] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.

[30] Junbo Ke, Yangyang Xu, Chao Wang, and You-Wei Wen. Content-aware frequency encoding for implicit neural representations with Fourier-Chebyshev features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3646–3655, June 2026.

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

[32] Emanuel Knill, Gerardo Ortiz, and Rolando D. Somma. Optimal quantum measurements of expectation values of observables. *Physical Review A*, 75(1):012328, 2007. doi: 10.1103/PhysRevA.75.012328.

[33] Sukhun Ko, Seokhyun Youn, Dahyeon Kye, Kyle Min, Chanho Eom, and Jihyong Oh. FLAIR: Frequency- and locality-aware implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pages 4877–4887, June 2026.

[34] Chamin Hewa Koneputugodage, Yizhak Ben-Shabat, Sameera Ramasinghe, and Stephen Gould. VI^3NR: Variance informed initialization for implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13477–13486, June 2025.

[35] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[36] Anastasios Kyrillidis, Amir Kalev, Dohyung Park, Srinadh Bhojanapalli, and Constantine Caramanis. Provable compressed sensing quantum state tomography via non-convex methods. *npj Quantum Information*, 4: 36, 2018. doi: 10.1038/s41534-018-0080-4.

[37] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 562–570, 2015.

[38] Guangxi Li, Xuanqiang Zhao, and Xin Wang. Quantum self-attention neural networks for text classification. *arXiv preprint arXiv:2205.05625*, 2022.

[39] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16252–16262, 2022.

[40] Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. FINER: Flexible spectral-bias tuning in implicit NEural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2713–2722, 2024. doi: 10.1109/CVPR52733.2024.00262.

[41] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–arnold networks. In *International Conference on Learning Representations (ICLR)*, 2025.

[42] Owen Lockwood and Mei Si. Reinforcement learning with quantum variational circuit. In *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, volume 16, pages 245–251, 2020.

[43] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[44] Ali Mehrabian, Parsa Mojarad Adi, Moein Heidari, and Ilker Hacihaliloglu. Implicit neural representations with fourier kolmogorov-arnold networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025. arXiv:2409.09323.

[45] Hela Mhiri, Leo Monbroussou, Mario Herrero-Gonzalez, Slimane Thabet, Elham Kashefi, and Jonas Landman. Constrained and vanishing expressivity of quantum fourier models. *Quantum*, 9:1847, September 2025. doi: 10.22331/q-2025-09-03-1847. URL `https://doi.org/10.22331/q-2025-09-03-1847`.

[46] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020. doi: 10.1007/978-3-030-58452-8_24.

[47] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

[48] Tiago Novello, Diana Aldana, Andre Araujo, and Luiz Velho. Tuning the frequencies: Robust training for sinusoidal neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3071–3080, June 2025.

[49] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13503–13513, 2022.

[50] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Taming the waves: sine as activation function in deep neural networks. OpenReview, 2016. URL `https://openreview.net/forum?id=Sks3zF9eg`.

[51] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[53] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020. doi: 10.22331/q-2020-02-06-226.

[54] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[55] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, pages 142–158. Springer, 2022.

[56] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 1041–1044, 2014. doi: 10.1145/2647868.2655045. URL `https://doi.org/10.1145/2647868.2655045`.

[57] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk. WIRE: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18507–18516, 2023. doi: 10.1109/CVPR52729.2023.01775.

[58] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

[59] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.

[60] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

[61] Shayan Shekarforoush, David B. Lindell, David J. Fleet, and Marcus A. Brubaker. Residual multiplicative filter networks for multiscale reconstruction. In *Advances in Neural Information Processing Systems*, volume 35, pages 17713–17725, 2022.

[62] Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicit neural representation with fourier reparameterized training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[63] Kexuan Shi, Hai Chen, Leheng Zhang, and Shuhang Gu. Inductive gradient adjustment for spectral bias in implicit neural representations. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 54864–54891. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/shi25a.html`.

[64] S Shin, YS Teo, and H Jeong. Exponential data encoding for quantum supervised learning. *Physical Review A*, 107(1):012422, 2023.

[65] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[66] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[67] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473, 2020.

[68] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10753–10764, 2021.

[69] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8:131–162, 2007.

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

[71] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002. doi: 10.1109/TSA.2002.800560. URL `https://doi.org/10.1109/TSA.2002.800560`.

[72] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.

[74] Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit, version 0.92, 2019. URL `https://doi.org/10.7488/ds/2645`.

[75] Noah L Wach, Manuel S Rudolph, Fred Jendrzejewski, and Sebastian Schmitt. Data re-uploading with a single qudit. *arXiv preprint arXiv:2302.13932*, 2023.

[76] Hanrui Wang, Yongshan Ding, Jiaqi Gu, Yujun Lin, David Z Pan, Frederic T Chong, and Song Han. Quantumnas: Noise-adaptive search for robust quantum circuits. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 692–708. IEEE, 2022.

[77] Shuteng Wang, Christian Theobalt, and Vladislav Golyanik. Quantum visual fields with neural amplitude encoding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 158535–158559. Curran Associates, Inc., 2025. doi: 10.52202/085713-5297. URL `https://proceedings.neurips.cc/paper_files/paper/2025/file/e895155ee8e30f07f4e5e3d37475aa49-Paper-Conference.pdf`.

[78] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[79] Marco Wiedmann, Maniraman Periyasamy, and Daniel D. Scherer. Fourier analysis of variational quantum circuits for supervised learning. *arXiv preprint arXiv:2411.03450*, 2024. URL `https://arxiv.org/abs/2411.03450`.

[80] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Learning deep time-index models for time series forecasting. In *International Conference on Machine Learning*, pages 37217–37237. PMLR, 2023.

[81] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 41(2):641–676, 2022. doi: 10.1111/cgf.14505.

[82] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[83] Zhan Yu, Hongshun Yao, Mujin Li, and Xin Wang. Power and limitations of single-qubit native quantum neural networks. *Advances in Neural Information Processing Systems*, 35:27810–27823, 2022.

[84] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[85] Jiaming Zhao, Wenbo Qiao, Peng Zhang, and Hui Gao. Quantum implicit neural representations. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 60940–60956. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/zhao24l.html`.

[86] Ellen D Zhong, Tristan Bepler, Joseph H Davis, and Bonnie Berger. Reconstructing continuous distributions of 3d protein structure from cryo-em images. *arXiv preprint arXiv:1909.05215*, 2019.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract is expanded and explained in the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: limitations are addressed in the paper.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: theories are explained in the main body, proven in a high level, and proven fully in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Experiments are detailed.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Code and instructions are released at `https://github.com/Kiiwii10/SQMM`, and all datasets are public (Section 5).

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Experiment settings are detailed.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Where appropriate, error bars are included.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix 14.5 describes the compute nodes, and Section 5 and Appendix 14.5 give the experiment settings.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: It is complied.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work does not have direct social impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We do not see a possible misuse of our paper.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Assets used which are not ours are credited.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The released code is documented in its README.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs were used for writing and drafting.

Guidelines:

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

8 Notation

Table 7: Notation.

Symbol	Meaning
$x \in \mathbb{R}^d$, y, $\hat{y}(x)$	coordinate, target value, prediction
$\mathrm{PE}(x)$, $\gamma_i(x)$, $\gamma_M(x)$	positional encoding, upload variable of band i, one time memory upload
n, i, L_i	number of bands, band index, number of uploads in band i
D_i, q_i, M, m	local data register of band i and its qubit count, memory register and its qubit count
η_{D_i}, $\sigma_M^{(i)}(x)$	new data state of band i, memory state after band i ($\sigma_M^{(0)}$ is the initial memory)
$\mathcal{B}_i$, $\mathcal{S}_{i,\ell}$, $\mathcal{V}_{i,\ell}$	band channel, input dependent upload layer, input independent (trainable, read, write) block
$\xi_i(x)$	joint state of $D_i \otimes M$ after band i
$O_{i,a}$, $\boldsymbol{f}_i(x)$, $d_{f,i}$, d_F	band observables, band features, their number per band and in total
G_θ	classical decoder
$\Omega_{i,x}^{\mathrm{loc}}$, $\Omega_{M,x}^{(i)}$, $\Omega_{f_i,x}$	frequency support of band i, of the memory after band i, of the features of band i
$\tau_\mu^{(i)}$	matrix valued Fourier coefficient of $\sigma_M^{(i)}$ at frequency μ
$\mathcal{B}_{i,\nu}$, $\ell_{i,a,\nu}$	Fourier coefficient of the band channel, linear readout functional at frequency ν
$c_{\omega,a}^{(i)}$	Fourier coefficient of feature a of band i
$\Omega_{\mathrm{grid}} = \{\bar{\omega}_h\}_{h=1}^{N_\omega}$	finite frequency grid of the alignment loss
$\widehat{a}(u)$, Φ, ε	estimated Fourier coefficients of a sampled signal u, design matrix, ridge ($\varepsilon = 0$ in all experiments)
$\mathcal{Q}_m$, Q_s, $\chi_{i,s}^M$	m qubit Pauli strings, one string, memory Pauli signal $\mathrm{Tr}[Q_s \sigma_M^{(i)}]$
P_i^F, P_i^M	accumulated prefix feature spectrum, memory occupancy spectrum at boundary i
$\mathcal{N}_\delta$, δ	normalization to unit mass with smoothing δ
λ_{align}, λ_{out}	weights of $\mathcal{L}_{\mathrm{align}}$ and of the output spectral control $\mathcal{L}_{\mathrm{out}}$
sg	stop gradient

9 Quantum and Fourier Preliminaries

This appendix provides additional background to Section 3 and the definitions and proofs used by Section 4. The basic quantum notation is included to make the method readable to non QML reviewers.

Qubits, registers, and density matrices. A qubit is a two level quantum system with computational basis states $|0\rangle$ and $|1\rangle$. A pure one qubit state has the form $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with $|\alpha|^2 + |\beta|^2 = 1$. An n_q qubit register lives in $(\mathbb{C}^2)^{\otimes n_q}$, whose dimension is 2^{n_q}. When a subsystem may be entangled with another subsystem, it is described by a density matrix ρ, a positive semidefinite operator with $\mathrm{Tr}[\rho] = 1$. A register is a group of qubits. If ρ_{DM} is a joint state on a data register D with q qubits and memory register M with m qubits, then $\rho_{DM} \in \mathbb{C}^{2^{q+m} \times 2^{q+m}}$, and the reduced memory state is $\sigma_M = \mathrm{Tr}_D[\rho_{DM}] \in \mathbb{C}^{2^m \times 2^m}$.

Gates and channels. Quantum gates are unitary transformations. Typical single qubit rotation gates are

$$R_X(\theta) = e^{-\mathrm{i}\theta X/2}, \qquad R_Y(\theta) = e^{-\mathrm{i}\theta Y/2}, \qquad R_Z(\theta) = e^{-\mathrm{i}\theta Z/2}, \tag{21}$$

where $X, Y, Z \in \mathbb{C}^{2\times 2}$ are Pauli matrices. Two qubit gates, such as CNOT, CZ, or trainable Ising couplings, create correlations between qubits. For the proofs, it is convenient to treat a circuit block as a linear channel $\mathcal{C}$ acting on density matrices. Unitary gates are the special case $\mathcal{C}(\rho) = U\rho U^\dagger$.

Partial trace identities. For registers D and M, the partial trace Tr_D is the unique linear map satisfying

$$\mathrm{Tr}[A_M \ \mathrm{Tr}_D(X_{DM})] = \mathrm{Tr}[(I_D \otimes A_M)X_{DM}]$$

for every operator A_M on M. If K acts only on D, then

$$\mathrm{Tr}_D[(K \otimes I_M)X_{DM}] = \mathrm{Tr}_D[X_{DM}(K \otimes I_M)]\,.$$

This is the cyclicity of trace inside the subsystem being traced. It is used in Proposition 5.

Measurement and expectation values. For a Hermitian observable O and state ρ, the expectation value is $\langle O\rangle = \mathrm{Tr}[O\rho]$. SQMM uses expectation valued Pauli readouts as features. These features can be estimated either exactly in statevector simulation or by repeated circuit executions on hardware. They should not be interpreted as single shot selective measurement outcomes from the same coherent memory trajectory. The formal nonselective measurement statement is given in Proposition 5.

Kraus form of local measurements. A nonselective measurement on a data register D is represented by Kraus operators $\{K_\zeta\}_\zeta$ on D that satisfy

$$\sum_\zeta K_\zeta^\dagger K_\zeta = I_D.$$

The post measurement joint state averaged over outcomes is

$$\sum_\zeta (K_\zeta \otimes I_M)\rho_{DM}(K_\zeta^\dagger \otimes I_M).$$

When one conditions on a particular outcome, the corresponding branch must be divided by its probability. The distinction between averaged and conditioned states is the reason SQMM treats readouts as expectation valued probes rather than as reused single shot outcomes.

Operator spaces and channel coefficients. Let $\mathcal{L}(\mathcal{H})$ denote the complex vector space of linear operators on a finite dimensional Hilbert space $\mathcal{H}$. A quantum channel is a completely positive trace preserving linear map on $\mathcal{L}(\mathcal{H})$. Although a channel maps density matrices to density matrices, its linear extension acts on arbitrary operators in $\mathcal{L}(\mathcal{H})$. This convention is used throughout the proofs when Fourier coefficient maps such as $\mathcal{B}_{i,\nu}$ act on matrices $\eta_{D_i} \otimes \tau_\mu^{(i)}$. The coefficient maps in a Fourier expansion of a channel need not themselves be physical channels. They are linear superoperators whose phase weighted sum is the physical channel at each coordinate value.

Data reuploading and coordinate Fourier support. Data reuploading circuits repeatedly inject the uploaded variable $\gamma_i(x) \in \mathbb{R}^{g_i}$ into parameterized gates, interleaved with trainable input independent layers. For formulas in the upload domain, write $z \in \mathbb{R}^{g_i}$ for a generic uploaded variable and reserve $\gamma_i(x)$ for the coordinate dependent value. Each input dependent upload gate is generated by a finite dimensional Hermitian operator. A typical component gate has the form

$$U_{i,\ell,r}(z) = \exp\left(-\mathrm{i}z_r H_{i,\ell,r}\right), \qquad r = 1, \ldots, g_i, \tag{22}$$

where $H_{i,\ell,r}$ is the encoding generator for upload component r. Since $H_{i,\ell,r}$ has a finite spectrum, the matrix entries of this gate are finite sums of phases whose frequencies are the negated eigenvalues of $H_{i,\ell,r}$, so the induced channel and its expectation values have frequencies given by eigenvalue gaps. Across layers, accessible upload frequencies are finite sums of such gaps [60, 9, 83].

Linear algebra notation. For a complex matrix A, $A^\dagger$ denotes the conjugate transpose. The Frobenius norm is $\|A\|_F^2 = \mathrm{Tr}(A^\dagger A)$. The Hilbert Schmidt inner product is

$$\langle A, B\rangle_{\mathrm{HS}} = \mathrm{Tr}(A^\dagger B).$$

The ridge estimate in Eq. (51) is computed on the chosen finite frequency grid only. When $\varepsilon = 0$, exact coefficient recovery on that grid requires Φ to have full column rank. When $\varepsilon > 0$, the estimator is a stabilized projection and its coefficients are biased toward zero.

Fourier supports and conjugation. A finite operator valued Fourier expansion over coordinates has the form

$$F(x) = \sum_{\omega\in\Omega} e^{\mathrm{i}\langle\omega,x\rangle} F_\omega,$$

where $\Omega \subset \mathbb{R}^d$ is finite and each F_ω is independent of x. The set Ω may be a superset of the nonzero coefficient support. Consequently, all support statements in the proofs are upper bounds. If $F(x)$ has support Ω, then $F(x)^\dagger$ has support contained in $-\Omega$, with coefficient $F_\omega^\dagger$ at frequency $-\omega$. Thus products such as $U(x)\eta U(x)^\dagger$ remain finite Fourier whenever $U(x)$ is finite Fourier.

In the uploaded variable domain one may write

$$\mathcal{S}_{i,\ell}(z) = \sum_{\kappa\in\Omega_{i,\ell,\gamma}} e^{\mathrm{i}\langle\kappa,z\rangle}\mathcal{S}_{i,\ell,\kappa}, \qquad \Omega_{i,\ell,\gamma} \subset \mathbb{R}^{g_i}. \tag{23}$$

After substituting $z = \gamma_i(x) = A_i\mathrm{PE}(x) + b_i$, the layer is analyzed on the coordinate domain through

$$\mathcal{S}_{i,\ell}(\gamma_i(x)) = \sum_{\nu\in\Omega_{i,\ell,x}} e^{\mathrm{i}\langle\nu,x\rangle}\mathcal{S}_{i,\ell,\nu}. \tag{24}$$

For affine raw coordinate uploads this expansion is exact, with coordinate frequencies induced by $A_i^\top\kappa$. For fixed sinusoidal positional encodings, the analysis can instead be viewed in the lifted coordinate $\mathrm{PE}(x)$, or projected onto a finite coordinate grid. If an implementation places a trigonometric polynomial inside a rotation angle, the resulting coordinate expansion may be countable rather than finite. In that case the same support algebra applies to any finite truncation, and the alignment loss uses exactly such a finite working grid.

Pauli basis normalization. The m qubit Pauli strings form an orthogonal basis of $\mathcal{L}((\mathbb{C}^2)^{\otimes m})$ under the Hilbert Schmidt inner product. With the unnormalized convention used here,

$$\mathrm{Tr}(Q_s Q_{s'}) = 2^m \mathbf{1}\{s = s'\}.$$

Hence every operator A on m qubits admits the expansion

$$A = 2^{-m} \sum_{Q_s\in\mathcal{Q}_m} \mathrm{Tr}(Q_s A) Q_s.$$

Parseval identity for this orthogonal basis gives

$$\|A\|_F^2 = 2^{-m} \sum_{Q_s\in\mathcal{Q}_m} |\,\mathrm{Tr}(Q_s A)|^2.$$

10 Proofs for Spectral Memory Propagation

This section proves the statements summarized in Section 4.2. The proofs use the coordinate domain notation of the method. Thus $x \in \mathbb{R}^d$, $\gamma_i(x) \in \mathbb{R}^{g_i}$, coordinate frequencies lie in $\mathbb{R}^d$, and phases have the form $e^{\mathrm{i}\langle\omega,x\rangle}$. All local supports $\Omega_{i,x}^{\mathrm{loc}}$ are effective coordinate supports after composing the uploaded variable map $x \mapsto \gamma_i(x)$. Unbarred frequencies μ, ν, ω denote theoretical coordinate frequencies. Barred frequencies $\bar{\omega}_h$ are reserved for the finite grid used in Appendix 12. The symbol $\tau_\omega^{(i)}$ always denotes the matrix Fourier coefficient of the memory density matrix at frequency ω. Throughout, D_i denotes the data register of band i, M the shared memory register, I_M the identity on memory, η_{D_i} the fixed new data state, $\mathcal{S}_{i,\ell}(\gamma_i(x))$ the input dependent encoding layers, and $\mathcal{V}_{i,\ell}$ the input independent trainable layers.

Lemma 2 (closure under composition)**.** *Let $\mathcal{A}(x)$ and $\mathcal{C}(x)$ be channel families with finite coordinate domain Fourier expansions*

$$\mathcal{A}(x) = \sum_{\alpha \in \Omega_{\mathcal{A}}} e^{\mathrm{i}\langle \alpha, x\rangle} \mathcal{A}_\alpha, \qquad \mathcal{C}(x) = \sum_{\beta \in \Omega_{\mathcal{C}}} e^{\mathrm{i}\langle \beta, x\rangle} \mathcal{C}_\beta, \tag{25}$$

where $\Omega_{\mathcal{A}}, \Omega_{\mathcal{C}} \subset \mathbb{R}^d$ are finite and each coefficient map is independent of x. Then

$$(\mathcal{A} \circ \mathcal{C})(x) = \sum_{\omega \in \Omega_{\mathcal{A}} + \Omega_{\mathcal{C}}} e^{\mathrm{i}\langle \omega, x\rangle} \left(\sum_{\alpha + \beta = \omega} \mathcal{A}_\alpha \circ \mathcal{C}_\beta \right). \tag{26}$$

In particular, the composition is finite Fourier on the coordinate domain.

Proof. Expanding the composition gives

$$(\mathcal{A} \circ \mathcal{C})(x) = \sum_{\alpha \in \Omega_{\mathcal{A}}} \sum_{\beta \in \Omega_{\mathcal{C}}} e^{\mathrm{i}\langle \alpha + \beta, x\rangle} \mathcal{A}_\alpha \circ \mathcal{C}_\beta. \tag{27}$$

Grouping terms with the same total frequency $\omega = \alpha + \beta$ gives the stated expression. □

Proposition 1 (band closure under memory read and write insertion)**.** *Consider a bare data reuploading band*

$$\mathcal{B}_i(\gamma_i(x)) = \mathcal{W}_{i,L_i+1} \circ \mathcal{S}_{i,L_i}(\gamma_i(x)) \circ \mathcal{W}_{i,L_i} \circ \cdots \circ \mathcal{S}_{i,1}(\gamma_i(x)) \circ \mathcal{W}_{i,1}, \tag{28}$$

where each $\mathcal{S}_{i,\ell}(\gamma_i(x))$ is input dependent and each $\mathcal{W}_{i,\ell}$ is input independent. After inserting any input independent memory read and memory write layers, the resulting band can still be written as

$$\mathcal{B}_i(\gamma_i(x)) = \mathcal{V}_{i,L_i+1} \circ \mathcal{S}_{i,L_i}(\gamma_i(x)) \circ \mathcal{V}_{i,L_i} \circ \cdots \circ \mathcal{S}_{i,1}(\gamma_i(x)) \circ \mathcal{V}_{i,1}, \tag{29}$$

for suitable input independent blocks $\mathcal{V}_{i,\ell}$. Consequently, memory coupling preserves the reuploading form on the enlarged register $D_i \otimes M$.

Proof. Each inserted read or write layer is independent of x and $\gamma_i(x)$, so it can be absorbed into the nearest neighboring input independent block. For example, a read layer $\mathcal{R}$ placed after $\mathcal{S}_{i,\ell}(\gamma_i(x))$ can be absorbed by redefining $\mathcal{V}_{i,\ell+1} = \mathcal{W}_{i,\ell+1} \circ \mathcal{R}$. A write layer $\mathcal{W}$ placed after the last encoding layer is absorbed into the closing block $\mathcal{V}_{i,L_i+1}$. The same argument applies to memory only layers. The only input dependent objects remain the encoding channels $\mathcal{S}_{i,\ell}(\gamma_i(x))$, so the alternating reuploading structure is unchanged. □

Corollary 1 (finite Fourier stack)**.** *If every band $\mathcal{B}_j(\gamma_j(x))$ has a finite effective coordinate Fourier expansion with support $\Omega^{\mathrm{loc}}_{j,x}$, then every finite prefix channel through the first i bands has a finite effective coordinate Fourier expansion.*

Proof. The case $i = 1$ is immediate. The inductive step follows from Lemma 2 applied to the next band and the existing prefix channel. Input independent initialization, partial trace operations, and tensoring with a fixed new data state are linear maps independent of x, so they do not introduce new coordinate frequencies. An encoded initialization has a finite expansion by Proposition 2. □

Proposition 2 (memory initialization with optional initial encoding)**.** *Let the memory register have m qubits. If the initial memory state is $\eta_M = |0\rangle\langle 0|^{\otimes m}$ or $\eta_M = |+\rangle\langle +|^{\otimes m}$, and no input dependent gate is applied to memory before the first band, then*

$$\sigma^{(0)}_M(x) = \eta_M, \qquad \Omega^{(0)}_{M,x} = \{0\}. \tag{30}$$

If an input dependent memory encoding with the positional encoding of the bands is applied once at the beginning, define $\gamma_M(x) = A_M \mathrm{PE}(x) + b_M$, let $U_M(\gamma_M(x))$ be a finite product of memory upload gates, for example gates $R_Y((\gamma_M(x))_k)$ on memory qubits, and write $U_M(x) := U_M(\gamma_M(x))$. Then

$$\sigma^{(0)}_M(x) = U_M(x) \eta_M U_M(x)^\dagger = \sum_{\mu \in \Omega^{(0)}_{M,x}} e^{\mathrm{i}\langle \mu, x\rangle} \tau^{(0)}_\mu \tag{31}$$

for some finite support $\Omega^{(0)}_{M,x} \subset \mathbb{R}^d$ and matrices $\tau^{(0)}_\mu \in \mathbb{C}^{2^m \times 2^m}$, or approximately for a truncated support. The memory recursion and the downstream readout proof remain valid with this nontrivial initial support.

Proof. For $\eta_M = |0\rangle\langle 0|^{\otimes m}$ or $\eta_M = |+\rangle\langle +|^{\otimes m}$ without input dependent preprocessing, $\sigma_M^{(0)}(x)$ is constant in x. A constant function has the Fourier expansion $e^{\mathrm{i}\langle 0,x\rangle}\eta_M$, so its support is $\{0\}$.

Now suppose that the memory is passed through an input dependent unitary $U_M(\gamma_M(x))$ once at the beginning, with $\gamma_M(x) = A_M \mathrm{PE}(x) + b_M$. For a single gate $R_Y((\gamma_M(x))_k)$,

$$R_Y((\gamma_M(x))_k) = \cos\left(\frac{(\gamma_M(x))_k}{2}\right) I - \mathrm{i}\sin\left(\frac{(\gamma_M(x))_k}{2}\right) Y. \tag{32}$$

Under the same finite or truncated coordinate support assumption used for the data uploading gates, each matrix entry of $R_Y((\gamma_M(x))_k)$ has a finite or truncated Fourier expansion over x. Products of finitely many such memory encoding gates also have finite or truncated expansions by the argument of Lemma 2. The density matrix $U_M(x)\eta_M U_M(x)^\dagger$ is formed by products of entries from $U_M(x)$ and their complex conjugates. Therefore it has a finite or truncated operator valued Fourier expansion with some support $\Omega_{M,x}^{(0)}$.

The proof of Proposition 3 only assumes that the incoming memory has an operator valued Fourier expansion. It does not require the base support to be $\{0\}$. Therefore the same recursion applies when the memory is initialized by $|0\rangle$ or $|+\rangle$ and then receives an initial positional encoding driven by the same $\mathrm{PE}(x)$ used by the bands. The only change is that the global support starts from $\Omega_{M,x}^{(0)}$ rather than $\{0\}$, so the theory is unchanged after replacing the trivial base support by this encoded base support. □

Proposition 3 (operator valued Fourier recursion on memory)**.** *Assume the memory entering band $i+1$ has the form*

$$\sigma_M^{(i)}(x) = \sum_{\mu \in \Omega_{M,x}^{(i)}} e^{\mathrm{i}\langle \mu, x\rangle} \tau_\mu^{(i)}, \qquad \Omega_{M,x}^{(i)} \subset \mathbb{R}^d, \qquad \tau_\mu^{(i)} \in \mathbb{C}^{2^m \times 2^m}, \tag{33}$$

and that band $i+1$ has the effective coordinate domain channel expansion

$$\mathcal{B}_{i+1}(\gamma_{i+1}(x)) = \sum_{\nu \in \Omega_{i+1,x}^{\mathrm{loc}}} e^{\mathrm{i}\langle \nu, x\rangle} \mathcal{B}_{i+1,\nu}, \qquad \Omega_{i+1,x}^{\mathrm{loc}} \subset \mathbb{R}^d. \tag{34}$$

Then the outgoing memory state

$$\sigma_M^{(i+1)}(x) = \mathrm{Tr}_{D_{i+1}}\Big[\mathcal{B}_{i+1}(\gamma_{i+1}(x))\big(\eta_{D_{i+1}} \otimes \sigma_M^{(i)}(x)\big)\Big] \tag{35}$$

has the expansion

$$\sigma_M^{(i+1)}(x) = \sum_{\omega \in \Omega_{M,x}^{(i+1)}} e^{\mathrm{i}\langle \omega, x\rangle} \tau_\omega^{(i+1)}, \tag{36}$$

where

$$\tau_\omega^{(i+1)} = \sum_{\mu+\nu=\omega} \mathrm{Tr}_{D_{i+1}}\Big[\mathcal{B}_{i+1,\nu}\big(\eta_{D_{i+1}} \otimes \tau_\mu^{(i)}\big)\Big]. \tag{37}$$

Consequently,

$$\Omega_{M,x}^{(i+1)} \subseteq \Omega_{M,x}^{(i)} + \Omega_{i+1,x}^{\mathrm{loc}}. \tag{38}$$

Proof. Substituting the two Fourier expansions into the memory update gives

$$\sigma_M^{(i+1)}(x) = \mathrm{Tr}_{D_{i+1}}\left[\left(\sum_\nu e^{\mathrm{i}\langle \nu, x\rangle} \mathcal{B}_{i+1,\nu}\right)\left(\eta_{D_{i+1}} \otimes \sum_\mu e^{\mathrm{i}\langle \mu, x\rangle} \tau_\mu^{(i)}\right)\right] \tag{39}$$

$$= \sum_{\mu,\nu} e^{\mathrm{i}\langle \mu+\nu, x\rangle} \mathrm{Tr}_{D_{i+1}}\Big[\mathcal{B}_{i+1,\nu}\big(\eta_{D_{i+1}} \otimes \tau_\mu^{(i)}\big)\Big]. \tag{40}$$

Regrouping terms by $\omega = \mu + \nu$ gives Eq. (37) and the support bound. □

Proposition 4 (downstream readout coefficients)**.** *Under the assumptions of Proposition 3, let $O_{i+1,a}$ be a Hermitian observable used for feature component a of band $i+1$. The feature extraction map*

$$f_{i+1,a}(x) = \mathrm{Tr}\Big[(O_{i+1,a} \otimes I_M)\, \mathcal{B}_{i+1}(\gamma_{i+1}(x))\big(\eta_{D_{i+1}} \otimes \sigma_M^{(i)}(x)\big)\Big] \tag{41}$$

admits a finite or truncated Fourier expansion whose support is contained in $\Omega_{M,x}^{(i)} + \Omega_{i+1,x}^{\mathrm{loc}}$. Namely,

$$f_{i+1,a}(x) = \sum_{\omega \in \Omega_{M,x}^{(i)} + \Omega_{i+1,x}^{\mathrm{loc}}} c_{\omega,a}^{(i+1)} e^{\mathrm{i}\langle \omega, x\rangle}, \tag{42}$$

with coefficients

$$c_{\omega,a}^{(i+1)} = \sum_{\mu+\nu=\omega} \ell_{i+1,a,\nu}(\tau_\mu^{(i)}), \qquad \ell_{i+1,a,\nu}(X) = \mathrm{Tr}\big[(O_{i+1,a} \otimes I_M)\, \mathcal{B}_{i+1,\nu}\big(\eta_{D_{i+1}} \otimes X\big)\big]. \tag{43}$$

Here $\ell_{i+1,a,\nu}$ is the readout functional associated with observable $O_{i+1,a}$ and band frequency coefficient $\mathcal{B}_{i+1,\nu}$. It takes a memory side matrix X, appends the fixed data state, applies the channel coefficient, and returns a scalar expectation value.

Proof. The proof repeats the expansion in Proposition 3, but applies the feature extraction functional rather than tracing out the data register. The functional is linear because tensoring with a fixed operator, applying $\mathcal{B}_{i+1,\nu}$, and taking a trace are linear in X. Linearity gives Eq. (43). □

Corollary 2 (global support bound)**.** *If every band satisfies the assumptions of Proposition 3 and the memory is initialized independently of x, so $\Omega_{M,x}^{(0)} = \{0\}$, then*

$$\Omega_{M,x}^{(i)} \subseteq \sum_{j=1}^{i} \Omega_{j,x}^{\mathrm{loc}}, \qquad \Omega_{f_{i+1},x} \subseteq \sum_{j=1}^{i+1} \Omega_{j,x}^{\mathrm{loc}}. \tag{44}$$

If the memory receives an initial input dependent encoding as in Proposition 2, then

$$\Omega_{M,x}^{(i)} \subseteq \Omega_{M,x}^{(0)} + \sum_{j=1}^{i} \Omega_{j,x}^{\mathrm{loc}}, \qquad \Omega_{f_{i+1},x} \subseteq \Omega_{M,x}^{(0)} + \sum_{j=1}^{i+1} \Omega_{j,x}^{\mathrm{loc}}. \tag{45}$$

Proof. Apply Proposition 3 recursively. For input independent initialization, Proposition 2 gives $\Omega_{M,x}^{(0)} = \{0\}$. For an initial memory encoding driven by the same $\mathrm{PE}(x)$ used by the bands, Proposition 2 gives the finite or truncated base support $\Omega_{M,x}^{(0)}$. Applying Proposition 4 to the next readout gives the output support bound. □

11 Readout Probes and Memory Semantics

The main text separates coherent memory propagation from expectation valued readout. This section states the formal measurement fact behind that separation.

Proposition 5 (memory state after nonselective local readout)**.** *Let $\rho_{DM} \in \mathbb{C}^{2^{q+m} \times 2^{q+m}}$ be the joint post write state of a data register D with q qubits and memory register M with m qubits, and let $\sigma_M = \mathrm{Tr}_D[\rho_{DM}]$. Let a nonselective measurement act only on D, with Kraus operators $\{K_\zeta\}_\zeta$ satisfying $\sum_\zeta K_\zeta^\dagger K_\zeta = I_D$. Then the reduced memory state after averaging over outcomes is unchanged.*

$$\sum_\zeta \mathrm{Tr}_D\Big[(K_\zeta \otimes I_M)\rho_{DM}(K_\zeta^\dagger \otimes I_M)\Big] = \sigma_M. \tag{46}$$

Proof. Using cyclicity inside the traced subsystem,

$$\sum_\zeta \mathrm{Tr}_D\Big[(K_\zeta \otimes I_M)\rho_{DM}(K_\zeta^\dagger \otimes I_M)\Big] \tag{47}$$

$$= \sum_\zeta \mathrm{Tr}_D\Big[\rho_{DM}(K_\zeta^\dagger K_\zeta \otimes I_M)\Big] = \mathrm{Tr}_D\left[\rho_{DM}\left(\sum_\zeta K_\zeta^\dagger K_\zeta \otimes I_M\right)\right] = \mathrm{Tr}_D[\rho_{DM}]. \tag{48}$$

□

Selective outcomes. If one conditions on a particular outcome ζ, the memory state becomes

$$\sigma_{M|\zeta} = \frac{\mathrm{Tr}_D[(K_\zeta \otimes I_M)\rho_{DM}(K_\zeta^\dagger \otimes I_M)]}{p_\zeta}, \qquad p_\zeta = \mathrm{Tr}[(K_\zeta^\dagger K_\zeta \otimes I_M)\rho_{DM}], \tag{49}$$

which generally depends on ζ. Therefore, SQMM features are expectation values estimated from readout probes or separate repeated executions, not single shot selective outcomes whose post measurement memory state is reused downstream.

12 Details of Spectral Memory Alignment

This section expands the vector notation used in Section 4.3. The propagation lemma uses unbarred theoretical frequencies, while the loss is evaluated only on the finite grid $\Omega_{\mathrm{grid}} = \{\bar{\omega}_1, \ldots, \bar{\omega}_{N_\omega}\} \subset \mathbb{R}^d$.

Let $x_1, \ldots, x_T \in \mathbb{R}^d$ be the coordinate grid used only for the auxiliary spectral loss. For a sampled scalar signal $u = (u(x_1), \ldots, u(x_T))^\top \in \mathbb{C}^T$, define the Fourier design matrix $\Phi \in \mathbb{C}^{T \times N_\omega}$ by

$$\Phi_{th} = e^{\mathrm{i}\langle \bar{\omega}_h, x_t \rangle}. \tag{50}$$

Coefficients on Ω_{grid} are estimated by ridge stabilized least squares,

$$\widehat{a}(u) = (\Phi^\dagger \Phi + \varepsilon I_{N_\omega})^{-1} \Phi^\dagger u. \tag{51}$$

Here $I_{N_\omega} \in \mathbb{R}^{N_\omega \times N_\omega}$ is the identity matrix. In exact simulation, ε can be set to zero when Φ has full column rank. For finite shot estimates or ill conditioned grids, $\varepsilon > 0$ stabilizes the coefficient estimate.

For feature component a of band j, define the sampled signal

$$u_{j,a}^F = (f_{j,a}(x_1), \ldots, f_{j,a}(x_T))^\top, \qquad \widehat{c}_{j,a,h}^F = \big[\widehat{a}(u_{j,a}^F)\big]_h. \tag{52}$$

Collecting all feature components of band j at grid frequency $\bar{\omega}_h$ gives

$$\widehat{\boldsymbol{c}}_{j,h}^F = (\widehat{c}_{j,1,h}^F, \ldots, \widehat{c}_{j,d_{f,j},h}^F) \in \mathbb{C}^{d_{f,j}}. \tag{53}$$

The prefix feature occupancy used in the main text is therefore

$$P_i^F(\bar{\omega}_h) = \sum_{j=1}^{i} \|\widehat{\boldsymbol{c}}_{j,h}^F\|_2^2 = \sum_{j=1}^{i} \sum_{a=1}^{d_{f,j}} |\widehat{c}_{j,a,h}^F|^2. \tag{54}$$

This is an occupancy measure over extracted quantum features, not a spectrum of a linear layer or deployed decoder contribution. The occupancy target records which frequencies have been represented by earlier band features before the final decoder.

To estimate the memory spectrum, use the unnormalized Pauli string basis

$$\mathcal{Q}_m = \{I, X, Y, Z\}^{\otimes m}, \qquad |\mathcal{Q}_m| = 4^m, \qquad \mathrm{Tr}(Q_s Q_{s'}) = 2^m \mathbf{1}\{s = s'\}. \tag{55}$$

For each basis element, define the sampled memory signal

$$u_{i,s}^M = (\mathrm{Tr}[Q_s \sigma_M^{(i)}(x_1)], \ldots, \mathrm{Tr}[Q_s \sigma_M^{(i)}(x_T)])^\top, \qquad \widehat{c}_{i,s,h}^M = \big[\widehat{a}(u_{i,s}^M)\big]_h. \tag{56}$$

Collecting the Pauli probe coefficients at grid frequency $\bar{\omega}_h$ gives

$$\widehat{\boldsymbol{c}}_{i,h}^M = (\widehat{c}_{i,s,h}^M)_{Q_s \in \mathcal{Q}_m} \in \mathbb{C}^{4^m}. \tag{57}$$

The memory occupancy spectrum is

$$P_i^M(\bar{\omega}_h) = 2^{-m} \sum_{Q_s \in \mathcal{Q}_m} |\widehat{c}_{i,s,h}^M|^2. \tag{58}$$

If the memory state has the exact expansion

$$\sigma_M^{(i)}(x) = \sum_{\omega \in \Omega_{M,x}^{(i)}} e^{\mathrm{i}\langle \omega, x \rangle} \tau_\omega^{(i)}, \tag{59}$$

then, when every frequency of $\Omega_{M,x}^{(i)}$ is a discrete Fourier frequency of the sampling grid and $\varepsilon = 0$,

$$\widehat{c}_{i,s,h}^{M} = \mathrm{Tr}[Q_s \tau_{\bar{\omega}_h}^{(i)}]. \tag{60}$$

The Pauli strings form an orthogonal operator basis under the Hilbert Schmidt inner product. For any operator A on m qubits,

$$A = 2^{-m} \sum_{Q_s \in \mathcal{Q}_m} \mathrm{Tr}(Q_s A) Q_s, \tag{61}$$

and therefore

$$\|A\|_F^2 = \mathrm{Tr}(A^\dagger A) = 2^{-m} \sum_{Q_s \in \mathcal{Q}_m} |\,\mathrm{Tr}(Q_s A)|^2. \tag{62}$$

Applying this identity to $A = \tau_{\bar{\omega}_h}^{(i)}$ gives

$$P_i^M(\bar{\omega}_h) = \|\tau_{\bar{\omega}_h}^{(i)}\|_F^2 \tag{63}$$

up to finite grid estimation error. The identity Pauli string contributes only to the zero frequency trace component in the ideal normalized state setting.

Approximate and truncated spectra. The exact identity above requires an exact finite expansion on the coordinate domain and a frequency grid containing the frequency of interest. If the composed circuit has countable or continuous spectral content, or if relevant frequencies are absent from Ω_{grid}, the estimator computes the ridge projection of the sampled memory signals onto the chosen grid. In that case P_i^M is a proxy for the resolved memory spectrum rather than the full memory spectrum. The resulting limitations are standard finite grid spectral limitations, such as, off grid frequencies can leak into nearby grid coefficients, unresolved high frequencies are ignored, insufficient or nonuniform sampling can cause aliasing, and the ridge parameter can bias small coefficients. The propagation lemma still describes the algebra of the exact or truncated expansion being analyzed, but the alignment loss only constrains the portion visible through the selected probes, samples, and grid.

On hardware, estimating the full Pauli basis is expensive. Practical implementations may use a fixed subset of probes or randomized Pauli measurements, which gives a lower cost proxy for the same memory spectrum.

For any nonnegative spectrum $R : \Omega_{\mathrm{grid}} \to \mathbb{R}_{\geq 0}$, the normalized spectrum used by the alignment loss is

$$\mathcal{N}_\delta R(\bar{\omega}_h) = \frac{R(\bar{\omega}_h) + \delta}{\sum_{g=1}^{N_\omega} R(\bar{\omega}_g) + N_\omega \delta}. \tag{64}$$

The stop gradient in Eq. (19) is applied only to the normalized prefix feature occupancy, not to the memory occupancy.

13 Limitations

Spectral approximation and memory alignment. The spectral propagation result is exact when the composed memory state admits the finite coordinate Fourier expansion in Eq. (8). In implementations with sinusoidal positional encodings, nonlinear angle maps, finite sampling, or hardware estimates, this expansion is generally analyzed through a finite truncation or grid projection. The alignment objective in Section 4.3 inherits this approximation, since it matches the memory spectrum and prefix feature spectrum only on the chosen grid Ω_{grid} and only through the selected Pauli memory probes. Appendix 12 makes this proxy interpretation explicit. Frequencies outside the grid are not directly constrained. They can leak into nearby coefficients, alias under the sampled coordinate set, remain unresolved when they are too high frequency for the grid, or be suppressed by the ridge term in Eq. (51). Therefore $\mathcal{L}_{\mathrm{align}}$ should be interpreted as a practical mechanism for shaping the resolved memory spectrum.

Hardware execution and measurement cost. SQMM also inherits practical limitations common to parameterized QML and recurrent quantum models. Pauli readout features are expectation values, so hardware execution requires repeated circuit runs with fixed parameters. For a band feature $f_{i,a}(x)$, the circuit is executed up to band i, $O_{i,a}$ is measured on D_i, and outcomes are averaged across shots. The measured state is not reused for later bands. Downstream features use separate executions that preserve the coherent memory path without conditioning on single shot outcomes, as formalized in Appendix 11. This creates a measurement overhead. Estimating the d_F readout observables at one coordinate uses $\mathcal{O}(d_F/\mathrm{e}^2)$ shots for standard error e, so a pass over T coordinates uses $\mathcal{O}(d_F T/\mathrm{e}^2)$ shots [32]. Full spectral alignment is more expensive and uses $\mathcal{O}(4^m T(n-1)/\mathrm{e}^2)$ shots per training update for an m qubit memory, T sampled coordinates and $n-1$ boundaries [36], unless subset probes or shadow based estimators are used [27]. In addition, noise affects both the memory evolution and the readout estimates. Appendix 14.2 evaluates controlled circuit perturbations and observes smooth degradation, but this remains a simulator based study rather than a complete characterization of device noise, calibration error, decoherence, or measurement drift on real hardware.

14 Further Experiments and Ablations

In this section, we further explore SQMM. All SQMM results use the model of Section 5 at the epochs and growth intervals of the corresponding main table, unless stated otherwise.

14.1 Bands & Time

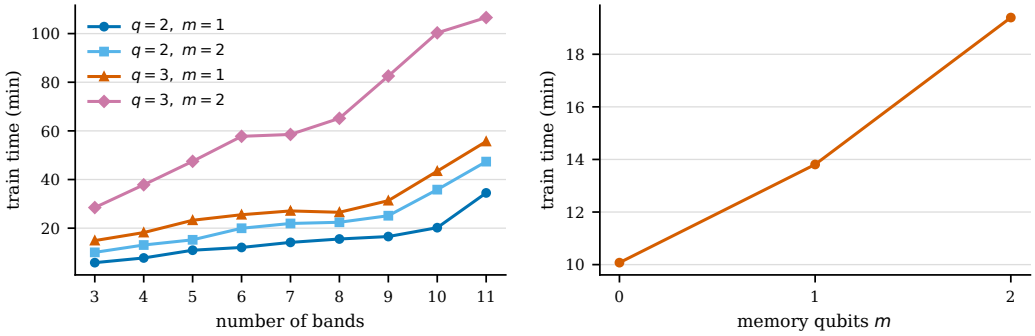

Figure 4: **Mixed state scaling ablation on Cello.** Training time against the number of sequential bands (left) and the number of memory qubits (right). Each point is one run with 600 training epochs and a fixed depth per band.

The setup for Figure 4 matches the cello data protocol in Section 5.1. The models were run without spectral loss, without classical preprocessing, and with linear postprocessing for 600 epochs. Runs used five data reuploads per band. The goal was to isolate how much the quantum component captures on its own and how circuit width and band count affect wall time and MSE. Adding bands and memory qubits lowered the reconstruction error in these runs, at the training time cost shown

in Figure 4. Results reaffirm the known simulator cost of added qubits, where each additional qubit doubles the state dimension and increases time and memory requirements as demonstrated in the right graph. Even a very small model with $q = 3$, $m = 1$, 11 bands, and 627 quantum parameters can take almost an hour to train on 1000 coordinates. Bands in the 5 to 7 range gave the best tradeoff among runtime, parameter count, and reconstruction quality, which guided the later model search toward architectures with fewer than 7 bands.

14.2 Noise & Hardware

We perturb every gate angle with Gaussian noise of standard deviation σ during training and evaluation (Table 8). The error grows smoothly with σ on the images and from $\sigma = 0.03$ on Cello, and the images are more sensitive to gate noise than Cello. Up to $\sigma = 0.03$ SQMM stays below every evaluated baseline on all four samples, and on Cello it does so at every σ. Figure 5 shows the reconstructions at every σ.

Table 8: Gaussian gate angle noise with SD σ in training and evaluation of the final model, MSE↓ ($\times 10^{-3}$), mean ± SD. $\sigma = 0$ is the noiseless model.

Perturbation	Cello	Astronaut	Camera	Coffee
$\sigma = 0$	0.48±0.08	1.12±0.10	0.38±0.08	0.78±0.05
$\sigma = 0.01$	0.46±0.11	1.46±0.11	0.44±0.05	0.97±0.32
$\sigma = 0.03$	0.56±0.06	1.62±0.14	0.68±0.05	1.10±0.16
$\sigma = 0.1$	0.73±0.14	2.48±0.56	1.03±0.15	1.80±0.36
$\sigma = 0.3$	1.13±0.19	4.89±0.58	2.38±0.76	3.90±1.00

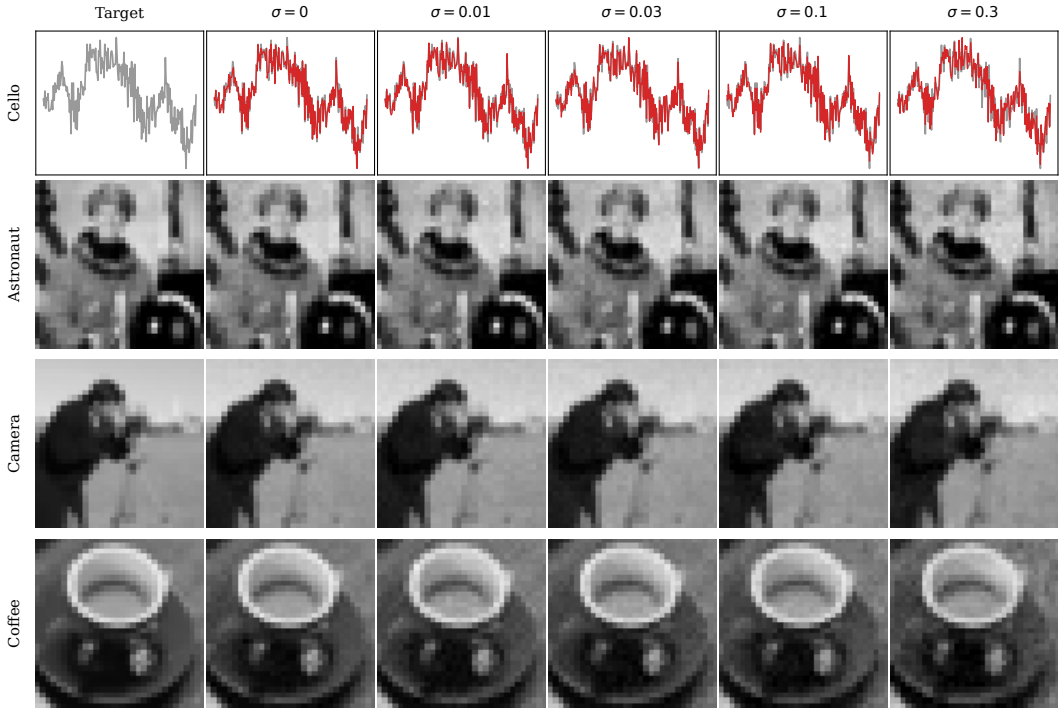

Figure 5: Reconstructions under gate angle noise of SD σ in training and evaluation. Cello shows samples 400 to 600 with the target in grey and the prediction in red.

14.3 Loss Extension

1D rFFT. For 1D uniformly sampled audio, Eq. (15) with $\varepsilon = 0$ and Ω_{grid} the nonnegative discrete Fourier frequencies equals the real FFT scaled by $1/T$, since the columns of Φ are orthogonal. The feature and probe traces are real valued, so the negative frequency coefficients are the conjugates

of the positive ones and carry no further information. The 1D real FFT is therefore an efficient implementation of the finite grid spectral projection on this grid.

Final output spectral control. The ablation in Table 9 also uses a control loss that applies the same spectral comparison directly to the final prediction. This loss is not the proposed memory alignment objective. It does not inspect $\sigma_M^{(i)}(x)$, does not use prefix occupancies, and does not provide an intermediate target for the memory lane. It only asks the final predicted signal to have the same normalized spectral energy distribution as the target signal on a finite frequency grid. For a generic nonnegative spectrum R on a finite grid Ω, write this normalization compactly as

$$\mathcal{N}_\delta R(\omega) = \frac{R(\omega) + \delta}{\sum_{\omega' \in \Omega} R(\omega') + |\Omega|\delta}. \tag{65}$$

Let $x_1, \ldots, x_T$ be the sampled coordinates used in a training batch, and let

$$\hat{y}_t = \hat{y}(x_t), \qquad y_t = y(x_t).$$

For each output channel $o \in \{1, \ldots, d_y\}$, estimate Fourier coefficients on a grid Ω_{out} using the same FFT, DFT, or least squares estimator as above. Write these coefficients as

$$\widehat{a}_{\hat{y},o}(\omega), \qquad \widehat{a}_{y,o}(\omega), \qquad \omega \in \Omega_{\mathrm{out}}.$$

The final output spectral energies are

$$P_{\hat{y}}^{\mathrm{out}}(\omega) = \sum_{o=1}^{d_y} \left|\widehat{a}_{\hat{y},o}(\omega)\right|^2, \qquad P_y^{\mathrm{out}}(\omega) = \sum_{o=1}^{d_y} \left|\widehat{a}_{y,o}(\omega)\right|^2. \tag{66}$$

After applying the same normalization operator as Eq. (18), the output spectral control loss is

$$\mathcal{L}_{\mathrm{out}} = \sum_{\omega \in \Omega_{\mathrm{out}}} \left(\mathcal{N}_\delta P_{\hat{y}}^{\mathrm{out}}(\omega) - \mathrm{sg}\big[\mathcal{N}_\delta P_y^{\mathrm{out}}(\omega)\big]\right)^2. \tag{67}$$

The corresponding control objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{out}} \mathcal{L}_{\mathrm{out}}. \tag{68}$$

The stop gradient is applied only to the target spectrum. Gradients flow through the final prediction $\hat{y}$, so this control regularizes the predictor output itself. In contrast, Eq. (19) shapes the propagated memory spectrum after each prefix and leaves the supervised task loss responsible for fitting the final signal values. In the experiments, $\mathcal{L}_{\mathrm{out}}$ uses $\lambda_{\mathrm{out}} = 0.1$.

No memory controls. The no memory control of Table 2 keeps the classical part of SQMM, including the projection that prepares the memory, and replaces the circuit by independent bands of four qubits. In every band the two data qubits are uploaded in every layer and read out through the same Pauli Z and ZZ features as in SQMM. The two further qubits are prepared once from the memory projection, take part in every entangling layer and are never measured. They start from $|0\rangle$ in every band, so nothing passes between bands. The bands use $[1, 2, 3, 3, 4]$ layers, which gives 650 parameters against 642 for SQMM, and the control is trained with the protocol of Table 1 without an auxiliary loss, since it has no shared memory to align. Table 9 repeats this control on all four samples as No memory, local qubits, with 662 parameters against 654 for SQMM on the images, and uses two further controls. No memory qubits removes the memory qubits and their couplings, leaving two qubits per band. No memory qubits with a wider decoder widens the decoder to 11 hidden units, which restores the parameter count but not the qubit count.

Mechanism ablation. Table 9 extends the mechanism ablation of Table 2 to all four samples. Removing the memory qubits gives the largest error on every sample, and a wider decoder at a similar parameter count recovers only part of the gap. The control with local qubits has a higher error than SQMM on Cello, Astronaut and Camera and the same error on Coffee. Read only memory lies between no memory and read and write memory. $\mathcal{L}_{\mathrm{out}}$ leaves the error of the memory model nearly unchanged, while $\mathcal{L}_{\mathrm{align}}$ gives the lowest error on every sample. The one memory qubit settings train SQMM with $\mathcal{L}_{\mathrm{align}}$ and a single memory qubit. The first keeps the bands of SQMM, and the second adds one reupload to band 1 and two to each of bands 2 to 5, which matches the parameter count of SQMM. With one memory qubit the error rises on every sample. At the parameter count of SQMM the added reuploads close part of the gap, most on Cello where the two errors lie within one SD, and SQMM keeps the lowest error on every sample. The second memory qubit is therefore worth more than the same number of parameters spent on reuploads.

Table 9: Mechanism and control ablation on the classic benchmark, MSE↓ ($\times 10^{-3}$), mean ± SD. Params give the live parameters of each setting. No memory qubits removes the memory qubits and their couplings, and No memory, local qubits is the control of Table 2. Both one qubit memory rows use $\mathcal{L}_{\text{align}}$, and the added reuploads match the parameter count of SQMM. Best in **bold**.

Setting	Params 1D	Params 2D	Cello	Astronaut	Camera	Coffee
No memory qubits	534	544	0.82±0.06	2.94±0.54	0.89±0.07	1.59±0.42
No memory qubits, wider decoder	636	646	0.65±0.08	1.94±0.18	0.73±0.20	1.33±0.21
No memory, local qubits	650	662	0.53±0.06	1.41±0.18	0.47±0.09	0.80±0.06
Read only memory	626	638	0.61±0.13	1.52±0.09	0.48±0.09	1.00±0.21
Read and write memory	642	654	0.52±0.08	1.23±0.21	0.42±0.06	0.88±0.11
Memory with $\mathcal{L}_{\text{out}}$	642	654	0.54±0.06	1.20±0.15	0.42±0.07	0.88±0.08
One qubit memory with $\mathcal{L}_{\text{align}}$	588	599	0.66±0.09	1.78±0.20	0.59±0.11	1.34±0.11
One qubit memory, more reuploads	642	653	0.52±0.05	1.38±0.17	0.49±0.08	1.21±0.21
Memory with $\mathcal{L}_{\text{align}}$ (Ours)	642	654	**0.48**±0.08	**1.12**±0.10	**0.38**±0.08	**0.78**±0.05

Memory switched off at evaluation. To test whether the trained model uses its memory, we take the five runs of Table 1 and evaluate each again with every memory read and write coupling set to zero. The read and write gates are controlled rotations and XY gates, which reduce to the identity at zero angle, so the bands run exactly as trained while no information passes through the memory register. Nothing is retrained. Switching the memory off raises the error by a factor of 36 on Cello, 388 on Astronaut, 291 on Camera and 149 on Coffee (Table 10), far above the error of a model trained without memory. The trained bands therefore rely on what the memory carries, and the memory register is not a component that the model learns to ignore.

Table 10: Memory switched off at evaluation on the classic benchmark, MSE↓ ($\times 10^{-3}$), mean ± SD over the five runs of Table 1. The first two rows evaluate the same trained models, with every read and write coupling set to zero in the second. The last row is the model without memory qubits of Table 9. * marks our model.

Setting	Cello	Astronaut	Camera	Coffee
SQMM*, memory on	0.48±0.08	1.12±0.10	0.38±0.08	0.78±0.05
Memory switched off at evaluation	17.05±4.65	442.40±194.67	110.84±37.98	113.69±50.65
Trained without memory qubits	0.82±0.06	2.94±0.54	0.89±0.07	1.59±0.42

Memory anatomy. Memory is an input dependent reduced quantum state rather than a feature vector, so we resolve it at every band boundary along three axes in Table 11. Spectral alignment compares the memory occupancy spectrum with the accumulated prefix spectrum by cosine similarity and Jensen Shannon divergence. Coordinate spread is the variance of $\langle P \rangle(x)$ across coordinates summed over the 15 non identity Paulis, which measures how far the state moves with the input. Write innovation is the norm of the Bloch vector that a write produces from a maximally mixed input, which isolates what the write injects from what memory retains. With $\mathcal{L}_{\text{align}}$, cosine agreement rises from the first to the last boundary while the coordinate spread decreases with depth. The write innovation is largest at the first boundary. Switching the memory off at evaluation raises the error of the trained Astronaut model by a factor of 388 (Table 10).

Table 11: Memory anatomy on Astronaut at each band boundary with $\mathcal{L}_{\text{align}}$.

Boundary	Cosine↑	JS↓	Coordinate spread	Write innovation
1	0.75	0.12	1.64	0.21
2	0.82	0.11	1.26	0.15
3	0.82	0.14	1.02	0.08
4	0.82	0.17	0.80	0.09

14.4 Finite shot measurement

Table 12 trains and evaluates SQMM with every expectation estimated from S shots, on all four samples at every S. On every sample the error decreases with S, and on Cello at 2048 shots it is 59.95 % above exact evaluation. The images need more shots than Cello to approach exact evaluation. On Cello SQMM is below every evaluated baseline from 32 shots on, while on the images its error at 2048 shots is still above that of the strongest baselines. Figure 6 shows how the reconstructions change with the shot count.

Table 12: MSE↓ ($\times 10^{-3}$) with expectations estimated from S shots in training and evaluation, mean $\pm$ SD.

Shots S	Cello	Astronaut	Camera	Coffee
16	3.65±0.23	18.39±1.76	7.64±0.78	10.04±0.45
32	3.05±0.12	12.24±0.82	5.89±0.84	8.15±1.01
64	2.30±0.17	9.50±0.70	4.07±0.29	5.82±0.43
128	1.81±0.14	7.86±0.43	3.38±0.26	4.61±0.57
256	1.39±0.08	6.17±0.57	2.63±0.44	3.76±1.14
512	1.12±0.07	4.70±0.60	2.21±0.38	3.00±0.35
1024	0.96±0.09	3.73±0.64	1.70±0.19	2.22±0.20
2048	0.76±0.06	2.86±0.53	1.27±0.16	1.79±0.16
exact	0.48±0.08	1.12±0.10	0.38±0.08	0.78±0.05

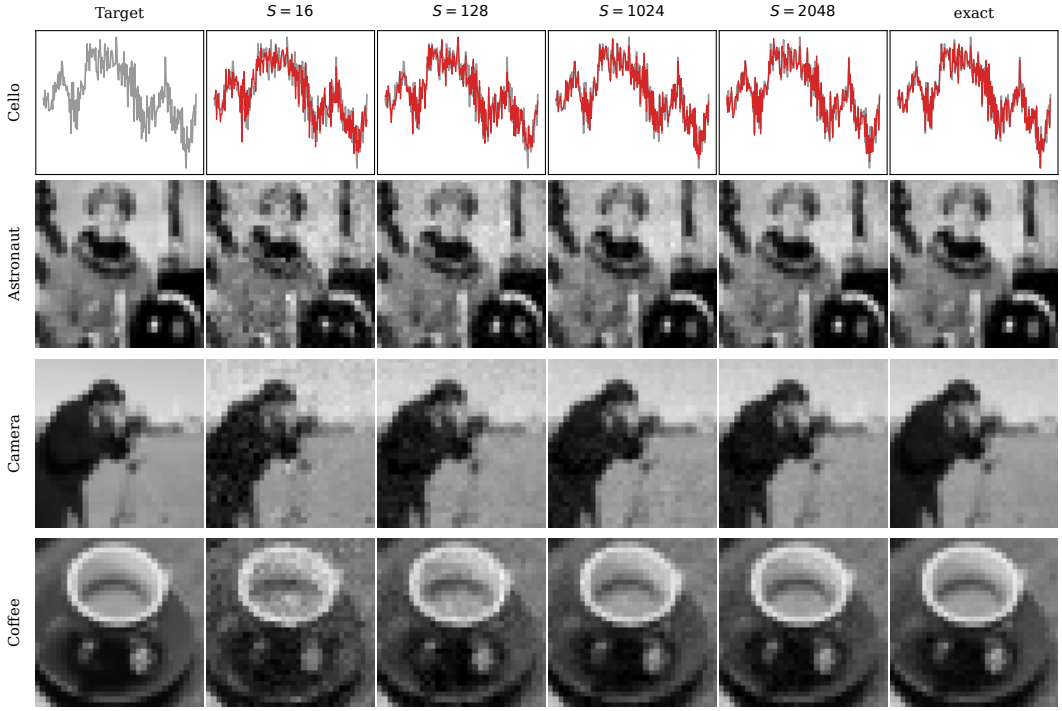

Figure 6: Reconstructions with expectations estimated from S shots in training and evaluation. Cello shows samples 400 to 600 with the target in grey and the prediction in red.

14.5 Model configurations

All experiments run on CPU nodes of a shared internal cluster, and the CPU type and memory of a node vary with the availability of the shared system. Table 13 lists the settings of SQMM and Table 14 the configurations of the classical baselines. The same SQMM is used for every dataset, with the coordinate encoding and spectrum bins of the input dimension and three outputs for RGB images.

Table 13: Settings of SQMM.

Component	Setting
Bands	5 bands of 2 data qubits with 2, 3, 4, 4 and 5 uploads
Encoding	the raw coordinate and 16 dyadic sinusoidal frequencies uploaded through R_Y in 1D, the raw coordinates and 8 consecutive frequencies per axis uploaded through R_Y and R_X with a shared bias in 2D
Memory	2 qubits coupled to every data qubit, no entangler of its own, initialized by R_Y and R_X uploads of the encoding
Read	controlled R_Y before the first upload of every band, an extra controlled R_X in band 5
Write	XY couplings in bands 1 to 4, none in band 5
Readout	Pauli Z and ZZ expectations of every band, concatenated
Decoder	MLP with one hidden layer of width 5 and GELU, three outputs for RGB images
Training	Adam, learning rate 0.03, full batch, one update per epoch
Band growth	one band per growth interval of 25 updates for 300 epochs and 50 updates for 600 epochs, with a fade in of 5 updates
$\mathcal{L}_{\text{align}}$	$\lambda_{\text{align}} = 3$, $\delta = 10^{-8}$, boundaries 1 to 3, switched off after four growth intervals
Bins	all $\lfloor T/2 \rfloor + 1$ nonnegative real FFT bins in 1D (501 on Cello, 513 on the audio clips), the 32×32 lowest signed frequencies of the grid FFT in 2D ($N_\omega = 1{,}024$ at every image size)

Spectral memory alignment. The feature spectrum uses the Pauli Z and ZZ features of every active band, and the memory spectrum uses the 4^m full Pauli signals $\chi^M_{i,s}(x)$ of the memory state after band i, which are available in simulation. All coordinates lie on regular grids, so Eq. (51) is used with $\varepsilon = 0$ and DFT frequencies, where it equals the discrete Fourier transform scaled by $1/T$. The implementation uses the orthonormal FFT, which multiplies both spectra by T. Energies are summed over the feature components or Pauli words of a band. The prefix target P^F_i is the cumulative sum of the band energies of bands 1 to i, normalized after summation, and P^M_i is normalized in the same way, so the factors 2^{-m} and T cancel up to a rescaling of δ to δ/T for the feature spectrum and to $\delta/(2^m T)$ for the memory spectrum. $\mathcal{L}_{\text{align}}$ is the squared difference of the normalized spectra averaged over the active boundaries and bins, without frequency weighting. Boundary i enters the loss when band $i+1$ is added, and the loss is switched off when the fifth band is added, so boundary 4, which is read only by the last band, is not supervised. The target is detached, and gradients of the memory side reach every parameter the memory state depends on.

Classical baselines. Every classical baseline has one configuration for audio (1D), fitted on Cello, GTZAN, UrbanSound8K and VCTK, and one for images (2D), fitted on Astronaut, Camera, Coffee, CIFAR10, Kodak and DIV2K with three outputs for RGB. A configuration fixes the hidden width, the number of hidden layers and the learning rate. The 1D shape is the shape selected for the audio comparison and the 2D shape the shape selected for the CIFAR10 comparison, both from every width and depth shape whose parameter count falls inside the budget. The learning rate is selected from $[3 \cdot 10^{-2}, 10^{-3}]$, by the lowest geometric mean of the best epoch MSE over Cello and the audio corpora in 1D and over Astronaut, Camera, Coffee and CIFAR10 in 2D. Kodak and DIV2K reuse the selected 2D configuration. All baselines are trained with Adam, full batch updates and no learning rate schedule on the MSE of targets in $[-1, 1]$, for the epochs of the corresponding SQMM experiment, and are evaluated with their best epoch weights.

Every other setting keeps the default of the official implementation. Two baselines change a default. MSA-INR uses a signed input domain and, in 2D, a frequency comb rescaled to the Nyquist band of the 32×32 grid. The published CAFE encoder of 88 random Fourier features and Chebyshev order 30 admits a single hidden unit within the budget, so we also report CAFE (enc.), which scales the encoder down to 5 features and order 2 in 1D and 21 features and order 7 in 2D.

Quantum baselines. QIREN uses its published hybrid architecture of width 8 with 2 hybrid layers and CZ entanglers, with spectrum layer 2 (649 parameters in 1D, 657 in 2D), and QIREN+ uses spectrum layer 3 (793 in 1D, 801 in 2D), and learning rate 0.03. QVF uses its published 671 parameter image model with 6 embedding frequencies, a hidden width of 10, one repetition and learning rate 0.01.

Table 14: Configurations of the classical baselines. Shape is hidden width $\times$ hidden layers. TUNER has no 1D configuration since its initialization requires 2D coordinates.

	1D (audio)		2D (images)	
Model	Shape	lr	Shape	lr
FINER	10×6	10^{-3}	24×1	$3 \cdot 10^{-3}$
SIREN	14×3	10^{-3}	14×3	$3 \cdot 10^{-3}$
NeRF	17×1	10^{-2}	12×3	$3 \cdot 10^{-2}$
KAN	2×4	$3 \cdot 10^{-2}$	2×4	10^{-2}
WIRE	25×2	$3 \cdot 10^{-3}$	25×2	$3 \cdot 10^{-3}$
R+RFF	10×5	$3 \cdot 10^{-3}$	24×1	$3 \cdot 10^{-2}$
ReLU	11×4	$3 \cdot 10^{-3}$	11×4	10^{-2}
Tanh	16×2	$3 \cdot 10^{-2}$	11×4	$3 \cdot 10^{-2}$
IGA	14×3	10^{-3}	14×3	$3 \cdot 10^{-3}$
FLAIR	10×4	$3 \cdot 10^{-3}$	17×2	$3 \cdot 10^{-2}$
TUNER			25×1	$3 \cdot 10^{-3}$
VI3NR	10×6	$3 \cdot 10^{-2}$	10×6	$3 \cdot 10^{-2}$
MSA-INR	24×1	$3 \cdot 10^{-2}$	24×1	$3 \cdot 10^{-2}$
CAFE (enc.)	13×1	$3 \cdot 10^{-2}$	4×1	$2 \cdot 10^{-2}$
CAFE	1×1	$2 \cdot 10^{-2}$	1×1	10^{-2}

14.6 Base benchmark best runs

Table 15 reports the base benchmark of Table 1 in the best run convention of QIREN, with all evaluated baselines and QIREN+, which adds one spectrum layer to QIREN at a larger parameter count. SQMM has the lowest best run error on every sample.

Table 15: Classic INR benchmark in the best run convention of QIREN, MSE$\downarrow$ ($\times 10^{-3}$) of the best single run. The SQMM rows take the best of the five runs in Table 1. QIREN+ is QIREN with one extra spectrum layer. * marks our model. Best in **bold**.

Model	Params 1D	Params 2D	Cello	Astronaut	Camera	Coffee
SQMM*	642	654	**0.38**	**1.01**	**0.28**	**0.73**
SQMM$^\star_{2048}$	642	654	0.71	2.36	1.09	1.57
QIREN	649	657	6.25	5.34	1.80	1.73
QVF		671		11.00	3.22	7.76
QIREN+	793	801	5.66	3.17	1.39	1.31
FINER	691	697	3.32	2.49	1.25	1.54
SIREN	673	687	4.93	1.82	0.44	1.96
NeRF	698	997	4.42	4.52	1.16	2.82
KAN	702	1222	3.07	43.40	21.02	21.10
WIRE	1294	1311	4.81	19.19	26.41	23.89
R+RFF	661	673	5.45	4.23	2.50	3.61
ReLU	672	683	7.65	11.36	3.15	4.21
Tanh	689	683	14.99	17.79	7.32	12.56
IGA	673	687	4.84	1.38	0.59	1.11
FLAIR	703	687	4.69	4.40	2.69	2.64
TUNER		701		4.73	5.38	3.78
VI3NR	691	701	7.78	6.24	3.22	3.76
MSA-INR	673	697	6.37	1.58	0.99	1.08
CAFE (enc.)	703	709	2.81	3.42	1.01	3.50
CAFE	625	715	11.52	30.97	36.94	19.73

14.7 Audio results

Table 16 compares all models on the three audio corpora. SQMM has the lowest error on every corpus. FINER and CAFE (enc.) are the closest baselines, while QIREN, QIREN+ and MSA-INR stay close to an order of magnitude above SQMM. Table 17 reports further SQMM metrics. Speech

(VCTK) is reconstructed with the lowest MSE and the highest PSNR, and music (GTZAN) has the highest log spectral distance.

Table 16: Audio reconstruction MSE↓ ($\times 10^{-3}$) for all models, mean ± SD. QIREN+ adds one spectrum layer to QIREN. ⋆ marks our model. Best in **bold**.

Model	Params	GTZAN	US8K	VCTK
SQMM⋆	642	**4.71**±**0.17**	**6.11**±**0.27**	**1.00**±**0.03**
QIREN+	793	40.92±1.15	50.18±0.34	9.73±0.76
QIREN	649	44.19±1.88	51.83±0.86	10.70±0.10
FINER	691	5.14±1.06	8.56±2.35	1.66±0.21
SIREN	673	8.65±0.61	15.55±1.49	2.98±0.15
NeRF	698	8.89±0.44	13.96±0.72	2.59±0.15
KAN	702	17.26±0.37	18.48±0.38	3.75±0.12
WIRE	1294	25.64±12.78	34.37±7.31	5.51±2.73
R+RFF	661	37.53±7.30	48.50±4.40	7.47±1.54
ReLU	672	58.90±4.77	58.16±1.37	11.41±0.77
Tanh	689	92.35±0.35	70.14±0.73	19.29±0.39
IGA	673	8.45±0.56	15.47±1.22	2.69±0.13
FLAIR	703	25.03±3.93	39.47±3.75	4.87±0.54
VI3NR	691	50.98±3.19	56.52±1.69	11.14±0.42
MSA-INR	673	42.11±3.39	52.21±1.55	8.51±0.90
CAFE (enc.)	703	6.42±1.34	10.83±3.75	2.25±0.34
CAFE	625	32.75±34.02	36.96±28.28	28.75±39.99

Table 17: SQMM audio metrics per corpus, mean ± SD.

Corpus	MSE↓ ($\times 10^{-3}$)	PSNR↑ (dB)	LSD↓
GTZAN	4.71±0.17	24.55±0.17	7.10±0.10
US8K	6.11±0.27	22.92±0.22	5.38±0.08
VCTK	1.00±0.03	33.94±0.17	5.57±0.06

14.8 Kodak and DIV2K

Table 18 compares all models on the 24 Kodak images at native resolution, and Tables 19 and 20 give the resolution ladders. SQMM has the lowest error at every resolution on both datasets, with the fewest parameters. The error of SQMM and of most baselines grows with resolution, since the parameter budget is fixed while the number of pixels grows. MSA-INR is the closest baseline up to 64^2, and SIREN or IGA at larger sizes, and the relative gap between SQMM and the closest baseline narrows as the resolution grows. All 2D configurations are selected on 32×32 images, and WIRE and TUNER degrade sharply from 64^2 on. In Figure 7 the remaining SQMM error concentrates on edges and fine textures, while smooth regions are reproduced closely.

Table 18: Kodak at native resolution (768×512, 24 images) for all models, MSE ($\times 10^{-3}$), mean $\pm$ SD. * marks our model. Best in **bold**.

Model	Params	MSE↓ ($\times 10^{-3}$)
SQMM*	666	**32.08±0.15**
FINER	747	39.13±0.59
SIREN	717	34.44±0.61
NeRF	1023	34.79±0.40
KAN	1274	54.10±1.35
WIRE	1383	47.58±3.21
R+RFF	723	38.54±0.64
ReLU	707	41.51±1.37
Tanh	707	45.00±0.35
IGA	717	34.04±0.56
FLAIR	723	39.04±0.39
TUNER	753	118.47±0.99
VI3NR	723	41.73±1.00
MSA-INR	747	35.64±0.26
CAFE (enc.)	719	48.45±2.09
CAFE	719	63.64±0.51

Table 19: Kodak resolution ladder (24 images, RGB), MSE↓ ($\times 10^{-3}$), mean $\pm$ SD. * marks our model. Best in **bold**.

Model	Params	32^2	64^2	128^2	256^2	native 768×512
SQMM*	666	**1.58±0.10**	**5.43±0.03**	**11.54±0.05**	**18.52±0.04**	**32.08±0.15**
FINER	747	5.82±0.36	10.18±0.26	17.32±0.24	25.03±0.34	39.13±0.59
SIREN	717	2.70±0.33	6.89±0.26	13.18±0.47	20.57±0.55	34.44±0.61
NeRF	1023	3.80±0.42	8.76±0.68	14.94±0.26	21.57±0.23	34.79±0.40
KAN	1274	19.49±1.72	25.53±1.26	34.69±1.71	42.10±2.41	54.10±1.35
WIRE	1383	19.27±1.68	52.10±3.73	53.46±9.73	37.37±3.40	47.58±3.21
R+RFF	723	3.35±0.07	9.47±0.18	16.89±0.29	24.53±0.34	38.54±0.64
ReLU	707	5.46±0.75	12.11±0.95	19.81±0.93	27.44±1.07	41.51±1.37
Tanh	707	7.17±0.29	14.42±0.39	22.68±0.33	30.46±0.45	45.00±0.35
IGA	717	3.11±0.34	7.06±0.25	13.65±0.36	20.94±0.54	34.04±0.56
FLAIR	723	3.48±0.08	9.57±0.46	17.57±0.28	25.37±0.14	39.04±0.39
TUNER	753	11.31±0.24	62.25±0.58	93.88±0.41	104.37±0.26	118.47±0.99
VI3NR	723	5.91±1.01	12.63±1.51	20.59±1.94	28.11±1.87	41.73±1.00
MSA-INR	747	1.85±0.06	6.84±0.06	14.02±0.05	21.54±0.04	35.64±0.26
CAFE (enc.)	719	7.50±1.90	16.21±3.13	26.82±1.80	35.03±2.60	48.45±2.09
CAFE	719	26.90±7.16	41.96±12.87	44.68±1.65	52.47±0.42	63.64±0.51

Table 20: DIV2K resolution ladder (10 images, RGB), MSE↓ ($\times 10^{-3}$), mean ± SD. The largest size is the 512^2 prepared image. * marks our model. Best in **bold**.

Model	Params	32^2	64^2	128^2	256^2	full 512^2
SQMM*	666	**3.13**±**0.08**	**11.13**±**0.06**	**23.16**±**0.34**	**35.15**±**0.43**	**54.13**±**0.44**
FINER	747	8.59±0.56	19.16±0.35	32.75±0.69	45.71±0.65	65.18±0.65
SIREN	717	5.22±0.74	14.37±0.75	27.07±1.02	39.34±0.74	58.41±1.14
NeRF	1023	6.91±0.20	18.57±0.66	30.87±0.38	42.68±1.26	61.08±0.53
KAN	1274	28.79±1.74	43.67±2.16	60.42±2.71	74.62±0.54	93.99±2.11
WIRE	1383	27.27±2.02	81.35±4.27	90.97±8.55	66.31±3.86	80.39±2.91
R+RFF	723	6.12±0.22	18.42±0.41	32.63±0.76	45.80±0.94	65.14±0.85
ReLU	707	12.08±1.70	27.58±3.08	42.66±2.66	56.28±2.63	75.50±3.10
Tanh	707	16.12±0.70	32.78±1.19	49.41±0.73	62.30±0.85	82.72±1.97
IGA	717	5.34±0.55	15.22±0.60	28.11±0.98	40.02±0.40	58.40±0.39
FLAIR	723	7.06±0.13	19.83±0.28	35.18±0.30	48.18±0.70	68.11±1.31
TUNER	753	16.02±0.25	95.56±0.92	147.69±0.30	164.13±1.04	182.94±1.22
VI3NR	723	10.95±1.39	24.18±2.15	39.20±1.88	52.54±2.25	72.22±2.38
MSA-INR	747	3.58±0.09	13.66±0.11	27.35±0.41	39.84±0.31	59.10±0.46
CAFE (enc.)	719	11.43±0.73	33.81±4.74	53.77±1.85	67.65±3.37	87.98±3.18
CAFE	719	41.56±8.76	70.45±23.49	74.60±1.70	88.81±2.42	107.59±2.59

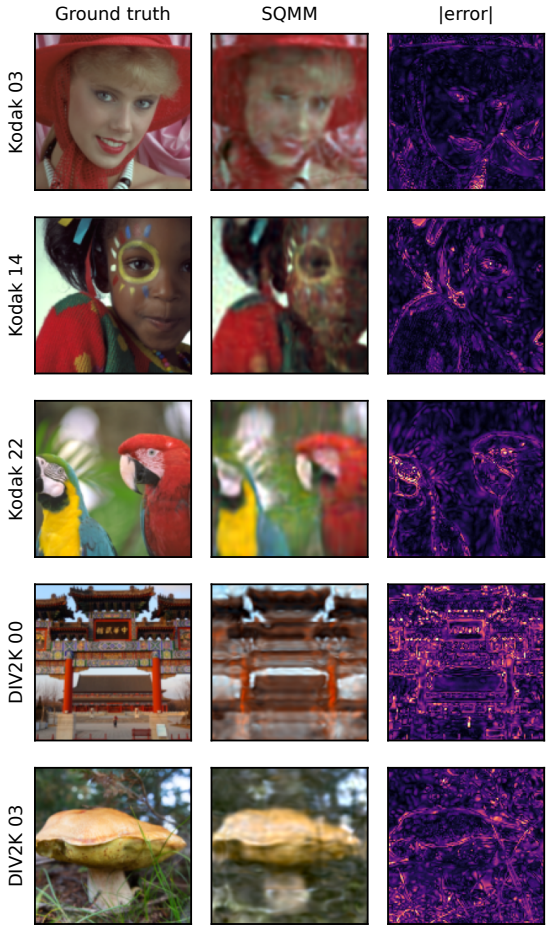

Figure 7: SQMM reconstructions of Kodak and DIV2K images at 128×128. Columns show the ground truth, the SQMM reconstruction, and the absolute error.

14.9 Ansatz Diagram

Figure 8 draws one band of the trained Cello model gate by gate, in the read, upload and write order of Section 4.

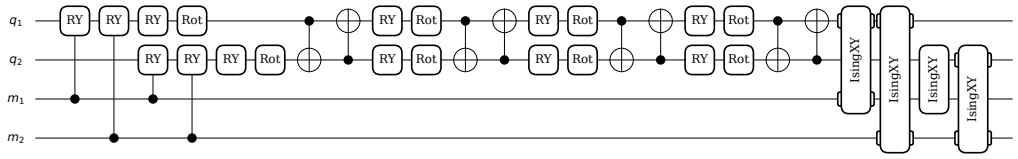

Figure 8: Band 3 of SQMM, recorded from the trained Cello model. Controlled R_Y reads from the memory qubits m_1, m_2 precede the four R_Y uploads on the data qubits q_1, q_2, each followed by an entangling layer, and XY couplings write to the memory at the end of the band.

14.10 CIFAR10 image reconstruction

Table 21 compares all models on the 100 grayscale CIFAR10 images. SQMM is best on all three metrics. MSA-INR, IGA and SIREN are the closest baselines, with about twice the MSE and more than twice the LPIPS of SQMM. Figure 9 shows the images with their SQMM reconstructions, which stay visually close to the ground truth at this resolution.

Table 21: Grayscale CIFAR10 metrics for all models, mean $\pm$ SD. MSE is on the $[-1, 1]$ range, SSIM and LPIPS on unit range images. QIREN+ adds one spectrum layer to QIREN. * marks our model. Best in **bold**.

Model	Params	SSIM↑ ($\times 10^{-2}$)	LPIPS↓ ($\times 10^{-4}$)	MSE↓ ($\times 10^{-3}$)
SQMM*	654	**99.21**$\pm$**0.03**	**3.44**$\pm$**0.25**	**0.46**$\pm$**0.02**
QIREN+	801	97.69$\pm$0.19	30.41$\pm$2.98	1.67$\pm$0.13
QVF	671	92.90$\pm$0.23	104.84$\pm$8.38	5.12$\pm$0.17
FINER	697	97.43$\pm$0.39	14.17$\pm$3.69	1.45$\pm$0.21
SIREN	687	98.37$\pm$0.32	8.22$\pm$1.89	0.98$\pm$0.21
WIRE	1311	83.21$\pm$2.51	306.79$\pm$62.91	18.23$\pm$2.68
R+RFF	673	96.14$\pm$0.74	93.60$\pm$47.00	2.78$\pm$0.62
ReLU	683	93.01$\pm$0.95	195.15$\pm$20.10	5.40$\pm$0.79
Tanh	683	88.06$\pm$0.88	478.55$\pm$67.66	8.85$\pm$0.62
IGA	687	98.56$\pm$0.27	7.64$\pm$1.70	0.98$\pm$0.18
FLAIR	687	96.24$\pm$0.46	86.76$\pm$28.50	2.81$\pm$0.38
TUNER	701	94.62$\pm$0.04	41.70$\pm$1.75	3.32$\pm$0.02
VI3NR	701	93.19$\pm$0.70	190.38$\pm$47.12	4.69$\pm$0.47
MSA-INR	697	98.59$\pm$0.14	10.56$\pm$1.93	0.90$\pm$0.11
CAFE (enc.)	709	93.81$\pm$1.79	57.85$\pm$26.60	4.59$\pm$1.87
CAFE	715	63.62$\pm$29.62	1430.30$\pm$2061.95	56.88$\pm$63.92

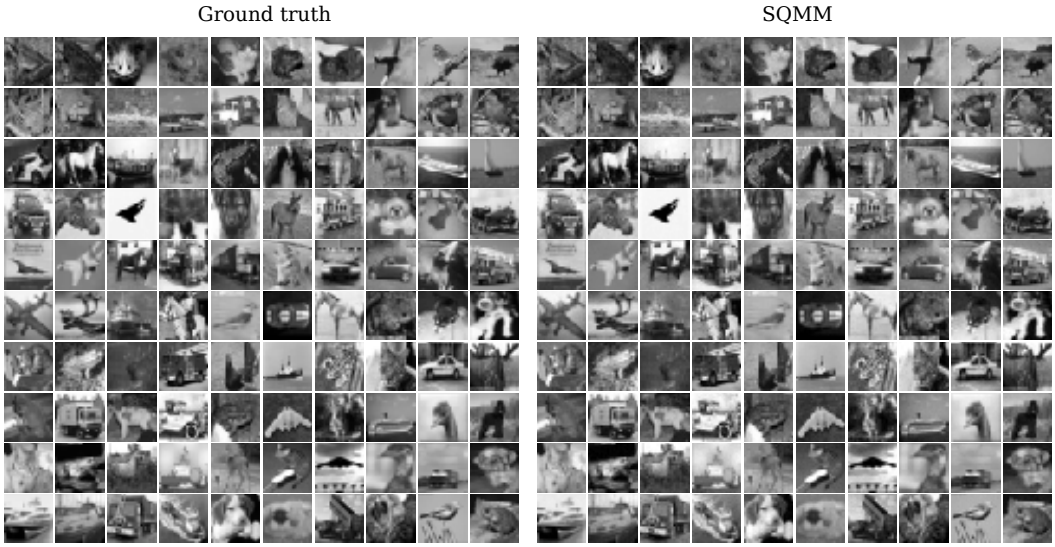

Figure 9: The 100 grayscale CIFAR10 images (left) and their SQMM reconstructions (right).