Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Fast Trainable Multilinear Bases for Image Compression

Shiwen An

an.s.0e57@m.isct.ac.jp

Institute of Science Tokyo

Zhongyi Ni

The Hong Kong University of Science and Technology (Guangzhou)

Huanhai Zhou

The Hong Kong University of Science and Technology (Guangzhou)

Jin-Guo Liu

The Hong Kong University of Science and Technology (Guangzhou)

Research Article

Keywords:

Posted Date: November 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11343443/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Fast Trainable Multilinear Bases for Image Compression

Shiwen An[1*], Zhongyi Ni[2], Huanhai Zhou[2], Jin-Guo Liu[2]

[1]Department of Information and Communication Engineering, Institute of Science Tokyo, Yokohama, 226-8501, Japan.
[2]Thrust of Advanced Materials, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, 511453, China.

*Corresponding author(s). E-mail(s): an.s.0e57@m.isct.ac.jp;

Abstract

The Discrete Fourier Transform (DFT), the Discrete Cosine Transform (DCT), and their block-wise variants underpin most deployed image and video codecs. Their effectiveness rests on three properties: their runtime is near-linear (up to a polylogarithmic factor) in the image size, they are exactly invertible, and they carry few to no parameters. We generalize these transforms to dataset-adaptive isometric multilinear bases that keep all three properties at the cost of a polylogarithmic number of parameters, by parameterizing each basis as an isometric tensor network, inspired by quantum many-body theory, and training one shared basis per image dataset with Riemannian optimization. Because the family contains the DFT and the DCT-IV (a variant of the DCT) as exact points, training can start from these transforms, and it improves on its starting point at the compression level it is trained for. We evaluate the bases on natural photographs and line drawings. On photographs, which the cosine basis already fits well, the best trained basis leads the DCT only under aggressive truncation. On Quick Draw line drawings, under a common codec at matched quality, the best trained basis stores each image in about 20% fewer bytes than the 8×8 block cosine transform of the Joint Photographic Experts Group (JPEG) format, not counting the one-off storage of the basis.

Keywords: Quantum Fourier transform, Tensor networks, Riemannian optimization, Image compression, Transform coding, Discrete cosine transform

1 Introduction

Transform coding sits at the core of most deployed image and video codecs, from the Joint Photographic Experts Group (JPEG) standard (Wallace 1992) to High Efficiency Video Coding (HEVC) (Sullivan et al. 2012). A fixed, fast transform maps image blocks or prediction residuals to coefficients, most of the signal energy lands in a few of them, and these coefficients are then quantized and entropy-coded (Goyal 2001; Mallat 2009). How well the transform concentrates energy sets how much the later stages can compress, so the transform is an important design choice in the pipeline. For decades this role has been filled by fixed trigonometric transforms. The most prominent are the Discrete Cosine Transform (DCT) (Ahmed et al. 1974) and its block-wise variants, together with the closely related Discrete Fourier Transform (DFT), whose fast algorithm, the Fast Fourier Transform (FFT) (Cooley and Tukey 1965), supplies the template for computing such transforms quickly.

No fixed basis, however, serves every image dataset equally well. A fixed basis has no parameters, so it cannot adapt to any of them. For

instance, the DCT comes close to the optimal Karhunen–Loève transform for sources near the first-order autoregressive (AR(1)) Gaussian regime (Jain 1979). Natural photographs sit near this regime but line drawings do not. To adapt the basis to the data, methods such as deep autoencoders (Theis et al. 2017; Ballé et al. 2017) use trainable nonlinear encoding and decoding maps. These learned transforms usually compress better, but cannot take over the role of the traditional transforms, because the maps themselves carry far more parameters than a single image. A useful *sparse basis*, one whose coefficients concentrate an image's energy in a few entries, should keep the properties that make the DCT and the FFT so effective:

1. **Negligible parameter overhead**: the basis is described by a number of parameters that is negligible compared with the image size.
2. **Fast to apply**: the transform and its inverse run in linear time, up to a factor polylogarithmic in the image size.
3. **Exactly invertible**: the inverse map is explicit, so the transform is lossless before truncation.

Under these constraints, we seek a systematic scheme that learns a basis for a given dataset, so that the dataset as a whole shrinks at the cost of storing only a negligible number of basis parameters.

Our insight comes from quantum computing, which supplies a precise dictionary between images and quantum states. An image with $N = 2^n$ pixels, flattened into a length-N vector, has exactly the shape of an n-qubit state vector. A quantum circuit composes unitary gates into a reversible linear map, so from the image's point of view a circuit is simply a change of basis. Simulating a circuit of $\mathrm{poly}(\log N)$ gates exactly takes $O(N\,\mathrm{poly}(\log N))$ time on a classical device, near-linear in the image size. The FFT itself is such a circuit, the quantum Fourier transform (QFT) (Coppersmith 1994), with its application cost derived in Appendix A.3. This dictionary has precedent in transform design: discrete wavelet transforms have been written as circuits of local unitary gates and new wavelets designed within that parameterization (Evenbly and White 2018), including wavelet families constructed with image compression in view (McCord and Evenbly 2022). A quantum circuit is also mathematically equivalent to a tensor network (Markov and Shi 2008), a multilinear map (Oseledets 2011) in which every tensor is constrained to preserve norms. For simplicity, we therefore parameterize the image basis as an isometric tensor network (Stoudenmire and Schwab 2016; Han et al. 2018) rather than as a quantum circuit (McClean et al. 2016; Cerezo et al. 2021a; Schuld and Killoran 2019; Benedetti et al. 2019).

What remains open is which network topology (the connectivity pattern of the tensors) best fits a given class of images, how to train it, and how the trained basis depends on the dataset. Since finding the best tensor decomposition is NP-hard even for a fixed structure (Hillar and Lim 2013), as is training the parameters of a fixed circuit (Bittel and Kliesch 2021), we settle the first two questions empirically.

We construct trainable isometric tensor-network bases from the FFT, DCT-IV, and quantum-inspired variants (Fig. 1) and train them with Riemannian optimization (Bécigneul and Ganea 2019; Hauru et al. 2021). For each topology, we learn *one shared basis per dataset* from its training images and reuse it unchanged for every held-out image. Training improves each basis over its initialization at the training keep ratio. On natural photographs (DIV2K; Agustsson and Timofte 2017), whose statistics the cosine basis already fits well, the best trained basis leads the DCT only under aggressive truncation. On non-photographic data such as Quick Draw line drawings (Google 2017; Ha and Eck 2018), the best trained basis stores each image in roughly 20% fewer bytes than the DCT at the same reconstruction quality under a common codec, not counting the one-off storage of the basis.

The rest of the paper proceeds as follows. Section 2 formulates the per-dataset sparse basis problem, Section 3 constructs the isometric tensor-network topologies, Section 4 describes the Riemannian optimization, and Section 5 reports the main results.

2 Problem formulation

We seek a family $\mathcal{F}$ of unitary image bases satisfying the transform-coding constraints of Section 1.

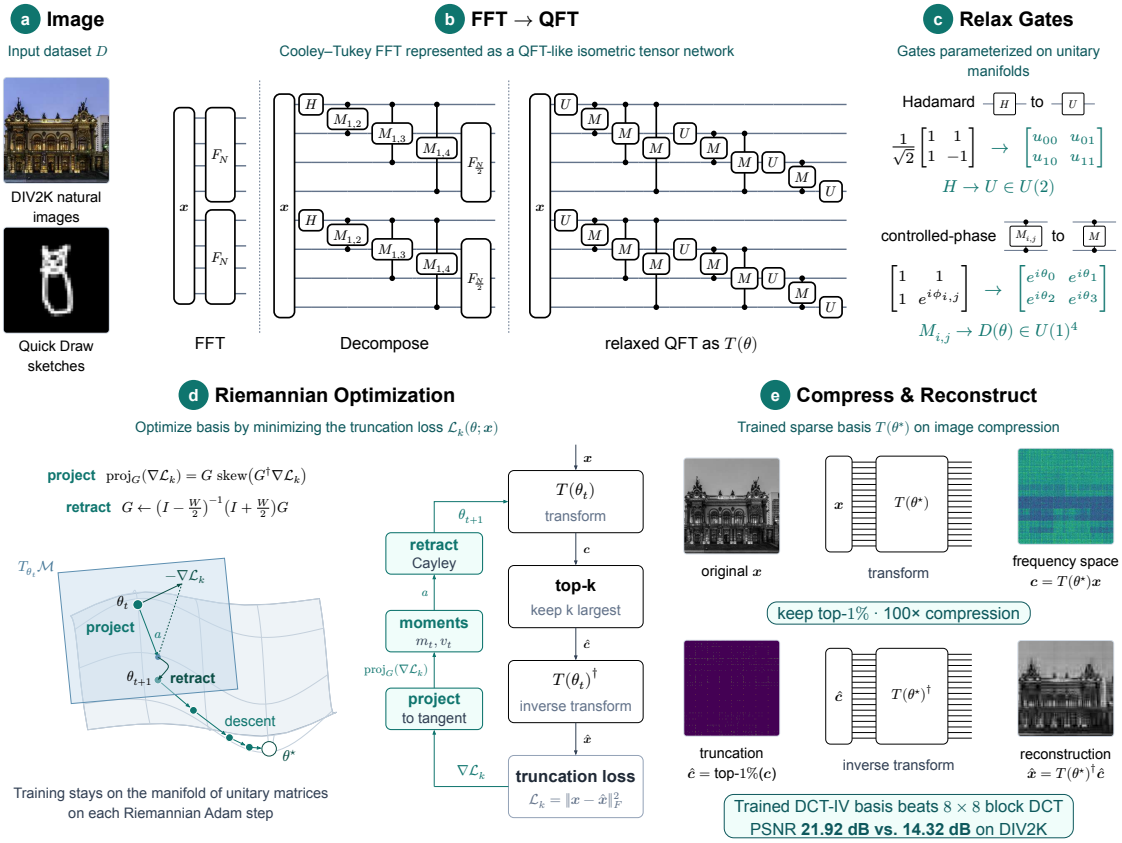

Fig. 1 Overview of the framework. (a) Training and evaluation datasets. The DIV2K example recurs in (e), and both examples are analyzed in Section 5.1. (b) The Cooley–Tukey FFT decomposed into controlled phases $M_{i,j}$ and rewritten as the relaxed QFT network. (c) Each gate family relaxed to a trainable unitary manifold. (d) Riemannian training on the truncation loss: project the gradient onto the tangent space $T_{\theta_t}\mathcal{M}$, update the Adam moments m_t, v_t, and retract with the Cayley map. (e) The trained basis applied at inference, here a DCT-IV keeping the top 1% of coefficients, with mean DIV2K test peak signal-to-noise ratios (PSNR). Example images from DIV2K (Agustsson and Timofte 2017) and Quick Draw (Google 2017; CC BY 4.0)

Each basis $T(\theta) \in \mathcal{F}$ is represented by an isometric tensor network whose gate parameters θ lie on the product manifold $\mathcal{M}$ of the gate factors (Section 3.2).

For a training sample $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L)}\}$ from a class of images and a retained-coefficient budget k, we learn one shared basis for the dataset by minimizing the mean post-truncation reconstruction error:

$$\mathcal{L}_k(\theta; \mathbf{x}) = \big\|\mathbf{x} - T(\theta)^\dagger\ \text{top-k}\big(T(\theta)\,\mathbf{x},\, k\big)\big\|_F^2,$$
$$\theta^\star = \arg\min_{\theta \in \mathcal{M}} \frac{1}{L}\sum_{\ell=1}^{L} \mathcal{L}_k\big(\theta; \mathbf{x}^{(\ell)}\big), \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm and top-k$(\cdot, k)$ keeps the k largest-magnitude coefficients and zeros the rest. After training, $T(\theta^\star)$ is stored once and applied to every held-out image from the dataset (Fig. 1e).

3 Isometric tensor-network bases

In order to find the best isometric tensor network $T(\theta)$'s structure, we realize a family $\mathcal{F}$ as tensor networks, whose small decomposed isometric tensors act directly on the binary digits of the image indices. An image $\mathbf{x} \in \mathbb{R}^{2^m \times 2^n}$ has one row and one column index. Writing both indices in binary recasts $\mathbf{x}$ as a tensor with $m + n$ binary legs, drawn as one open wire per qubit, . A

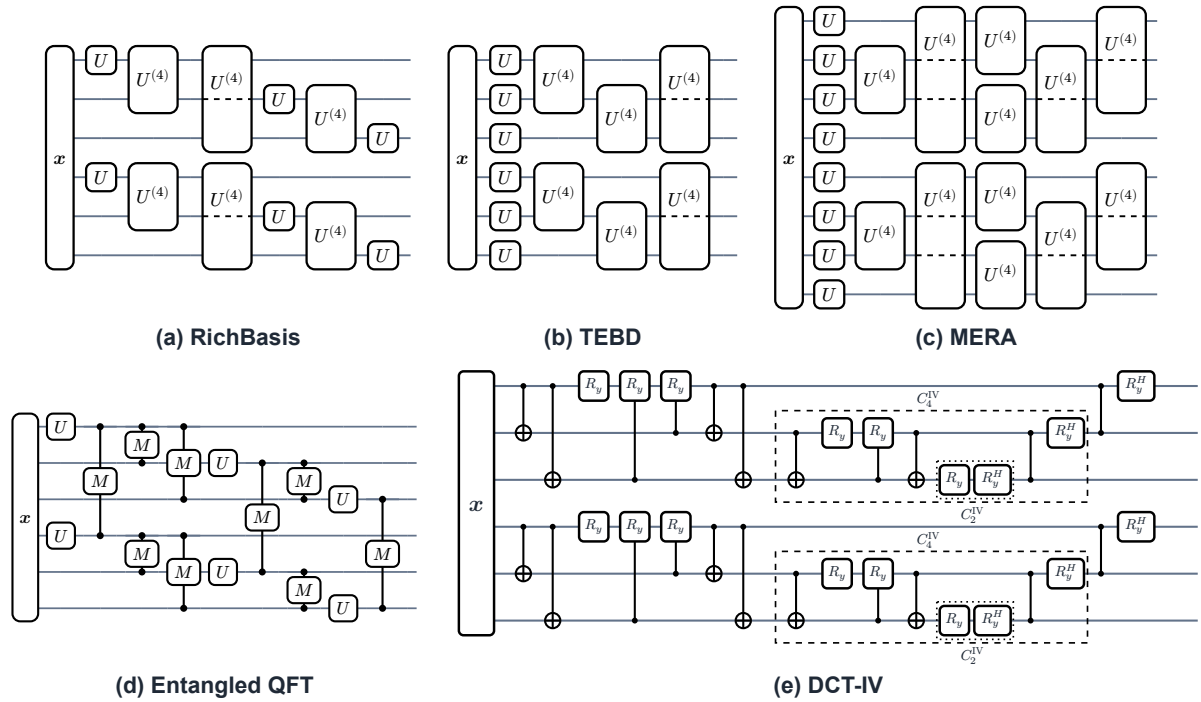

Fig. 2 Four circuit variants and the DCT-IV acting on input image $\mathbf{x}$, drawn at their relaxed gate families. (a)–(c) Topologies differing only in wiring, with relaxed Hadamard slots $U \in U(2)$ on one leg and general two-qubit gates $U^{(4)} \in U(4)$ on two (dashed wires pass through non-adjacent pairs). (d) Entanglement control based on existing QFT parametrization, within and across the two registers. Each M is a relaxed controlled phase $D(\boldsymbol{\theta}) \in U(1)^4$ on its two dotted wires. (e) The relaxed DCT-IV decomposition, with merge gates R_y^H relaxed on $O(2)$. Dots mark controls, $\oplus$ controlled-NOT targets, and dashed and dotted boxes the children C_4^{IV} and C_2^{IV}

basis $T(\theta)$ $\boxed{T(\theta)}$ is then an ordered contraction of small isometric tensors, called gates as in a quantum circuit, each acting on one or two legs. Each gate G is a 2×2 or 4×4 matrix on its legs, and being isometric it satisfies $G^\dagger G = I$, that is, it is unitary. Applying it amounts to attaching these gates to the image block $\boxed{\times}$ and absorbing them one at a time. The contraction order is fixed by the network's recursion rather than optimized; where the two-leg gates stay diagonal, fusing its levels gives the $O(2^{m+n}(m+n))$ fast-transform schedule noted in Appendix A.3. The construction of $\boxed{T(\theta)}$ begins with the classical FFT, which already has the network format.

3.1 Construction from fast transforms

For a $2^n \times 2^n$ image, the separable 2-D FFT maps $\mathbf{x} \mapsto F_N \mathbf{x} F_N^\top$, applying the per-axis DFT matrix F_N of size $N = 2^n$, computed by the Cooley–Tukey FFT, along each axis. The row and column indices of the image $\boxed{\times}$ thus form separate registers, each carrying a dense block F_N $\boxed{F_N}$ (Fig. 1b). One Cooley–Tukey step reduces F_N to $F_{N/2}$: a Hadamard H $\boxed{H}$ acts on the most-significant bit i, controlled phases $M_{i,j}$ $\boxed{M_{i,j}}$ couple it to each remaining bit j, and the half-size transform continues on the remaining legs, with

$$\boxed{H} = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$
$$\boxed{M_{i,j}} = M_{i,j} = \mathrm{diag}\big(1, 1, 1, e^{i\phi_{i,j}}\big),$$

and the twiddle phase $\phi_{i,j} = 2\pi/2^{j-i+1}$ fixed by the bit pair, as in the controlled R_{j-i+1} gate of the standard QFT circuit. The recursion terminates at F_2, which is H after the overall $N^{-1/2}$ normalization, and yields the QFT network, with one H per wire and one $M_{i,j}$ per pair of bits. Contracting it with the image $\boxed{\times}$ reproduces the classical FFT up to a fixed bit reversal, the overall $N^{-1/2}$ normalization, and the conjugate-root convention; Appendix A gives the full derivation.

We then make the tensor network T trainable by fixing its wiring and relaxing each gate within its natural matrix manifold (Edelman et al. 1998; Absil et al. 2008) (Fig. 1c); the last stage of Fig. 1b draws the resulting relaxed network. Each fixed Hadamard becomes an arbitrary unitary U, and each controlled phase keeps its diagonal form while freeing its diagonal entries into $D(\boldsymbol{\theta})$:

$$\boxed{H} \longrightarrow \boxed{U} = U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$$

$$\boxed{M_{ij}} \longrightarrow \boxed{M} = D(\boldsymbol{\theta}) = \operatorname{diag}\big(e^{i\theta_0}, e^{i\theta_1}, e^{i\theta_2}, e^{i\theta_3}\big),$$

with every $\theta \in \mathbb{R}/2\pi\mathbb{Z}$, a relaxed Hadamard ranges over $U(2)$ and a relaxed phase over $U(1)^4$; Appendix A.3 tabulates the same fixed-to-relaxed map for every gate in the family.

The QFT topology contains $m+n$ Hadamard gates and $P = \frac{m(m-1)}{2} + \frac{n(n-1)}{2}$ compact phase gates, so its implemented search space is the product manifold

$$\mathcal{M}_{\mathrm{QFT}}^{(m,n)} = U(2)^{m+n} \times \big(U(1)^4\big)^P, \qquad (2)$$

with one factor, $U(2)$ or $U(1)^4$, for each gate. Every gate is therefore unitary, and the network applies the gates in sequence, each on its own legs, so $T_{\mathrm{QFT}}(\theta) \in U(2^{m+n})$ at every parameter setting and its inverse is the adjoint network, applied gate by gate in reverse order. The same argument holds for every topology in the family, and Appendix B.2 shows that training preserves this unitarity. The parametric QFT contains the classical FFT at the Fourier parameter values and supplies the template for the rest of the family.

The DCT-IV isometric tensor network follows the same recurse–decompose–relax process as the QFT. Its recursion produces the sparse real circuit in Fig. 2e; relaxing those gates within their manifolds preserves unitarity, with the rotation and mirror gates staying real orthogonal, and retains the exact DCT-IV as its initialization. Appendix A.2 derives the recursion and bit reversal, with the gate-wise relaxation given in Appendix A.3.

3.2 Topology variants

The remaining topologies vary the wiring and the local gate (Fig. 2a–d), following circuit patterns from quantum many-body physics. *Entangled QFT* keeps the two QFT registers and adds one relaxed controlled phase $D(\boldsymbol{\theta})$ $\boxed{M}$ between each matched row–column pair. Setting the phases in these entangled pair gates to zero recovers the separable QFT. *TEBD* (time-evolving block decimation; Vidal 2004), *MERA* (multiscale entanglement renormalization ansatz; Vidal 2007), and *RichBasis* share a single gate set, the $U(2)$ single-qubit layer with unrestricted two-qubit gates in $U(4)$, and differ only in how those gates are wired. *TEBD* uses a nearest-neighbor ring, *MERA* a hierarchy of disentangler and isometry layers, every gate here a square two-qubit unitary, and *RichBasis* keeps the QFT's all-to-all wiring while relaxing each phase gate to a full two-qubit gate in $U(4)$. Fixing the shape of $T(\theta)$ fixes the product manifold $\mathcal{M}$ it optimizes at. Appendix A.3 lists each reference gate beside the family it relaxes to and defines these manifolds rigorously in Eq. (A11).

Since each gate lies in a unitary or real orthogonal factor of $\mathcal{M}$, every topology is unitary by the argument of Section 3.1, and its $O\big((m+n)^2\big)$ fixed-size gates (Table 1, counted in Appendix A.3) keep the parameter overhead polylogarithmic in the image size, as Section 1 requires.

4 Riemannian training

We use Riemannian optimization for the dataset-mean objective in Eq. (1), and report the budget as the keep ratio $\rho = k/2^{m+n}$, the retained fraction of all coefficients. Each training step evaluates $\mathcal{L}_k$ on a minibatch, averaged over its images, backpropagates it to a gradient on the gates (Appendix B.3), and then updates each gate on its prescribed manifold. We hold out 15% of the training slice for validation and return the parameters with the lowest validation loss.

4.1 Truncation loss

The loss has a coding interpretation that pins down what training can change. Write $\hat{\mathbf{x}} = T(\theta)^\dagger \,\text{top-k}\big(T(\theta)\,\mathbf{x},\, k\big)$ for the reconstruction drawn in Fig. 1d. Because $T(\theta)$ is unitary it preserves the Frobenius norm, so moving $T(\theta)$ inside the norm rewrites the pixel-domain error as the energy of the discarded coefficients, the sum of their squared magnitudes:

$$\mathcal{L}_k = \big\|\mathbf{x} - \hat{\mathbf{x}}\big\|_F^2 = \big\|T(\theta)\,\mathbf{x} - \text{top-k}\big(T(\theta)\,\mathbf{x},\, k\big)\big\|_F^2.$$

Table 1 Properties of each topology for a one-dimensional transform of length $N = 2^n$ on n qubits: the wiring (*Connectivity*), the numbers of *Gates* and of trainable real *Parameters*, the *Depth* in sequential gate applications, and the *Cost* of applying the transform. Ent. QFT, the Entangled QFT, counts its coupled register pair, and MERA requires a power-of-two n.

Property	**QFT**	**DCT-IV**	**Ent. QFT**	**TEBD**	**MERA**	**RichBasis**
Connectivity	all-to-all	all-to-all	all-to-all + row–column	ring	hierarchical	all-to-all
Gates	$\frac{1}{2}n(n+1)$	$\frac{3}{2}n(n-1)+3n-1$	$n(n+2)$	$2n$	$3n-2$	$\frac{1}{2}n(n+1)$
Parameters	$2n(n+1)$	$\frac{13}{2}n(n-1)+6n-4$	$4n(n+2)$	$20n$	$36n-32$	$4n(2n-1)$
Depth	$O(\log^2 N)$	$O(\log^2 N)$	$O(\log^2 N)$	$O(\log N)$	$O(\log N)$	$O(\log^2 N)$
Cost	$O(N \log N)$	$O(N \log^2 N)$	$O(N^2 \log N)$	$O(N \log N)$	$O(N \log N)$	$O(N \log^2 N)$

The same norm preservation fixes the total coefficient energy, $\|T(\theta)\,\mathbf{x}\|_F^2 = \|\mathbf{x}\|_F^2$, for every θ, so $\mathcal{L}_k$ is the part of the image's energy that the k retained coefficients fail to capture. For a fixed basis, keeping the k largest-magnitude coefficients already retains as much energy as any choice of k coefficients, so training can lower $\mathcal{L}_k$ only by changing the basis. It rotates the gates inside $T(\theta)$ so that the energy of the training images concentrates on fewer coefficients, leaving less of it outside the retained set. The truncation itself stays outside the transform.

The backward pass differentiates through top-k, which is not smooth. It still yields a well-defined gradient because the selection it makes is piecewise constant in θ: which k coefficients are retained changes only when a discarded coefficient overtakes a retained one in magnitude. Between such crossings, top-k applies one fixed selection, so the loss is simply the energy of a fixed set of discarded coefficients, a smooth function of θ, and automatic differentiation returns its exact gradient $\nabla\mathcal{L}_k$. A crossing is a parameter setting that a descent step essentially never lands on exactly, so the smoothed surrogates developed for discrete selections (Bengio et al. 2013; Jang et al. 2017; Xie and Ermon 2019) are not needed.

4.2 Manifold-constrained update

Every trainable gate G of $T(\theta)$ must keep $G^\dagger G = I$ (Section 3) and so stay on its manifold, but a step along the Euclidean gradient $\nabla\mathcal{L}_k$ with respect to G would break this constraint. The return path in Fig. 1d repairs this with two geometric operations, stated here for a unitary gate G.

Projection fixes the direction. A matrix W is skew-Hermitian if $W^\dagger = -W$. Differentiating $G^\dagger G = I$ along a curve of gates through G with velocity Δ gives $\Delta^\dagger G + G^\dagger \Delta = 0$, so $W = G^\dagger\Delta$ is skew-Hermitian. The directions that keep the constraint to first order, which form the tangent space at G, are therefore exactly $\Delta = G\,W$ with W skew-Hermitian (Edelman et al. 1998; Absil et al. 2008). Projection keeps the gradient's nearest tangent component in the Frobenius norm, $\mathrm{proj}_G(\nabla\mathcal{L}_k) = G\,\mathrm{skew}\big(G^\dagger\nabla\mathcal{L}_k\big)$, where $\mathrm{skew}(A) = \frac{1}{2}(A - A^\dagger)$ is the skew-Hermitian part of A, so stepping against it is steepest descent among the feasible moves.

Retraction fixes the step. The manifold is curved, so a finite step along a tangent direction still leaves it, and each update maps the stepped gate back. For an increment a built from the projected direction, the Cayley retraction forms $W = \mathrm{skew}\big(a\,G^\dagger\big)$ and applies $G_+ = \big(I - \frac{1}{2}W\big)^{-1}\big(I + \frac{1}{2}W\big)\,G$, which is exactly unitary for every skew-Hermitian W and matches $G + a$ to first order.

Riemannian Adam (Kingma and Ba 2015; Bécigneul and Ganea 2019) applies this sequence gate by gate, replacing the plain gradient step $a = -\eta\ \mathrm{proj}_G(\nabla\mathcal{L}_k)$ of step size η with an adaptive one built from moment estimates and transporting the first moment to the new tangent space by re-projection. All reported experiments use this optimizer, whose moment equations, orthogonal and phase-gate cases, and step-size schedule are given in Appendix B.

4.3 Initialization

Initialization is topology-specific: QFT and DCT-IV begin at their defining fixed-transform values, RichBasis at the same QFT point, Entangled QFT at the separable QFT with small seeded pair couplings, and TEBD and MERA at seeded feasible parameters. Because the objective is non-convex, these choices do not provide a global-optimality guarantee, and randomly initialized parameterized

Table 2 Mean test PSNR (dB) on 100 held-out images per dataset (image size in parentheses) at keep ratios ρ. *Ours* are the trained bases, *Classical* the fixed 8×8 block transforms. Per column, the highest mean is **bold** and the second highest underlined. MERA is undefined on Quick Draw's 5 qubits per axis (—). Per-image PSNRs, standard errors (0.1–1.0 dB per cell) and paired-bootstrap intervals are archived with the benchmarks (An et al. 2026a).

	DIV2K (256^2)					**Quick Draw** (32^2)				
$\rho =$	0.01	0.05	0.10	0.20	0.40	0.01	0.05	0.10	0.20	0.40
Ours										
QFT	21.00	24.68	27.14	30.84	37.54	12.87	16.43	19.22	23.82	33.51
Entangled QFT	21.00	24.68	27.14	30.84	37.54	12.87	16.43	19.22	23.82	33.51
RichBasis	14.54	25.67	28.84	33.31	40.93	13.42	18.57	22.73	29.59	44.11
TEBD	20.53	25.69	28.63	32.91	40.37	**13.43**	18.55	22.69	29.56	43.88
MERA	21.30	25.77	28.60	32.78	40.09	—	—	—	—	—
DCT-IV	**21.92**	**26.11**	28.82	32.77	39.60	13.08	**18.92**	**23.79**	**31.57**	**44.62**
Classical										
DFT (8×8)	14.31	24.09	26.70	30.37	36.72	12.42	15.82	18.55	22.81	32.25
DCT-II (8×8)	14.32	25.71	**28.98**	**33.59**	**41.43**	12.70	16.89	20.36	26.12	39.00

circuits are known to face flat optimization landscapes (McClean et al. 2018; Cerezo et al. 2021b). The QFT gate-unfreezing random-seed tests and DCT-IV perturbation studies in Appendices C and D empirically describe the initializations considered.

5 Experiments

The training framework on the full isometric tensor-network family is tested on DIV2K and Quick Draw line drawings. DIV2K images are the high-resolution split, converted to grayscale, center-cropped to a square, and Lanczos-resampled to 256×256. Quick Draw images are the dataset's 28×28 rasterized bitmaps zero-padded to 32×32. Image pixels in both DIV2K and Quick Draw are scaled to $[0, 1]$. Every dataset contributes a fixed 500-image training slice and 100 test images, drawn once with a fixed seed shared by every method.

All learned bases are trained at $\rho = 0.1$ and evaluated at five keep ratios. The two datasets sit at opposite sides of the AR(1)–Gaussian family with covariance $R_{ij} = \sigma_{\mathrm{AR}}^2 \rho_{\mathrm{AR}}^{|i-j|}$. The measured lag-1 autocorrelation (Fig. 3) places DIV2K near the $\hat{\rho}_{\mathrm{AR}} \to 1$ limit, where the DCT-II is asymptotically optimal (Appendix A.4), and Quick Draw well below it. Quality is reported as the peak signal-to-noise ratio $\mathrm{PSNR} = 10 \log_{10}(1/\mathrm{MSE})$ in dB computed for each of the 100 test images, with the mean squared error $\mathrm{MSE} = \mathcal{L}_k / 2^{m+n}$ the loss per pixel, and reconstructions clipped to the real pixel range before scoring. Table 2 benchmarks

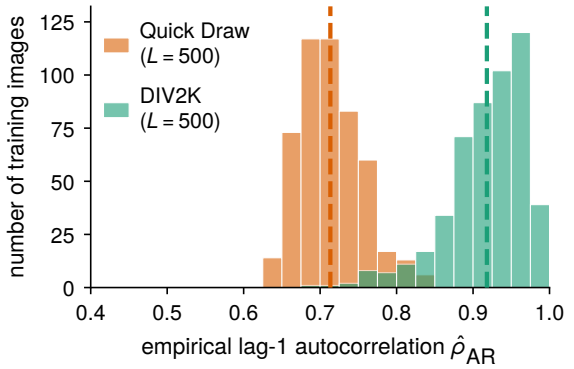

Fig. 3 Empirical lag-1 autocorrelation $\hat{\rho}_{\mathrm{AR}} = \frac{1}{2}(\hat{\rho}_{\mathrm{row}} + \hat{\rho}_{\mathrm{col}})$ across each 500-image training slice. Dashed vertical lines mark the dataset means

the family against JPEG's 8×8 block DCT-II and DFT references. End-to-end learned codecs are excluded because their bits-per-pixel rate is not comparable to ρ.

5.1 Topology comparison

We compare the six learned bases against the classical references on both datasets in Table 2. MERA needs a power-of-two qubit count per axis, so five compete on Quick Draw.

The DCT-IV has the highest learned-basis mean at two of the five DIV2K keep ratios and four of the five Quick Draw ones, while RichBasis takes the remaining DIV2K means. One exception is sharp rather than marginal. At DIV2K $\rho = 0.01$ RichBasis falls to 14.54 dB while DCT-IV reaches 21.92 dB. Retaining 1% of the coefficients is a

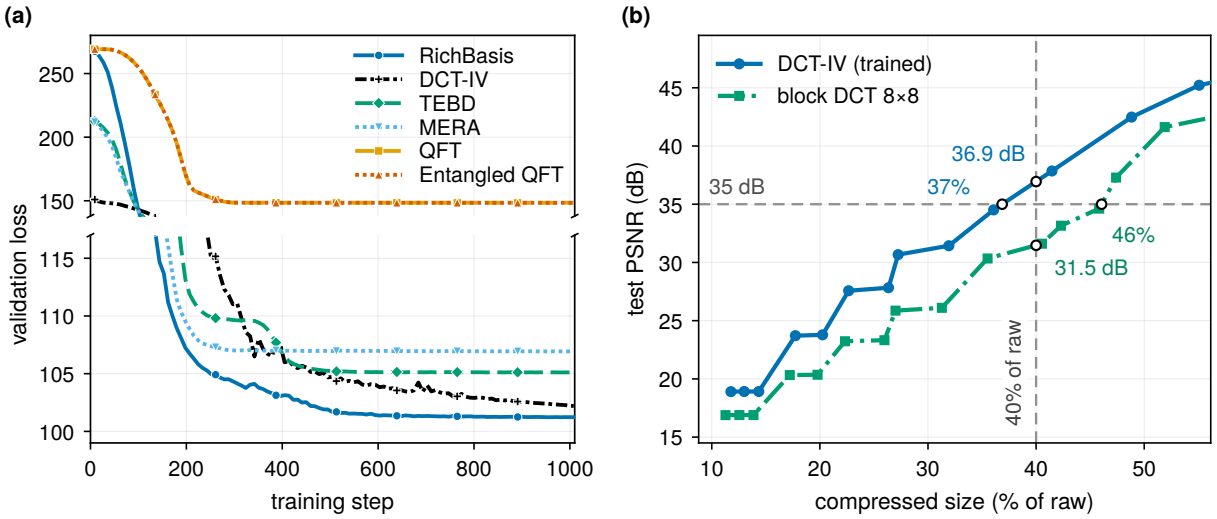

Fig. 4 Training and compression of the learned bases. (a) Validation loss on DIV2K (broken vertical axis). (b) Quick Draw rate–distortion under the byte-level codec. Open circles mark crossings with the dashed guides. At 35 dB the trained DCT-IV needs 37% of raw against the 8×8 block DCT's 46% (20% fewer bytes), and at 40% of raw it reaches 36.9 dB against 31.5 dB

tenfold extrapolation beyond the $\rho = 0.1$ budget the bases were trained at, and the objective is blind to the ordering *within* the retained set. Averaged over test images, the retained energy fraction $\|\text{top-k}(T\,\mathbf{x},\,k)\|_F^2/\|\mathbf{x}\|_F^2 = 1 - \mathcal{L}_k/\|\mathbf{x}\|_F^2$ exceeds 99% for both bases at their top 10% of coefficients, but the DCT-IV, anchored at the cosine transform's steeply ordered spectrum, concentrates 96% into its top 1% where the freely trained RichBasis keeps only 84%.

Across the whole table, the local gate manifold separates the learned bases more than the wiring does. TEBD, MERA, and RichBasis carry the same general two-qubit $U(4)$ gate under ring, hierarchical, and all-to-all wiring, yet sit within 0.9 dB of one another at every DIV2K ratio above 0.01. The DCT-IV, whose controlled X gates relax over the full $O(4)$ rather than a diagonal manifold in Section 3.2, lands in the same group. QFT and Entangled QFT, which keep the diagonal $U(1)^4$ phase, fall 1.0 to 3.4 dB below that group on DIV2K as ρ grows, and up to 11.1 dB on Quick Draw. They are numerically tied: the coupling, applied after the transform, cannot alter magnitude-top-k reconstruction. Figure 4a shows the same split in the validation loss $\frac{1}{|\mathcal{V}|}\sum_{\mathbf{x}\in\mathcal{V}}\mathcal{L}_k(\theta;\mathbf{x})$, the $|\mathcal{V}| = 75$ images held out from each training slice: the four dense-gate bases floor within 6 of one another and the diagonal pair floors roughly 45 above the group mean.

Against the classical references, every learned basis beats the 8×8 DFT at every keep ratio on both datasets. The cosine reference is the harder one. On Quick Draw the three dense-gate bases (RichBasis, TEBD, DCT-IV) beat the 8×8 DCT-II at every ratio, by up to 5.6 dB, while QFT and Entangled QFT lead it only at $\rho = 0.01$. On AR(1)-like DIV2K the best learned basis leads only under aggressive truncation and trails from $\rho = 0.10$ upward. The learned bases pay off where energy compaction is scarcest, the regime the next section prices in bytes.

The example reconstructions in Figs. 5 and 6 add what the table's mean PSNRs cannot show. The dense-gate bases preserve the sketch strokes that the diagonal pair blurs, and on DIV2K the learned circuits, apart from the RichBasis collapse at $\rho = 0.01$ noted above, avoid the block artifacts of the 8×8 references under aggressive truncation.

5.2 Compression at matched quality

Figure 4b prices the reconstruction quality of Table 2 in bytes, and the trained basis wins on

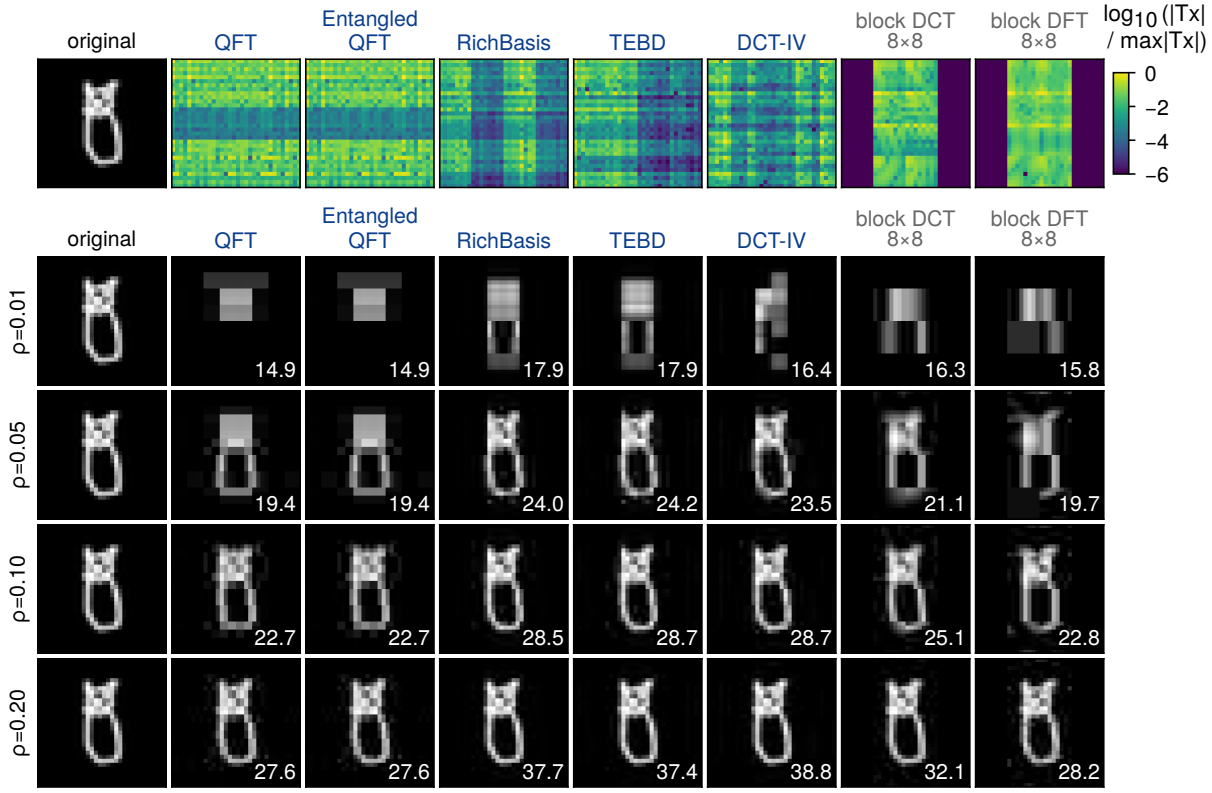

Fig. 5 Frequency-domain coefficient magnitude (top, $\log_{10}(|T\mathbf{x}|/\max|T\mathbf{x}|)$, peak-normalized per panel) and reconstructions at keep ratios ρ (per-cell PSNR in dB) for a Quick Draw cat sketch (32×32). The leftmost cell is the original. The columns are the five trained bases (QFT, Entangled QFT, RichBasis, TEBD, DCT-IV) in the order of Table 2, then the 8×8 classical DCT and DFT references, with learned bases in blue and classical references in gray. MERA is omitted, as Quick Draw's qubit count per axis, 5, is not a power of two. Image from Quick Draw (Google 2017; CC BY 4.0)

both axes of the rate–distortion plane. Horizontally, at matched quality, reaching 35 dB costs the trained DCT-IV 20% fewer bytes than JPEG's fixed 8×8 block DCT (about 30% fewer near 27 dB). Vertically, at matched size, the trained basis gains +5.4 dB at 40% of raw (8 bits per pixel), so at equal storage the block DCT's mean-squared pixel error is about 3.5 times larger. These two operating points are our main compression result.

The comparison holds the codec fixed and changes only the basis. Each image is transformed, the k largest coefficients are kept and uniformly quantized to b bits, and the values and positions are entropy-coded into a file, the standard transform-coding chain, with the byte layout in the benchmark repository. The basis itself is one parameter file per dataset and is excluded from the reported sizes, as are the transform definitions of the classical references. Sweeping the keep ratio and the bit depth $b \in \{6, 8, 10\}$ traces one curve per plotted basis.

A possible explanation is the spatial extent of the content. A sketch stroke runs across the frame while an 8×8 block sees it only in fragments, so a full-image basis may reach a matched quality with fewer coefficients, each of which would otherwise be stored. The saving is essentially the same when the stored corpus is the training corpus itself (19.7% at 35 dB). Throughout, the comparison is basis against basis under the same lossy codec and budget, not between complete codecs.

6 Conclusion

6.1 Summary

Our contribution is a trainable family of unitary image bases built from the classical fast transforms. Reading the FFT and the DCT-IV as isometric tensor networks and relaxing each gate

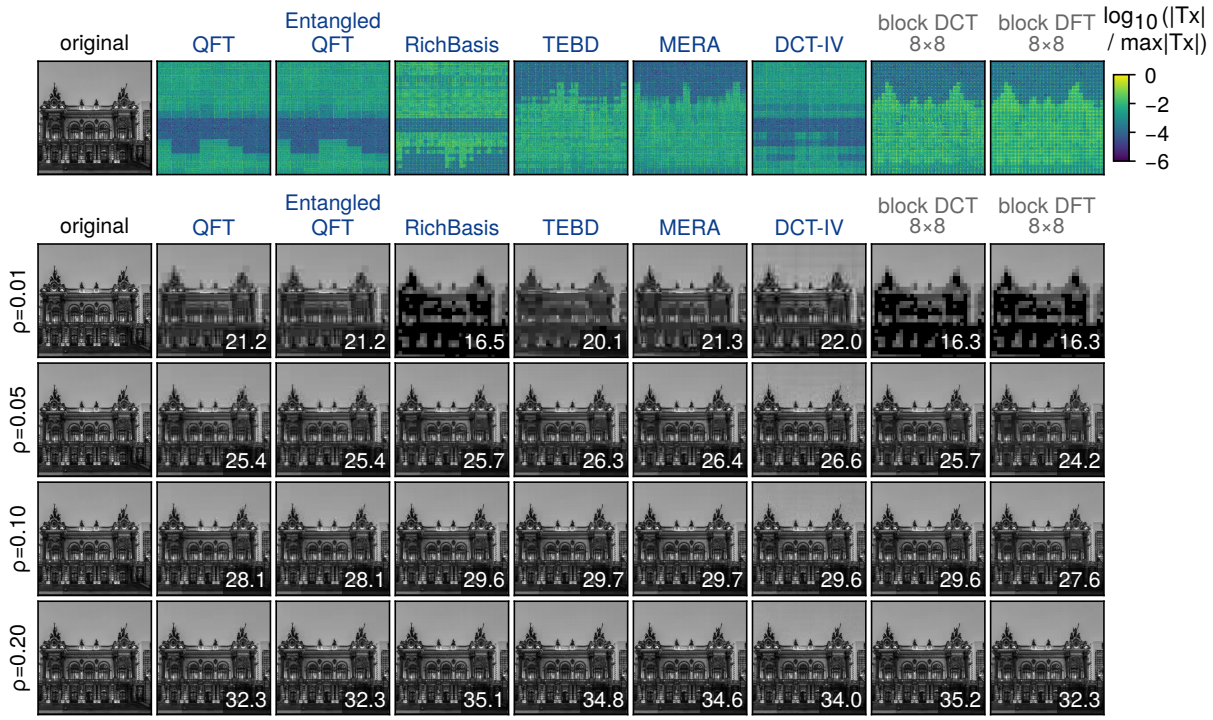

Fig. 6 Frequency-domain coefficient magnitude (top, $\log_{10}(|T\mathbf{x}|/\max|T\mathbf{x}|)$, peak-normalized per panel) and reconstructions at keep ratios ρ (per-cell PSNR in dB) for a DIV2K façade (256×256, drawn from the training slice). The leftmost cell is the original. The columns are all six trained bases of Table 2, including MERA, in its order, then the 8×8 classical DCT and DFT references, with learned bases in blue and classical references in gray. Image from DIV2K (Agustsson and Timofte 2017)

within its matrix manifold gives bases that contain these transforms as exact points, apply in near-linear time, and add only polylogarithmically many parameters. We train one shared basis per dataset with Riemannian optimization, which keeps every gate unitary throughout training, so each trained basis remains invertible.

The experiments show that, within the tested family, the local gate manifold matters more than the wiring: the dense-gate bases lead the diagonal-phase pair at every keep ratio from 0.05 upward, while rewiring the same $U(4)$ gate changes the DIV2K means by less than 1 dB above $\rho = 0.01$. Against the 8×8 block DCT, the outcome depends on the dataset. On Quick Draw, DCT-IV, TEBD, and RichBasis have a higher mean PSNR at all five tested keep ratios, while QFT and Entangled QFT lead only at $\rho = 0.01$. On DIV2K, the best learned mean leads only at the two lowest keep ratios. Under a common codec, the trained DCT-IV stores Quick Draw images in about 20% fewer bytes at 35 dB, excluding the one-off storage of its parameters.

6.2 Outlook and limitations

The limited effect of wiring changes in the tested family (Section 5.1) motivates exploring other training objectives, and the trained bases may also serve tasks beyond compression.

On the optimization side, the objective can move beyond hard top-k truncation. The loss of Eq. (1) is blind to the ordering within the retained set (the RichBasis drop at $\rho = 0.01$ in Section 5.1) and never sees the quantizer or the entropy coder that the codec of Section 5.2 applies afterward. The hard selection could give way to soft thresholding, the proximal operator of the ℓ_1 norm (Donoho 1995), which shrinks every coefficient toward zero by a threshold instead of keeping exactly k. The reconstruction then varies continuously with the coefficients, and training targets an ℓ_1-type sparsity rather than a fixed budget. The squared error has many alternatives as

well. Objectives averaged over several keep ratios, perceptual criteria such as the structural similarity index (Wang et al. 2004), and rate–distortion losses evaluated on the coded byte stream would each train the basis against a quantity closer to how it is used. Training cost has similar room. Each step pays the gate-by-gate contraction cost of Appendix A.3, and fusing recursion levels where the two-leg gates stay diagonal, or optimizing the contraction order outright (Liu et al. 2021, 2023), would lower it toward the fast-transform schedule and admit larger registers and deeper topologies.

Beyond compression, Fourier-type bases serve several other tasks that a trained basis could take on. A related trainable relaxation of the QFT already outperforms fixed transforms in image inpainting (An and Slavakis 2026). The coefficients of a trained basis also give a data-adapted frequency description of a dataset for spectral analysis, as in Figs. 5 and 6. Fourier neural operators apply a fixed FFT in each layer (Li et al. 2021), which a trained fast basis could replace. Finally, each basis is itself a quantum circuit on $m + n$ qubits, so it could act as a trained circuit in quantum kernel methods (Havlíček et al. 2019; Schuld and Killoran 2019). Since a unitary alone leaves the overlap of two states unchanged, such a kernel would need to compare the states after measurement or truncation.

Two limitations bound the result. Each basis is trained offline for one dataset, so the method is a basis-design tool rather than a deployable codec, and the topology is fixed in advance. Comparisons against wavelets, including circuit-designed wavelet families (Evenbly and White 2018; McCord and Evenbly 2022), and against learned dictionaries would place the family within the broader transform-coding toolbox. Other corpora that sit away from the AR(1) regime, such as diagrams or the frames of drawn and animated video, are natural targets for the same train-once recipe but remain untested.

Acknowledgements. We thank Lei Wang from the Institute of Physics, Chinese Academy of Sciences, for the helpful discussion about the JPEG algorithm. The authors are grateful to the Institute of Science Tokyo for providing the research environment.

Statements and Declarations

Funding. This work was supported in part by the National Key R&D Program of China (Grant No. 2024YFB4504004), the National Natural Science Foundation of China (Grant No. 12404568), and the Japan Science and Technology Agency (JST) SPRING program (Grant No. JPMJSP2180).

Competing interests. The authors have no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. The DIV2K (Agustsson and Timofte 2017) and Quick Draw (Google 2017) datasets used in this study are publicly available. The trained bases and per-image evaluation records are archived on Zenodo as version 0.1.0 of the benchmark repository (An et al. 2026a), which is developed at https://github.com/zazabap/pdft-benchmarks.

Materials availability. Not applicable.

Code availability. The open-source implementation `pdft` is archived on Zenodo as version 0.3.0 (An et al. 2026b), the version behind every reported result, and is developed at https://github.com/zazabap/pdft. The benchmark suite with its reproduction scripts is archived with the data (An et al. 2026a).

Author contributions. Shiwen An, as first author, was involved in every part of the project, including the simulations, the code, and drafting the manuscript. Zhongyi Ni completed the prototype and reviewed the theory. Huanhai Zhou assisted with the code and the benchmark validation. Jin-Guo Liu supervised the project, guiding it with instructions and revisions throughout. All authors read and approved the final manuscript.

Use of AI tools. AI-assisted tools (Claude, Anthropic; and Codex, OpenAI) were used for language editing and for consistency checks across the manuscript, references, code, and archived outputs. All source verification, statistical and scientific decisions, and final content were independently reviewed and approved by the authors, who take full responsibility for the work.

Appendix A Network derivations

This appendix derives the parametric tensor networks trained in the main text. Starting from the Cooley–Tukey FFT, we apply local transformations so that every remaining local gate is a free element of a unitary or orthogonal manifold. The Fig. 7 shows the four transformations for an $n = 4$-leg example.

A.1 FFT decomposition

The DFT matrix of size $N = 2^n$ has entries $(F_N)_{i,j} = \omega^{(i-1)(j-1)}$ with $\omega = e^{-2\pi i/N}$ the primitive N-th root of unity, so F_N is the Vandermonde matrix

$$F_N = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{pmatrix}. \quad \text{(A1)}$$

Split the columns (the input index) at $N/2$ into a top half $\mathbf{x}_{\text{top}}$ and a bottom half $\mathbf{x}_{\text{bot}}$, and the rows (the output index) by the parity of the output frequency $i-1$. Since $\omega^{(i-1)(j+N/2-1)} = (-1)^{i-1}\,\omega^{(i-1)(j-1)}$, the rows computing even output frequencies add the two column halves and those computing odd frequencies subtract them, and each of the four resulting $(N/2)\times(N/2)$ blocks reduces to the half-size DFT up to the twiddle $\Omega_{N/2} = \mathrm{diag}(1, \omega, \ldots, \omega^{N/2-1})$:

$$\begin{aligned} [F_N]^{\text{even}}_{\text{left}} &= [F_N]^{\text{even}}_{\text{right}} = F_{N/2}, \\ [F_N]^{\text{odd}}_{\text{left}} &= -[F_N]^{\text{odd}}_{\text{right}} = F_{N/2}\,\Omega_{N/2}, \end{aligned} \quad \text{(A2)}$$

using $(\omega^2)^{(i-1)(j-1)} = (F_{N/2})_{i,j}$ on both row parities, $\omega^N = 1$ on the right-even block, and $\omega^{N/2} = -1$ on the right-odd block. Assembling the four blocks gives the Cooley–Tukey factorization

$$F_N\mathbf{x} = P_N \begin{pmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{pmatrix} \begin{pmatrix} I_{N/2} & I_{N/2} \\ \Omega_{N/2} & -\Omega_{N/2} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{\text{top}} \\ \mathbf{x}_{\text{bot}} \end{pmatrix}. \quad \text{(A3)}$$

Here P_N is the one-level perfect-shuffle interleaver. If $\mathbf{z} = (\mathbf{z}_{\text{e}}^{\mathsf{T}}, \mathbf{z}_{\text{o}}^{\mathsf{T}})^{\mathsf{T}}$ stores all even outputs before all odd outputs, then $(P_N\mathbf{z})_{2r} = (z_{\text{e}})_r$ and $(P_N\mathbf{z})_{2r+1} = (z_{\text{o}})_r$ for $0 \le r < N/2$. Thus P_N only restores natural output order at this level; the interleavers accumulated over all recursion levels compose into the full bit reversal of the standard QFT. Applied recursively, each level reduces an N-point DFT to two $(N/2)$-point DFTs plus $O(N)$ twiddle multiplications, giving total cost $O(N \log N)$. Equation (A3) is the input to the tensor-network construction.

Matrix-to-network construction.

The remaining steps preserve the matrix in Eq. (A3) while exposing its factors as local gates in Fig. 7.

Re-index (Fig. 7a). A vector of length $N = 2^n$ carries a single integer index that we write in most-significant-bit-first order as $i = 2^{n-1}q_0 + 2^{n-2}q_1 + \cdots + q_{n-1}$. Re-interpreting $\mathbf{x}$ as a rank-n tensor with one binary leg q_j per bit is purely notational, but it turns every linear operator in Eq. (A3) into a tensor-network component acting on specified legs.

Recurse (Fig. 7b). Take q_0 as the most-significant bit splitting Eq. (A3) into its top and bottom halves. The combined butterfly–twiddle on the input side factors through the identity

$$\begin{pmatrix} I_{N/2} & I_{N/2} \\ \Omega_{N/2} & -\Omega_{N/2} \end{pmatrix} = \begin{pmatrix} I_{N/2} & 0 \\ 0 & \Omega_{N/2} \end{pmatrix} (\tilde{H} \otimes I_{N/2}), \quad \text{(A4)}$$

with $\tilde{H} = \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ the unnormalized Hadamard matrix. Substituting Eq. (A4) into Eq. (A3) and reading q_0 as the leading binary leg gives the single-level factorization described in words in Section 3.1,

$$F_N = P_N\,(I_2 \otimes F_{N/2}) \begin{pmatrix} I_{N/2} & 0 \\ 0 & \Omega_{N/2} \end{pmatrix} (\tilde{H} \otimes I_{N/2}), \quad \text{(A5)}$$

reading Eq. (A5) right to left: a Hadamard factor on q_0, a diagonal twiddle controlled by that leg, the half-size DFT $F_{N/2}$ on $(q_1, \ldots, q_{n-1})$, and the one-level interleaver P_N returning the outputs to natural order.

Decompose (Fig. 7c). The controlled twiddle factors into a chain of two-leg phase gates,

$$\begin{aligned} \begin{pmatrix} I_{N/2} & 0 \\ 0 & \Omega_{N/2} \end{pmatrix} &= \prod_{j=1}^{n-1} \mathrm{CP}_{0,j}(\Lambda_j), \\ \Omega_{N/2} &= \mathrm{diag}(1, \omega^{N/4}) \otimes \cdots \otimes \mathrm{diag}(1, \omega). \end{aligned} \quad \text{(A6)}$$

Thus the twiddle is a chain of $n-1$ diagonal gates $\mathrm{CP}_{0,j}(\Lambda_j)$, each joining q_0 to a lower-significance

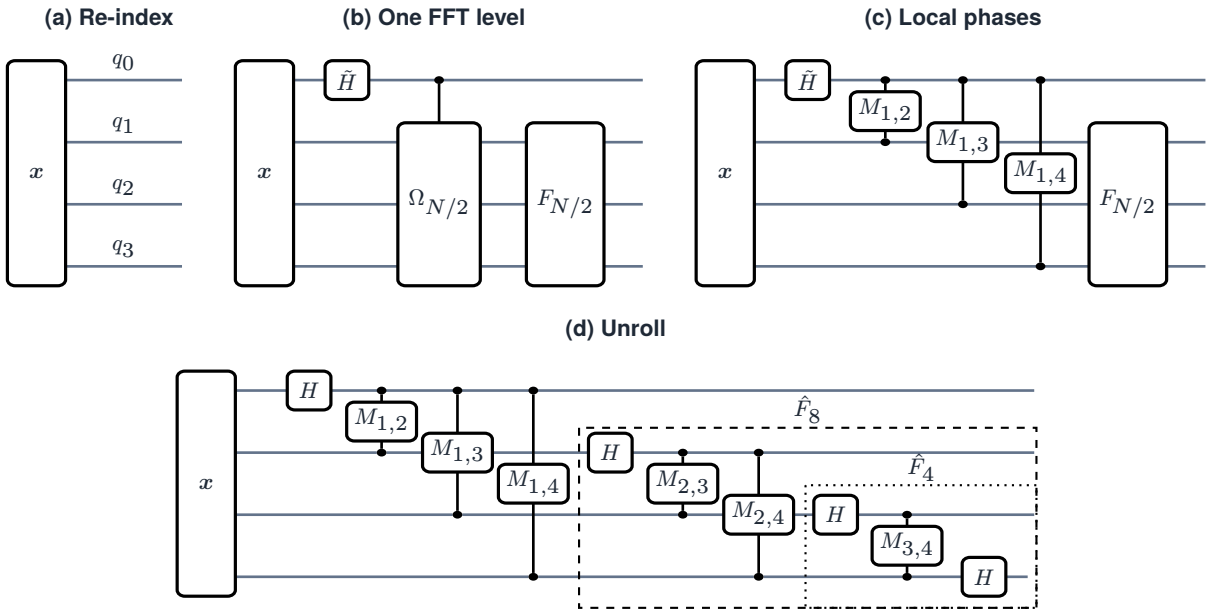

Fig. 7 FFT-to-QFT construction for $N = 16$: (a) binary re-indexing, (b) one Cooley–Tukey level, (c) local phase factors, and (d) recursive unrolling and relaxation. Panels (a)–(c) preserve the matrix. Panel (d) normalizes by $N^{-1/2}$ and relaxes each gate within its manifold. Here $\tilde{H}$ is the unnormalized Hadamard, $\Omega_{N/2}$ the twiddle factors, and $M_{i,j}$ the controlled phases. Dashed and dotted boxes mark the normalized children $\widehat{F}_8$ and $\widehat{F}_4$ ($\widehat{F}_N = N^{-1/2}F_N$), and fixed output reordering is omitted

leg q_j with the 2×2 block $\Lambda_j = \mathrm{diag}\big(1, \omega^{2^{n-1-j}}\big)$. These are the level-one instances of the designed controlled phases of Section 3.1: since leg q_j is bit $j+1$, the gate $\mathrm{CP}_{0,j}(\Lambda_j)$ is, up to the conjugation fixed below, the controlled phase $M_{1,j+1}$ drawn in Fig. 1b, and the recursion on $F_{N/2}$ supplies the pairs $M_{i,j}$ with $i > 1$. Here $\omega = e^{-2\pi i/N}$, while the quantum-circuit convention of Section 3.1 and the `pdft` implementation use the conjugate root $e^{2\pi i/N}$, under which the identical derivation yields the positive phases $\phi_{i,j} = 2\pi/2^{j-i+1}$.

Relax (Fig. 7d). Recursing on $F_{N/2}$ replays the butterfly-and-decomposition pair on $(q_1, \ldots, q_{n-1})$, down to $F_2 = \tilde{H}$ on the final leg. The fixed per-level interleavers compose across the recursion into the usual output bit reversal. For the unitary QFT, normalize F_N to $\widehat{F}_N = N^{-1/2}F_N$, so each butterfly is the normalized Hadamard $H = 2^{-1/2}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ of Section 3.1. Relaxation then replaces each fixed local gate by a trainable element of its manifold (Table 3). The resulting network $T(\theta)$ contains the normalized DFT at the Fourier point and is unitary at every parameter setting.

A.2 DCT-IV decomposition

For $N = 2^n$, the orthonormal DCT-IV has entries $(C_N^{\mathrm{IV}})_{u,v} = \sqrt{2/N}\cos\psi_{u,v}$, with kernel angle $\psi_{u,v} = \pi(2u+1)(2v+1)/(4N)$. We expose its radix-2 factorization directly as a tensor network on binary wires $q_0, q_1, \ldots, q_{n-1}$; q_0 selects the recursion branch and the remaining wires carry the child index. Radix-2 DCT-IV factorizations are classical; we use the recursive sparse form from the factorization survey (Plonka and Tasche 2005), read within the algebraic Cooley–Tukey framework (Püschel and Moura 2008).

Two kernel identities.

The construction rests on two elementary facts about the kernel angle $\psi_{u,v}$, both following directly from its definition. First, an input and its mirror collapse together,

$$\cos\psi_{u,\,N-1-v} = (-1)^u \sin\psi_{u,v}, \qquad \text{(A7)}$$

so x_v and x_{N-1-v} reach each output u through a single sine–cosine pair whose sine term carries the $(-1)^u$ sign; this is why Q_N below pairs v with $N-1-v$ and why one branch sign accounts for the whole mirror. Second, the even and odd parent

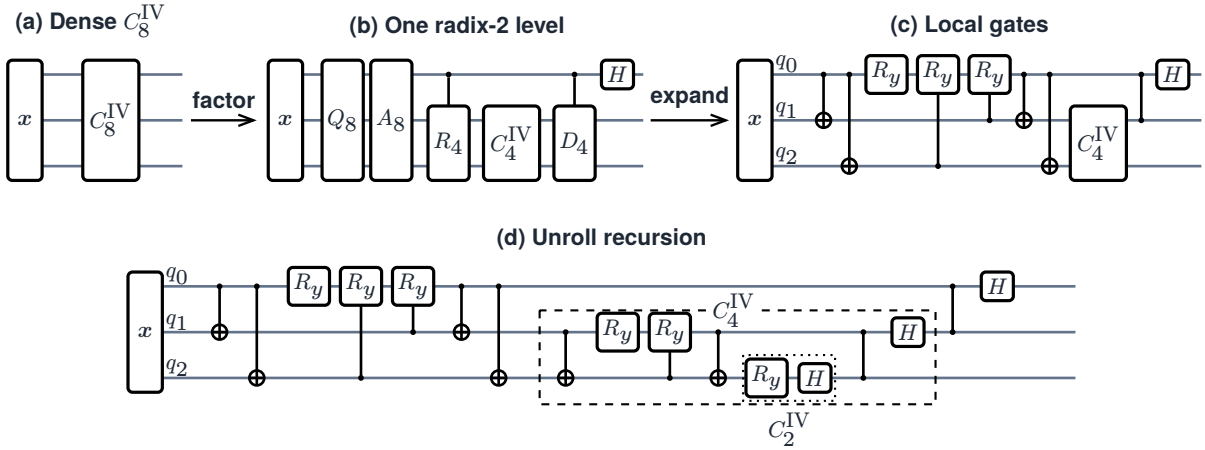

Fig. 8 DCT-IV tensor-network decomposition at $N = 8$. (a) The starting dense matrix C_8^{IV}. (b) One radix-2 level, equal to C_8^{IV} up to output order: Q_8 pairs each input with its mirror, A_8 rotates each pair, R_4 reverses the lower branch, both branches run the same child C_4^{IV}, D_4 applies the sign correction, and H merges the branches. (c) The local forms of Q_8, A_8, R_4, and D_4 as one- and two-qubit gates. (d) The recursion unrolled into an exact R_y network. Dashed and dotted boxes mark the children C_4^{IV} and C_2^{IV}, and the trainable manifolds are listed in Table 3. Dots mark controls, $\oplus$ marks a CNOT target, and a dot pair is a controlled Z. Fixed output wiring is omitted

outputs $2w$ and $2w+1$ that share a child output w differ only by a local offset,

$$\psi_{2w,v} = \alpha_{w,v} - \delta_v, \qquad \psi_{2w+1,v} = \alpha_{w,v} + \delta_v, \quad \text{(A8)}$$

with $\alpha_{w,v} = \frac{\pi(2w+1)(2v+1)}{2N}$ the length-$N/2$ DCT-IV kernel angle and $\delta_v = \frac{\pi(2v+1)}{4N}$ independent of w. A single rotation by δ_v on the input pair therefore prepares both outputs, and the shared child $C_{N/2}^{\mathrm{IV}}$ supplies the common $\alpha_{w,v}$ projection.

Matrix-to-network construction.

One radix-2 level factors the transform as

$$C_N^{\mathrm{IV}} = P_N (H \otimes I_{N/2})(I_{N/2} \oplus D_{N/2})(I_2 \otimes C_{N/2}^{\mathrm{IV}}) \\ (I_{N/2} \oplus R_{N/2})\, A_N Q_N. \quad \text{(A9)}$$

Read right to left, the factors pair and rotate the inputs, run the same half-size child on both branches, and merge the branches back into natural output order (Fig. 8b). The stages below define each factor and reduce it to one- and two-qubit gates while preserving the DCT-IV matrix.

Re-index (Fig. 8a–b). Intermediate vectors use branch-major order (q_0, v): positions v and $v + N/2$ carry the two recursion branches for the same child index v. For $0 \le v < N/2$, define

$$(R_{N/2}\mathbf{z})_v = z_{N/2-1-v}, \qquad Q_N = I_{N/2} \oplus R_{N/2},$$
$$(P_N\mathbf{z})_{2v} = z_v, \qquad (P_N\mathbf{z})_{2v+1} = z_{v+N/2}.$$

Thus Q_N pairs each input with its mirror, and P_N interleaves the two output branches. Both are fixed permutations. The recursion's separate factor $I_{N/2} \oplus R_{N/2}$ is the same matrix as Q_N, kept written out because it plays the branch-reversal role there rather than the pairing role.

Rotate (Fig. 8b). This factor applies the offset δ_v of the parity split Eq. (A8) to each pair. With $\Gamma_{N/2} = \mathrm{diag}(\cos\delta_v)$ and $\Sigma_{N/2} = \mathrm{diag}(\sin\delta_v)$, the block-diagonal pair rotation is

$$A_N = \begin{pmatrix} \Gamma_{N/2} & -\Sigma_{N/2} \\ \Sigma_{N/2} & \Gamma_{N/2} \end{pmatrix},$$

with pairwise action

$$\begin{pmatrix} (A_N\mathbf{z})_v \\ (A_N\mathbf{z})_{v+N/2} \end{pmatrix} = R_y(2\delta_v) \begin{pmatrix} z_v \\ z_{v+N/2} \end{pmatrix}.$$

Here $R_y(\vartheta) = \left(\begin{smallmatrix} \cos(\vartheta/2) & -\sin(\vartheta/2) \\ \sin(\vartheta/2) & \cos(\vartheta/2) \end{smallmatrix}\right)$. Since $\Gamma_{N/2}$ and $\Sigma_{N/2}$ are diagonal, different child indices never mix.

Recurse (Fig. 8b). Both branches use the same $C_{N/2}^{\mathrm{IV}}$ child, applied uncontrolled as $I_2 \otimes C_{N/2}^{\mathrm{IV}}$. The lower branch is reversed before the child by $I_{N/2} \oplus R_{N/2}$ and receives the sign correction $D_{N/2} = \mathrm{diag}((-1)^v)$ afterward. Only the shared child recurses, so no second transform template or transform-type ancilla is required.

Decompose (Fig. 8c). The lower wires encode $v = \sum_{p=1}^{n-1} 2^{n-1-p} q_p$, so reversal complements

every lower wire:

$$I_{N/2} \oplus R_{N/2} = \prod_{p=1}^{n-1} (I_2 \oplus X)_{q_0,q_p},$$

a chain of controlled-NOT (CNOT) gates controlled by q_0, with $X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ acting on each lower wire as listed in Table 3. Likewise, the affine angle $2\delta_v = \pi/(2N) + \sum_{p=1}^{n-1} q_p 2^{n-1-p}\pi/N$ gives

$$A_N = \left[\prod_{p=1}^{n-1} \mathrm{C}R_y\left(\frac{2^{n-1-p}\pi}{N}\right)_{q_p \to q_0}\right] R_y\left(\frac{\pi}{2N}\right)_{q_0}.$$

For $N \geq 4$, the sign correction in natural child order is

$$I_{N/2} \oplus D_{N/2} = (I_2 \oplus Z)_{q_0,q_{n-1}}.$$

Because the child's output bit reversal is deferred to the end, the parity bit of the child output sits on the wire adjacent to q_0, so this controlled Z acts on (q_0, q_1) as drawn in Fig. 8c.

Unroll (Fig. 8d). Recursively replace the shared $C^{\mathrm{IV}}_{N/2}$ box by the same cell down to $C^{\mathrm{IV}}_1 = [1]$. The fixed P_N factors accumulate into one output bit reversal; all other gates are listed with their trainable manifolds in Table 3.

A.3 Gate counts and manifolds

For a one-dimensional transform of length $N = 2^n$, the unrolled QFT contains n Hadamard-role gates and $n(n-1)/2$ controlled phases. Under the gate-wise relaxation of Table 3, its trainable manifold therefore has $4n + 4n(n-1)/2 = O(n^2) = O(\log^2 N)$ real parameters. The DCT-IV has the same recursion depth but a larger fixed local dictionary. At a level with r lower binary legs, its two reversal fans contribute $2r$ two-leg X gates, A_N contributes one base R_y and r controlled R_y gates, and the level ends with one normalized merge H and, for $r > 0$, one controlled Z. Summing over $r = 0, \ldots, n-1$ gives

$$N_{\mathsf{X}} = 2\sum_{r=0}^{n-1} r = n(n-1), \qquad N_{\mathrm{C}R_y} = \tfrac{n(n-1)}{2},$$
$$N_{R_y} = n, \qquad N_H = n, \qquad N_{\mathsf{Z}} = n-1. \tag{A10}$$

Thus both exact networks contain $O(n^2) = O(\log^2 N)$ fixed-size gates. For the DCT-IV, the omitted interleavers compose into one fixed output bit reversal, and no auxiliary leg is introduced. Induction on Eq. (A9), with base case $C^{\mathrm{IV}}_1 = [1]$, proves that the unrolled gates implement the orthonormal DCT-IV up to that output order.

For a $2^m \times 2^n$ image the separable transforms $F_{2^m} \otimes F_{2^n}$ and $C^{\mathrm{IV}}_{2^m} \otimes C^{\mathrm{IV}}_{2^n}$ add these counts across the two registers, so both stay $O(m^2 + n^2)$ in gate and parameter count. Applying the gates one at a time costs $O\big(2^{m+n}(m^2+n^2)\big)$ operations, the schedule the batched evaluation of Appendix B.3 executes; fusing each recursion level would recover the $O(2^{m+n}(m+n))$ fast-transform schedule. Fusing survives relaxation only where every gate of a level stays structured, as the diagonal $U(1)^4$ bonds do; the dense $U(4)$ and $O(4)$ bonds of RichBasis and the trained DCT-IV break it, so both retain the stepped cost, as the Cost row of Table 1 records.

Relaxation manifolds.

Every gate factor is one of three matrix manifolds, listed here with the dimensions the parameter counts of Table 1 add up to:

$$U(d) = \{U \in \mathbb{C}^{d\times d} : U^\dagger U = I_d\},$$
$$O(d) = \{Q \in \mathbb{R}^{d\times d} : Q^\top Q = I_d\},$$
$$U(1)^q = \{(z_1, \ldots, z_q) \in \mathbb{C}^q : |z_r| = 1\},$$

of real dimension d^2, $\frac{1}{2}d(d-1)$, and q respectively, so the factors that appear are $\dim U(2) = 4$, $\dim U(4) = 16$, $\dim O(2) = 1$, $\dim O(4) = 6$, and $\dim U(1)^4 = 4$. A point of $U(1)^q$ enters the network as the diagonal gate $\mathrm{diag}(z_1, \ldots, z_q)$.

The orthogonal factors are disconnected, $O(2)$ and $O(4)$ each splitting into a $\det = +1$ and a $\det = -1$ component. The DCT-IV rotation gates and controlled-rotation blocks start in the rotation component of $O(2)$, its merge gates in the reflection component at $R_y(0)H = H$, and the relaxed mirror gates in the $\det = -1$ component of $O(4)$ at the CNOT $I_2 \oplus X$; because training moves continuously, each gate stays in the component chosen at initialization, and paired components have the same dimension.

Table 3 is the complete dictionary: each fixed gate is replaced by a trainable element of one of these manifolds with the same wiring and arity, and the Diagram column gives the symbol that element carries in Fig. 2. The exact Fourier phase $M(\phi)$ is the point $\boldsymbol{\theta} = (0, 0, 0, \phi)$ of $U(1)^4$.

For a one-dimensional n-leg register (the coupled register pair in the Entangled QFT case), the gate-manifold products represented by the table

Table 3 Gate-wise relaxations of all six bases in `pdft`, with the group each gate trains on. References are exact for QFT, Entangled QFT, RichBasis, and DCT-IV, with the QFT's controlled phase between bits $i < j$ set to $\phi_{i,j} = 2\pi/2^{\,j-i+1}$. TEBD and MERA list their canonical gates (the Trotter gate $e^{-i\,\delta t\,h}$ of a bond Hamiltonian h over a time step δt, and the disentangler) and train from seeded values. Phase angles θ_r lie in $\mathbb{R}/2\pi\mathbb{Z}$, rotation angles ϑ in $\mathbb{R}/4\pi\mathbb{Z}$ with $c_\vartheta = \cos\frac{\vartheta}{2}$ and $s_\vartheta = \sin\frac{\vartheta}{2}$, and the reference angle ϑ_0 is fixed at its derived value.

Gate role	Reference gate	Relaxed gate	Group	Diagram
(1) *QFT* and *Entangled QFT* (which adds one matched row–column controlled phase per pair)				
Hadamard	$H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$	$U = \left(\begin{smallmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{smallmatrix}\right)$	$U(2)$	
Controlled phase	$M(\phi) = \left(\begin{smallmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&e^{i\phi} \end{smallmatrix}\right)$	$D(\boldsymbol{\theta}) = \left(\begin{smallmatrix} e^{i\theta_0} & 0 & 0 & 0 \\ 0 & e^{i\theta_1} & 0 & 0 \\ 0 & 0 & e^{i\theta_2} & 0 \\ 0 & 0 & 0 & e^{i\theta_3} \end{smallmatrix}\right)$	$U(1)^4$	
(2) *TEBD / MERA / RichBasis* (H as above; ring, hierarchical, and all-to-all wiring respectively)				
Two-qubit gate	$e^{-i\,\delta t\,h}$ (TEBD) disentangler (MERA) $M(\phi)$ (RichBasis)	$U^{(4)} = \left(\begin{smallmatrix} u_{00} & u_{01} & u_{02} & u_{03} \\ u_{10} & u_{11} & u_{12} & u_{13} \\ u_{20} & u_{21} & u_{22} & u_{23} \\ u_{30} & u_{31} & u_{32} & u_{33} \end{smallmatrix}\right)$	$U(4)$	
(3) *DCT-IV*				
Rotation	$R_y(\vartheta_0) = \left(\begin{smallmatrix} c_{\vartheta_0} & -s_{\vartheta_0} \\ s_{\vartheta_0} & c_{\vartheta_0} \end{smallmatrix}\right)$	$R_y(\vartheta) = \left(\begin{smallmatrix} c_\vartheta & -s_\vartheta \\ s_\vartheta & c_\vartheta \end{smallmatrix}\right)$	$O(2)$	
Merge	$H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$	$R_y(\vartheta)H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} c_\vartheta - s_\vartheta & c_\vartheta + s_\vartheta \\ c_\vartheta + s_\vartheta & s_\vartheta - c_\vartheta \end{smallmatrix}\right)$	$O(2)$	
Controlled rotation	$I_2 \oplus R_y(\vartheta_0) = \left(\begin{smallmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&c_{\vartheta_0} & -s_{\vartheta_0} \\ 0&0&s_{\vartheta_0} & c_{\vartheta_0} \end{smallmatrix}\right)$	$I_2 \oplus R_y(\vartheta) = \left(\begin{smallmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&c_\vartheta & -s_\vartheta \\ 0&0&s_\vartheta & c_\vartheta \end{smallmatrix}\right)$	$I_2 \oplus O(2)$	
Controlled Z (CZ)	$I_2 \oplus Z = \left(\begin{smallmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&-1 \end{smallmatrix}\right)$	$D(\boldsymbol{\theta})$ as above, $D(0,0,0,\pi) = I_2 \oplus Z$	$U(1)^4$	
Controlled X (CNOT)	$I_2 \oplus X = \left(\begin{smallmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{smallmatrix}\right)$	$V = \left(\begin{smallmatrix} v_{00} & v_{01} & v_{02} & v_{03} \\ v_{10} & v_{11} & v_{12} & v_{13} \\ v_{20} & v_{21} & v_{22} & v_{23} \\ v_{30} & v_{31} & v_{32} & v_{33} \end{smallmatrix}\right)$	$O(4)$	

are

$$
\begin{aligned}
\mathcal{M}^{(n)}_{\mathrm{QFT}} &= U(2)^n \times \big(U(1)^4\big)^{n(n-1)/2}, \\
\mathcal{M}^{(n)}_{\mathrm{TEBD}} &= U(2)^n \times U(4)^n, \\
\mathcal{M}^{(n)}_{\mathrm{MERA}} &= U(2)^n \times U(4)^{2(n-1)}, \\
\mathcal{M}^{(n)}_{\mathrm{Rich}} &= U(2)^n \times U(4)^{n(n-1)/2}, \\
\mathcal{M}^{(n)}_{\mathrm{DCT}} &= O(4)^{n(n-1)} \times O(2)^{n(n-1)/2+2n} \\
&\quad \times \big(U(1)^4\big)^{n-1}, \\
\mathcal{M}^{(m,n)}_{\mathrm{Ent}} &= \mathcal{M}^{(m)}_{\mathrm{QFT}} \times \mathcal{M}^{(n)}_{\mathrm{QFT}} \\
&\quad \times \big(U(1)^4\big)^{\min(m,n)}.
\end{aligned}
\tag{A11}
$$

The exponents count gates, one $U(2)$ factor per leg and one two-qubit factor per bond of the wiring drawn in Fig. 2 (the $n(n-1)/2$ all-to-all bonds of QFT and RichBasis, the n ring bonds of TEBD, and the $2(n-1)$ hierarchical bonds of MERA), or for the DCT-IV the role counts of Eq. (A10), so the number of factors is the Gates entry of Table 1.

The QFT and RichBasis products contain the exact QFT, since each $U^{(4)}$ gate of RichBasis can sit at the corresponding phase gate; Entangled QFT contains the separable QFT at zero coupling, and DCT-IV the bit-reversed exact transform. The ring and hierarchical wirings of TEBD and MERA offer no direct embedding of the QFT gate pattern at the sizes trained here. Section 4.3 gives each topology's initialization, and Appendix B.2 shows that every update keeps each gate unitary to machine precision, so the transform stays invertible throughout training; the gates assigned to $O(2)$ and $O(4)$ stay real only to about 10^{-3}. A 2-D basis uses independent row and column copies apart from Entangled QFT's matched-axis factors, and Table 1 counts every gate of one single-axis circuit, the coupled register pair in the Entangled QFT case.

A.4 DCT-IV versus DCT-II

The DCT-II used by JPEG is a strong coding baseline. It approaches the Karhunen–Loève basis of the AR(1) source of Section 5 as $\rho_{\mathrm{AR}} \to 1$ (Jain 1979; Britanak et al. 2007) and has classical fast factorizations of its own (Chen et al. 1977; Loeffler et al. 1989; Feig and Winograd 1992). The reason not to use it as the recursive anchor is structural.

Using the same reversal $R_{N/2}$ and interleaver P_N as above, the DCT-II radix-2 identity is

$$\begin{aligned} B_N &= \frac{1}{\sqrt{2}} \begin{pmatrix} I_{N/2} & R_{N/2} \\ I_{N/2} & -R_{N/2} \end{pmatrix}, \\ C_N^{\mathrm{II}} &= P_N \big(C_{N/2}^{\mathrm{II}} \oplus C_{N/2}^{\mathrm{IV}} \big) B_N. \end{aligned} \tag{A12}$$

Equation (A12) has two different children, $C_{N/2}^{\mathrm{II}}$ and $C_{N/2}^{\mathrm{IV}}$. It therefore does not define the one-template self-recursion of Fig. 8; an implementation of this recursion must retain two transform templates or add a transform-type leg (an ancilla) to select the child. In contrast, the repeated diagonal block in Eq. (A9) contains two identical copies of $C_{N/2}^{\mathrm{IV}}$, so no ancilla is required.

Appendix B Optimization details

B.1 Riemannian Adam update

This subsection records the implementation-specific update behind Section 4.2, in the same notation, with G_t a trainable gate at iteration t and g_t the ambient gradient $\nabla \mathcal{L}_k$ of the batch-mean loss of Appendix B.3, evaluated at G_t. For complex gates, the raw JAX gradient is first conjugated to follow the real-valued Wirtinger convention. The resulting gradient g_t at $G_t \in U(d)$ is projected as

$$\begin{aligned} \mathrm{proj}_G(g) &= G\, \mathrm{skew}(G^\dagger g), \\ \mathrm{skew}(A) &= \tfrac{1}{2}(A - A^\dagger). \end{aligned} \tag{B13}$$

For $O(d)$, the skew factor is already real, so the same formula applies; for a phase gate, $\mathrm{proj}_z(g) = i\, \mathrm{Im}(\bar{z} \odot g) \odot z$, with $\odot$ the elementwise product. These projections land in the tangent space at each point,

$$\begin{aligned} T_U\, U(d) &= \{UA : A^\dagger = -A\}, \\ T_Q\, O(d) &= \{QA : A^\top = -A\}, \end{aligned}$$

and $T_z\, U(1)^q = \{i\, \mathbf{t} \odot z : \mathbf{t} \in \mathbb{R}^q\}$; each tangent vector is the point itself times a skew factor, which is why the one projection formula covers the whole family. Writing $r_t = \mathrm{proj}_{G_t}(g_t)$ and $\widetilde{m}_{t-1}$ for the transported first moment, with $\widetilde{m}_0 = v_0 = 0$ and t counted from 1, the elementwise Adam moments are

$$m_t = \beta_1 \widetilde{m}_{t-1} + (1-\beta_1) r_t, \qquad \hat{m}_t = \frac{m_t}{1-\beta_1^t}, \tag{B14}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) |r_t|^2, \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}. \tag{B15}$$

Here v_t is a real elementwise accumulator and is not transported; this elementwise second moment departs from the per-factor scalar of Bécigneul and Ganea (2019). With $(\beta_1, \beta_2, \epsilon) = (0.9, 0.999, 10^{-8})$, the adaptive increment is

$$a_t = -\eta\, \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon). \tag{B16}$$

The elementwise division takes a_t off the tangent space; the skew in Eq. (B17) restores it. In the released benchmark runs of Table 2 and the perturbation study of Appendix D, the projected gradient is additionally clipped to a global norm of 1 before the moment update, and η follows a warmup and cosine-decay schedule; the robustness studies of Appendix C train at a fixed step size without clipping. For unitary gates, the implementation retracts this increment with the Cayley map quoted in Section 4.2,

$$\begin{aligned} W_t &= \mathrm{skew}(a_t G_t^\dagger), \\ G_{t+1} &= \big(I - \tfrac{1}{2} W_t\big)^{-1} \big(I + \tfrac{1}{2} W_t\big) G_t. \end{aligned} \tag{B17}$$

Because $W_t^\dagger = -W_t$, the Cayley factor is unitary. The real analogue preserves $O(d)$, while a phase gate is updated by normalizing $z_t + a_t$ to unit modulus. Finally, the first moment is transported by re-projection,

$$\begin{aligned} \widetilde{m}_t^{(G)} &= G_{t+1}\, \mathrm{skew}(G_{t+1}^\dagger m_t), \\ \widetilde{m}_t^{(z)} &= i\, \mathrm{Im}(\bar{z}_{t+1} \odot m_t) \odot z_{t+1}. \end{aligned} \tag{B18}$$

This projection transport is the first-order transport used by the optimizer, rather than exact geodesic parallel transport.

The released values of this schedule: the benchmark bases of Table 2 train for 112 epochs of 50-image minibatches (1008 steps), with η warmed up over the first 5% of steps to a peak of 3×10^{-3} and cosine-decayed to 3×10^{-4}; per-run values are recorded in each released basis's run-environment file.

B.2 Constraint preservation

Let $G_{p,t} \in \mathcal{M}_p$ be the pth of the network's P' gates at iteration t. Projection changes only the update direction, and factor-wise retraction returns $G_{p,t+1} \in \mathcal{M}_p$. Hence every gate remains on its assigned manifold. In the implementation the O-assigned gates are carried by the complex unitary kernel of Eq. (B13), which keeps Eq. (B19) at machine precision at every step; their realness is not separately re-projected and holds to about 10^{-3} in the released bases. For the ordered composition $T(\theta_t) = G_{P',t} \cdots G_{1,t}$,

$$T(\theta_t)^\dagger T(\theta_t) = G_{1,t}^\dagger \cdots G_{P',t}^\dagger G_{P',t} \cdots G_{1,t} = I. \tag{B19}$$

Thus the trained transform remains invertible, and because the gate dimensions and wiring stay fixed, training changes neither the parameter count nor the contraction cost of Appendix A.3.

B.3 Batched evaluation

The transform is never materialized as a dense $2^{m+n} \times 2^{m+n}$ matrix. Instead, $T(\theta)\,\mathbf{x}$ is evaluated on the binary-leg image tensor of Section 3, with every gate G_p kept as a separate tensor in the contraction, at the gate-by-gate cost of Appendix A.3. The gates are applied in their circuit order, which the topology fixes. This order is not searched and in particular is not delegated to JAX's default `einsum` path policy. Each gate is an axis-wise contraction against the image tensor: controlled rotations act on the slice their control selects, and diagonal gates enter as broadcast multiplications on the labels of their control and target legs, introducing no new legs. The fixed sequence is compiled once per topology with JAX (Bradbury et al. 2018) and cached. The adjoint $T(\theta)^\dagger$ reuses the same network with conjugated gate tensors and reversed gate legs, so the forward and inverse transforms share one parameter set.

A minibatch adds one batch leg. The per-image loss $\mathcal{L}_k$ of Eq. (1) is vectorized over the leading batch axis with the gate tensors broadcast across it, and the training objective is the batch mean. Reverse-mode differentiation through the compiled contraction (Liao et al. 2019; Roa-Villescas et al. 2024) therefore returns one accumulated ambient gradient per gate, which, once conjugated, is the g_t consumed by the update of Appendix B.1. Keeping every gate as a separate tensor in the contraction preserves its independent manifold constraint; gates that share a factor manifold are grouped and updated by one stacked projection–retraction kernel.

Appendix C Training robustness

This appendix assesses the robustness of the training procedure for the $N = 2^8$ QFT on the DIV2K dataset, using the gate numbering of Fig. 9. Appendix C.1 varies the order in which the gates are trained and their initialization, and a seed-robustness study varies the random seed behind the initialization and the batch subsample. Figure 10 collects the results. Test PSNRs in this appendix are scored on the original 50-image test set, as in Appendix D. The full training traces of the unfreeze runs, and the trained basis and endpoint scores of each reseeded run, are included in the benchmark archive (An et al. 2026a).

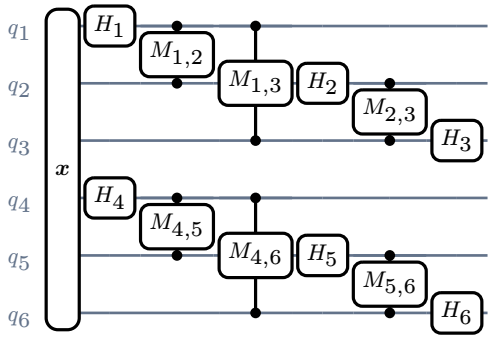

Fig. 9 The QFT network of Fig. 1b with its gates numbered, drawn for QFT(3, 3). The DIV2K runs use the analogous QFT(8, 8) (72 gates). The two registers act on the rows (q_1–q_3) and columns (q_4–q_6) of the input $\mathbf{x}$

C.1 Gate unfreezing

The trained operator is two independent copies of the QFT network drawn in Fig. 1b, one per image axis. Figure 9 draws the same network for

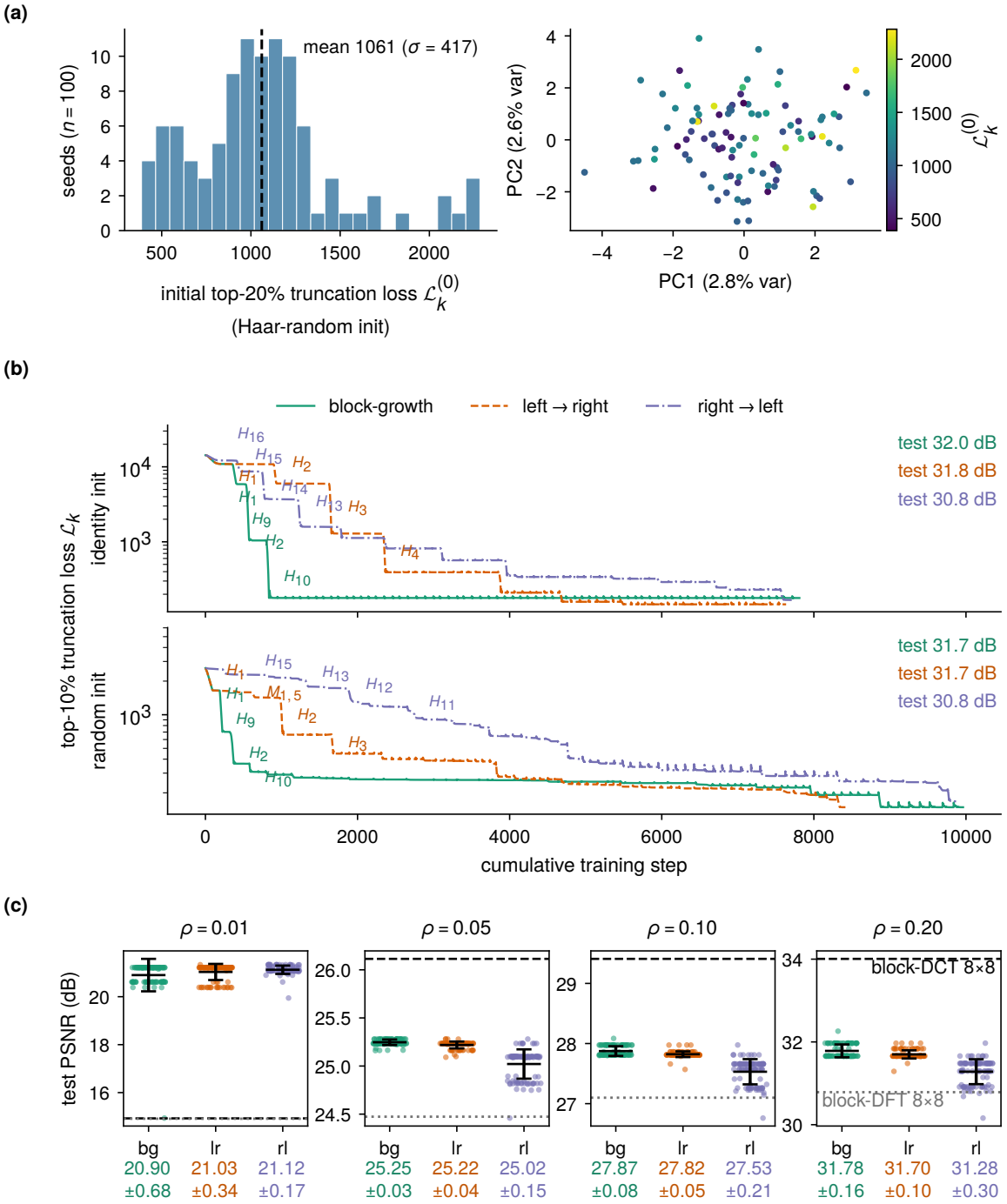

Fig. 10 Robustness of QFT(8, 8) training on DIV2K. (a) 100 Haar-random initializations: initial top-20% truncation loss $\mathcal{L}_k^{(0)}$ (left, mean dashed) and a 2-D principal component analysis (PCA) of the init parameters, colored by $\mathcal{L}_k^{(0)}$ (right). (b) Unfreeze dynamics: top-10% truncation loss $\mathcal{L}_k$ per thaw ordering from identity (top) and Haar-random (bottom) init. Labels mark the gates behind the largest drops, and endpoints give the test PSNR at $\rho = 0.20$. (c) Per-seed test PSNR for the block-growth (bg), left-to-right (lr) and right-to-left (rl) orderings, 100 Haar-random initializations each under the top-20% objective, with mean $\pm\,\sigma$ as error bars and numbers. Dashed and dotted lines are the 8×8 block DCT and DFT. At $\rho = 0.20$, 297/300 runs beat the block DFT (worst seed 30.16 dB)

three qubits per axis, with its gates numbered. H_i is the Hadamard-role $U(2)$ gate on qubit i and $M_{i,j}$ $(i < j)$ the controlled phase coupling qubits i and j within an axis, 72 gates in total at $m = n = 8$. Wires are numbered q_1 to q_{16} across the two registers, matching the gate numbering of the released runs rather than the per-register bit labels $q_0, \ldots, q_{n-1}$ of Appendix A, with the axis-1 register carrying q_9 to q_{16}.

We thaw the gates one at a time: each stage thaws one gate and trains all unfrozen gates to a plateau (Riemannian grad-norm $\|g\| < 10^{-5}$ or $|\Delta\mathcal{L}_k| < 10^{-5}$, minimum 5 steps, 800-step cap) on a fixed 50-image batch under the top-10% truncation loss $\mathcal{L}_k$, with the Adam moments reset between stages. We compare three thaw orderings and two initializations: the QFT-family identity initialization $T(\theta_{\mathrm{id}})\,\mathbf{x} = \mathbf{x}$, and Haar-random, which draws each Hadamard-role gate from the Haar distribution on $U(2)$, the unique unitarily invariant (uniform) measure on the group (Mezzadri 2007), and each controlled phase uniformly, with the seed shared across orderings.

Written on these gates, the three orderings are:

- *Block-growth* (`bg`): at stage s, thaw, on each axis in turn, qubit s and then its couplings to earlier qubits, gate by gate, growing a square $2^s \times 2^s$ transform; long-range couplings thaw only in the late stages.
- *Left-to-right* (`lr`): the QFT construction order (each qubit's Hadamard-role gate, then its couplings), axis 0 before axis 1.
- *Right-to-left* (`rl`): the exact reverse.

The `bg` partner of H_1 is therefore H_9, which is why H_9 is among the earliest large drops in Fig. 10b.

Figure 10b shows the resulting staircase of the truncation loss $\mathcal{L}_k$. The ordering changes the trajectory, `bg` collapsing within a few stages and `rl` descending late in long steps, but all six order × init combinations end within 1.2 dB of one another, with the two `rl` runs trailing by ∼1 dB, and every stage terminates on the loss-Δ plateau rather than the step cap. Training loss and test PSNR order the runs differently, since the loss is a top-10% proxy on the training batch while the PSNR is a 20%-keep score on the held-out set. The identity-init `bg` run, for instance, ends highest in training loss yet best in test PSNR.

Appendix D Initialization noise

The controlled DCT-IV circuit of Appendix A.2 is initialized at the exact analytic transform, and Table 2 reports its relaxed optimum after training. A disturbance sweep tests whether progressively broader local perturbations destroy the useful structure of the untrained transform, and whether training recovers from the perturbed starts. Starting from the exact init, we select a random fraction f of the DCT-IV's stored gate entries (counted below), apply on-manifold Gaussian perturbations of fixed scale $\sigma_{\mathrm{jit}} = 0.1$, and retrain for 1010 steps (101 epochs of ten 50-image minibatches) of the top-10% truncation loss $\mathcal{L}_k$ on the 500-image DIV2K-8q pool, scoring test PSNR at four keep ratios ρ on the original 50-image test set (the first half of the expanded set behind Table 2, so PSNRs differ slightly from its means). Ten disturbance rates from $f = 0.1\%$ (2 selected entries) to 100% (all of them), three seeds each, vary the *breadth* of an on-manifold disturbance at the fixed local scale σ_{jit}; robustness to larger-magnitude disturbances is not probed.

On-manifold perturbation.

Evaluating Eq. (A10) at $n = 8$ for the two registers gives 214 gates: 112 two-leg X gates and 102 structured single-qubit gates. The implementation stores the X gates as 4×4 arrays and the others as 2×2 arrays, giving $112 \cdot 16 + 102 \cdot 4 = 2200$ real-valued entries at exact initialization. The jitter treats these entries as one flat vector of length $E = 2200$ but projects each affected gate separately, so each perturbed start remains a valid unitary transform with the DCT-IV wiring. It is no longer the exact DCT-IV. We draw a uniform subset $S \subseteq \{1, \ldots, E\}$ with $|S| = \lfloor fE \rceil$. For each gate G it touches, we add Gaussian noise to the selected entries and re-project onto the gate's manifold,

$$\hat{G} = G + \sigma_{\mathrm{jit}}\,(\Xi \odot B_G), \qquad G' = \mathrm{polar}(\hat{G}), \quad \text{(D20)}$$

where $\Xi_{ij} \sim \mathcal{N}(0,1)$, B_G masks G's selected entries, and $\mathrm{polar}(\hat{G})$, the orthogonal factor of the polar decomposition, is the nearest orthogonal matrix to $\hat{G}$ (the controlled sign gate, stored as the phase stub $D(0,0,0,\pi)$ of Table 3, is jittered in its phase instead). A gate moves only if one of its entries is selected. Because the polar projection

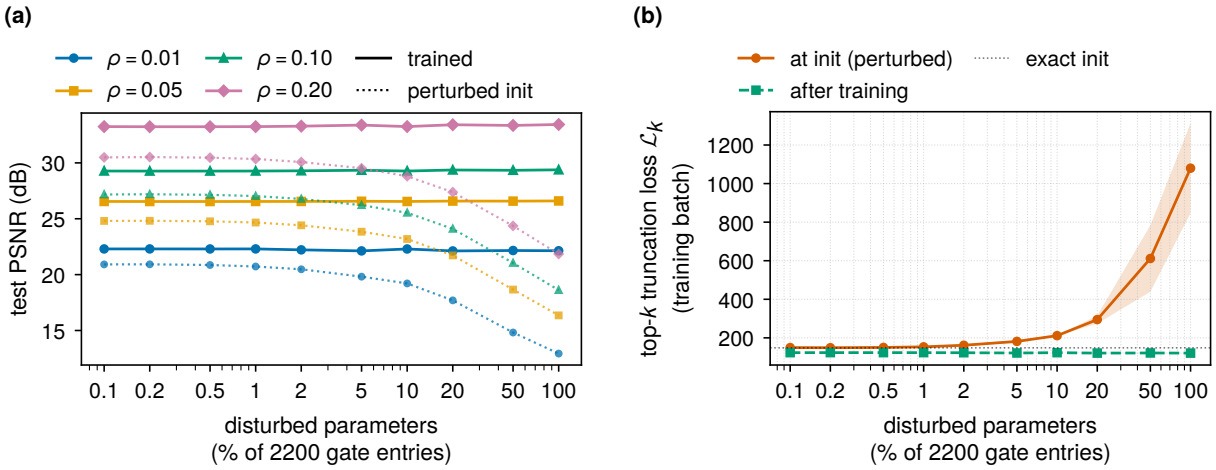

Fig. 11 Disturbance sweep of the exact DCT-IV initialization on DIV2K-8q at noise scale $\sigma_{\text{jit}} = 0.1$, against the disturbed fraction f of the 2200 gate entries (means over three seeds). (a) The perturbed-init PSNR (dotted) falls with f, while the trained PSNR (solid) stays within ∼0.2 dB of the undisturbed reference. (b) Training-batch loss at the perturbed init and after 1010 steps, with the exact-init loss dotted and ±1 standard deviation shaded

acts on the whole gate, f specifies how broadly gates are seeded with noise rather than the final fraction of entries that differ.

Training recovers the tested perturbations. For the tested noise scale and seeds, the seed-averaged trained endpoint is insensitive to f (solid lines in Fig. 11a), staying within ∼0.2 dB of the undisturbed exact-init endpoint (33.25 dB at $\rho = 0.20$) at every keep ratio, even though the perturbed-init PSNR falls by up to 8.6 dB across the sweep (dotted lines).

The same recovery shows up in the training loss (Fig. 11b). The perturbed init's truncation loss $\mathcal{L}_k$ on a fixed 50-image training batch, the value the optimizer sees at step 0, rises from 148 at the exact init to 1080 at $f = 100\%$. After 1010 steps every perturbed init returns to essentially the exact-init training loss (121 at $f = 100\%$ versus 123 at $f = 0$).

References

Absil PA, Mahony R, Sepulchre R (2008) Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ, https://doi.org/10.1515/9781400830244

Agustsson E, Timofte R (2017) 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), pp 1122–1131, https://doi.org/10.1109/CVPRW.2017.150

Ahmed N, Natarajan T, Rao KR (1974) Discrete cosine transform. IEEE Trans Comput C-23(1):90–93. https://doi.org/10.1109/T-C.1974.223784

An S, Slavakis K (2026) Quantum-inspired trainable and parameter-efficient tensor networks for image inpainting. arXiv:2609.17298

An S, Ni Z, Zhou H, et al (2026a) pdft-benchmarks: reproduction suite for Fast Trainable Multilinear Bases for Image Compression. Zenodo, software and data, version 0.1.0, https://doi.org/10.5281/zenodo.23297380

An S, Ni Z, Zhou H, et al (2026b) pdft: learning parametric quantum Fourier transforms via manifold optimization. Zenodo, software, version 0.3.0, https://doi.org/10.5281/zenodo.23296618

Ballé J, Laparra V, Simoncelli EP (2017) End-to-end optimized image compression. In: International Conference on Learning Representations (ICLR)

Bécigneul G, Ganea OE (2019) Riemannian adaptive optimization methods. In: International Conference on Learning Representations (ICLR)

Benedetti M, Lloyd E, Sack S, et al (2019) Parameterized quantum circuits as machine learning models. Quantum Sci Technol 4(4):043001. https://doi.org/10.1088/2058-9565/ab4eb5

Bengio Y, Léonard N, Courville A (2013) Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv:1308.3432

Bittel L, Kliesch M (2021) Training variational quantum algorithms is NP-hard. Phys Rev Lett 127(12):120502. https://doi.org/10.1103/PhysRevLett.127.120502

Bradbury J, Frostig R, Hawkins P, et al (2018) JAX: composable transformations of Python+NumPy programs. http://github.com/jax-ml/jax

Britanak V, Yip PC, Rao KR (2007) Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations. Academic Press, Amsterdam, https://doi.org/10.1016/B978-0-12-373624-6.X5000-0

Cerezo M, Arrasmith A, Babbush R, et al (2021a) Variational quantum algorithms. Nat Rev Phys 3(9):625–644. https://doi.org/10.1038/s42254-021-00348-9

Cerezo M, Sone A, Volkoff T, et al (2021b) Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat Commun 12(1):1791. https://doi.org/10.1038/s41467-021-21728-w

Chen WH, Smith CH, Fralick SC (1977) A fast computational algorithm for the discrete cosine transform. IEEE Trans Commun 25(9):1004–1009. https://doi.org/10.1109/TCOM.1977.1093941

Cooley JW, Tukey JW (1965) An algorithm for the machine calculation of complex Fourier series. Math Comput 19(90):297–301. https://doi.org/10.1090/S0025-5718-1965-0178586-1

Coppersmith D (1994) An approximate Fourier transform useful in quantum factoring. Tech. Rep. RC 19642, IBM Research, arXiv:quant-ph/0201067

Donoho DL (1995) De-noising by soft-thresholding. IEEE Trans Inf Theory 41(3):613–627. https://doi.org/10.1109/18.382009

Edelman A, Arias TA, Smith ST (1998) The geometry of algorithms with orthogonality constraints. SIAM J Matrix Anal Appl 20(2):303–353. https://doi.org/10.1137/S0895479895290954

Evenbly G, White SR (2018) Representation and design of wavelets using unitary circuits. Phys Rev A 97(5):052314. https://doi.org/10.1103/PhysRevA.97.052314

Feig E, Winograd S (1992) Fast algorithms for the discrete cosine transform. IEEE Trans Signal Process 40(9):2174–2193. https://doi.org/10.1109/78.157218

Google (2017) Quick, Draw! Dataset. https://quickdraw.withgoogle.com/data

Goyal VK (2001) Theoretical foundations of transform coding. IEEE Signal Process Mag 18(5):9–21. https://doi.org/10.1109/79.952802

Ha D, Eck D (2018) A neural representation of sketch drawings. In: International Conference on Learning Representations (ICLR)

Han ZY, Wang J, Fan H, et al (2018) Unsupervised generative modeling using matrix product states. Phys Rev X 8(3):031012. https://doi.org/10.1103/PhysRevX.8.031012

Hauru M, Van Damme M, Haegeman J (2021) Riemannian optimization of isometric tensor networks. SciPost Phys 10(2):040. https://doi.org/10.21468/SciPostPhys.10.2.040

Havlíček V, Córcoles AD, Temme K, et al (2019) Supervised learning with quantum-enhanced feature spaces. Nature 567(7747):209–212. https://doi.org/10.1038/s41586-019-0980-2

Hillar CJ, Lim LH (2013) Most tensor problems are NP-hard. J ACM 60(6):1–39. https://doi.org/10.1145/2512329

Jain AK (1979) A sinusoidal family of unitary transforms. IEEE Trans Pattern Anal Mach Intell PAMI-1(4):356–365. https://doi.org/10.1109/TPAMI.1979.4766944

Jang E, Gu S, Poole B (2017) Categorical reparameterization with Gumbel-Softmax. In: International Conference on Learning Representations (ICLR)

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR)

Li Z, Kovachki N, Azizzadenesheli K, et al (2021) Fourier neural operator for parametric partial differential equations. In: International Conference on Learning Representations (ICLR)

Liao HJ, Liu JG, Wang L, et al (2019) Differentiable programming tensor networks. Phys Rev X 9(3):031041. https://doi.org/10.1103/PhysRevX.9.031041

Liu JG, Wang L, Zhang P (2021) Tropical tensor network for ground states of spin glasses. Phys Rev Lett 126(9):090506. https://doi.org/10.1103/PhysRevLett.126.090506

Liu JG, Gao X, Cain M, et al (2023) Computing solution space properties of combinatorial optimization problems via generic tensor networks. SIAM J Sci Comput 45(3):A1239–A1270. https://doi.org/10.1137/22M1501787

Loeffler C, Ligtenberg A, Moschytz GS (1989) Practical fast 1-D DCT algorithms with 11 multiplications. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp 988–991, https://doi.org/10.1109/ICASSP.1989.266596

Mallat S (2009) A Wavelet Tour of Signal Processing: The Sparse Way, 3rd edn. Academic Press, Amsterdam, https://doi.org/10.1016/B978-0-12-374370-1.X0001-8

Markov IL, Shi Y (2008) Simulating quantum computation by contracting tensor networks. SIAM J Comput 38(3):963–981. https://doi.org/10.1137/050644756

McClean JR, Romero J, Babbush R, et al (2016) The theory of variational hybrid quantum-classical algorithms. New J Phys 18(2):023023. https://doi.org/10.1088/1367-2630/18/2/023023

McClean JR, Boixo S, Smelyanskiy VN, et al (2018) Barren plateaus in quantum neural network training landscapes. Nat Commun 9(1):4812. https://doi.org/10.1038/s41467-018-07090-4

McCord JC, Evenbly G (2022) Improved wavelets for image compression from unitary circuits. arXiv:2203.02556

Mezzadri F (2007) How to generate random matrices from the classical compact groups. Not Am Math Soc 54(5):592–604. arXiv:math-ph/0609050

Oseledets IV (2011) Tensor-train decomposition. SIAM J Sci Comput 33(5):2295–2317. https://doi.org/10.1137/090752286

Plonka G, Tasche M (2005) Fast and numerically stable algorithms for discrete cosine transforms. Linear Algebra Appl 394:309–345. https://doi.org/10.1016/j.laa.2004.07.015

Püschel M, Moura JMF (2008) Algebraic signal processing theory: Cooley–Tukey type algorithms for DCTs and DSTs. IEEE Trans Signal Process 56(4):1502–1521. https://doi.org/10.1109/TSP.2007.907919

Roa-Villescas M, Gao X, Stuijk S, et al (2024) Probabilistic inference in the era of tensor networks and differential programming. Phys Rev Res 6(3):033261. https://doi.org/10.1103/PhysRevResearch.6.033261

Schuld M, Killoran N (2019) Quantum machine learning in feature Hilbert spaces. Phys Rev Lett 122(4):040504. https://doi.org/10.1103/PhysRevLett.122.040504

Stoudenmire EM, Schwab DJ (2016) Supervised learning with tensor networks. In: Advances in Neural Information Processing Systems, pp 4799–4807

Sullivan GJ, Ohm JR, Han WJ, et al (2012) Overview of the High Efficiency Video Coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22(12):1649–1668. https://doi.org/10.1109/TCSVT.2012.2221191

Theis L, Shi W, Cunningham A, et al (2017) Lossy image compression with compressive autoencoders. In: International Conference on Learning Representations (ICLR)

Vidal G (2004) Efficient simulation of one-dimensional quantum many-body systems. Phys Rev Lett 93(4):040502. https://doi.org/10.1103/PhysRevLett.93.040502

Vidal G (2007) Entanglement renormalization. Phys Rev Lett 99(22):220405. https://doi.org/10.1103/PhysRevLett.99.220405

Wallace GK (1992) The JPEG still picture compression standard. IEEE Trans Consum Electron 38(1):xviii–xxxiv. https://doi.org/10.1109/30.125072

Wang Z, Bovik AC, Sheikh HR, et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612. https://doi.org/10.1109/TIP.2003.819861

Xie SM, Ermon S (2019) Reparameterizable subset sampling via continuous relaxations. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI), pp 3919–3925, https://doi.org/10.24963/ijcai.2019/544