DEPTH IS DEGREE: TEST-TIME-TRAINING UPDATES ARE POLYNOMIAL FILTERS, AND LEARNED STEP SIZES ONLY CHOOSE THE COEFFICIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We replace the learned constants of test-time-training updates — step sizes in TTT, momenta in Titans, conjugate-gradient coefficients in MesaNet, the single step of linear attention — and of unrolled restoration solvers with a polynomial head that computes them from the operator: it starts at the textbook solver, costs the same operator applications, and learns no solver constant. The head rests on one fact: each such update is a solver for a quadratic in-context loss, and we prove it is a polynomial filter in its operator whose degree is the number of operator applications, so learned step sizes and momenta choose a filter but cannot enlarge the class; the head computes the class-optimal one. The regime is one protocol on one GPU (42M and 125M parameters), where the ordering does not change with size. On multi-query associative recall the head ties learned step sizes, momentum, and conjugate gradient at the ten-seed median ($86/57$ at loads $32/64$); the tie is the claim, and the head is the only rule with no failed seed. Inside language models it is the lowest-perplexity update at degree four, by 1.8 over learned step sizes at 125M; on in-context regression with a dominant nuisance mode — a high-pass target — the exact solve and adaptive conjugate gradient fail at $2.7\times$ the head's error, which is the class optimum. The theorem is also a design rule: capacity enters only through the nonlinearities and re-estimated operators between solver steps — attention is one — and the solver constants choose within a class the head spans directly, so they can be computed rather than learned. In one unrolled solver network whose operator is re-estimated in every layer (image restoration) the bounded head ties the textbook solver with no rate to choose and no failed seed, while learned step sizes collapse at three of four solver rates and at the fourth land above the exact solve of their own system — a gain the theorem assigns to the re-estimated operator, not to the constants; the "each layer is an iteration" reading holds at initialization, and training replaces it with a measured spectral response. The rule is one line: learn the operator, compute the solver.

1 INTRODUCTION

A test-time-training layer makes its hidden state a model and its update rule a step of learning (Sun et al., 2024); that step is a solver: the hidden state of TTT-Linear is updated by gradient descent on the in-context key–value loss, of Titans by gradient descent with momentum (Behrouz et al., 2024), of MesaNet by conjugate gradient (von Oswald et al., 2026); multi-layer linear attention implements preconditioned gradient descent in context (Ahn et al., 2023), and key–value-binding TTT layers have just been shown to be learned linear attention at any depth (Liu et al., 2026a). The recipe is the one unrolled restoration networks were built from (Gregor & LeCun, 2010; Monga et al., 2021): fix the number of iterations of conjugate gradient, gradient descent, or ADMM, make each iteration a layer, and learn the constants the solver would otherwise set by hand — step sizes, momenta, expansion coefficients, thresholds — end to end on the task; in graph signal processing it produced lightweight alternatives to transformers for restoration (Zeng et al., 2019; Vu et al., 2021; Do et al., 2024; Hosseini et al., 2024; Cai et al., 2026; Yao et al., 2026b), with a graph learner in the role of attention. We ask what the further steps and their learned constants buy.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

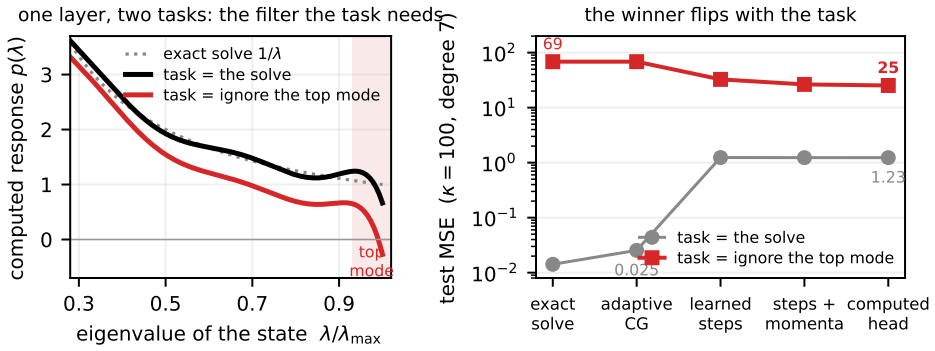


Figure 1: **The task, not the constants, decides the filter.** Left: the computed response of one update (degree 7) on its state's spectrum — a damped solve when the task is the in-context solve, a high-pass that notches the top mode when a dominant mode must be ignored. Right: the same five update rules on the two tasks (κ=100, medians of three seeds): when the task is the solve, adaptive CG wins and every fixed-constant rule ties; when it is not, solving is the failure and the computed head is the class optimum (Tables 3, 4).

One observation. In each of these networks the part that touches the operator — the solver — is a *polynomial filter*. A learned step size multiplies a vector already in a Krylov subspace, a learned momentum recombines two such vectors, and only an application of the operator raises the degree. So for conjugate gradient, gradient and heavy-ball descent, and Chebyshev and Lanczos iterations alike, every setting of the learned constants realizes

$$\mathbf{x}_T = p_\theta(\mathbf{A})\,\mathbf{b}, \qquad \deg p_\theta \le T,$$

a polynomial in the system operator whose degree is the number of operator applications: *depth is degree* (Theorem 1). The containment is classical (Saad, 2003; Greenbaum, 1997); what it says about *learning* is not. The constants choose a filter inside a class they cannot enlarge, and reach only a measure-zero, rank-deficient part of it: of the 29 constants of a 15-step unrolled conjugate gradient, 12 directions reach the output (Proposition 1), and learned step sizes reach only real-rooted polynomials, 42 dB short of a band response (Corollary 1, E4). In splitting methods (ADMM, ISTA) capacity enters through the nonlinearities and re-estimated operators, never the solver constants (Theorem 2).

One consequence. If the solver block is a polynomial filter, parameterize the filter. We replace the unrolled solver by a **closed-form polynomial head**: a filter of the same degree in the same operator, written in a Chebyshev basis in which it is *linear in its coefficients*, so that for a given network its class-optimal coefficients are one $(d{+}1)$-dimensional least-squares solve (Proposition 5). The head starts at the textbook solver's polynomial, costs the same d operator applications, has no step size to diverge, and gives the block the interpretability the unrolling promised: the network *is* this spectral response on this operator (Figure 5a); a gradient-trained variant is the ablation. The identification also predicts what fails: the gradient-trained head diverges where the coefficient scale explodes, and any update that targets the *solve* collapses under interference unless the operator is regularized for the task (Proposition 4). It explains three reported anomalies: more inner-loop steps at inference hurt (Liu et al., 2026a) because the step count is the degree; gradient *ascent* in the inner loop still trains (Liu et al., 2026a) because a step's sign is a coefficient the output projection absorbs; and mini-batch TTT beats its batch form (Sun et al., 2024) because distinct operators, not more steps, leave the class. And it settles a live design question: MesaNet's conjugate gradient, Titans' momentum and TTT's steps are one class, and the adaptive solve's edge exists only where the in-context system is hard.

What the numbers say. Table 1 and Figure 1 are the paper. On multi-query associative recall the closed-form head, with *zero* solver constants and no learning rate, ties learned step sizes, momentum and conjugate gradient at the ten-seed median — the tie is the claim: the constants were buying nothing — and is the only rule with no failed seeds: its worst run is $76/52$ at loads $32/64$ (Table 1).

Table 1: Headline numbers (restoration in Table 2). One network, one budget, one number of operator applications per row; columns differ only in how the coefficients are chosen. Constants: learned solver constants per block; rate grid: runs spent choosing a learning rate (Figure 2, left). The gradient-trained head (E7 $41/13$, E6 1.17) is the ablation between learned and computed coefficients. E7: momentum and exact over three seeds, the rest ten.

setting	metric	learned steps / momenta	**head**	textbook CG	exact
MQAR recall, load $32/64$ (%)	10-seed median	$87/57$ / $87/56$	$\mathbf{86/57}$	$87/56$	$71/48$
worst seed		$59/32$ / $70/46$	$\mathbf{76/52}$	$66/42$	$59/33$
constants; rate grid		4; 5 runs / 8; 5 runs	$\mathbf{0}$; $\mathbf{0}$	0; 0	0; 0
in-context regression, $\kappa{=}100$, $L{=}16$	loss	0.13–0.15	$\mathbf{0.125}$	0.014	0.014

The design rule. Learn the operator, compute the solver. The solver constants cannot add capacity and are the least reliable to train; a small algorithm replaces them at every load and budget we measured: where the target is low-pass the head costs nothing (Section 4.4); where the response must leave the solver's shape, the unrolling cannot follow and the head can (Section 4.3); where the task *is* the solve at large condition number, no fixed polynomial matches adaptive conjugate gradient, and where it is *not* the solve, solving is the failure (Section 4.1). In no regime are learned solver constants the right choice.

Contributions. (1) The closed-form polynomial head, a drop-in for the solver block of unrolled networks and the state update of test-time-training layers, its optimum one least-squares solve. (2) Depth is degree: a uniform theorem that the linear-solver portion of every such scheme is a polynomial filter of degree the number of operator applications, with its reachable set, a localization theorem for splitting methods, and the optimality dilemma. (3) Experiments on in-context regression, recall, language models and one unrolled solver network with a re-estimated operator, under one protocol.

Scope. The head replaces the solver block; the backbone is unchanged, and comparisons are between arms of one network. Learned *matrices* (Gregor & LeCun, 2010; Chen et al., 2018), *thresholds*, and *operators* leave the class (Theorem 2(ii)); the claim is scoped to learned *constants* in the *linear* portions of a scheme, which buy conditioning at best and dead parameters at worst, never capacity. In state-space terms, an input-dependent transition (selectivity) is a re-estimated operator, where Theorem 2 puts the capacity; step sizes and momenta are polynomial coefficients, where it puts none.

2 Related Work

Polynomial filters and unrolling. The founding observation of graph signal processing is that polynomials in a graph operator are spectral filters (Hammond et al., 2011; Shuman et al., 2013), and learning their coefficients in a Chebyshev (Defferrard et al., 2016) or Bernstein (He et al., 2021) basis is standard in graph learning; BernNet's box constraint is the ancestor of Proposition 5(iii). Unrolled solvers with learned constants parameterize the same class indirectly, incompletely, and with worse conditioning. LISTA and its analyses (Gregor & LeCun, 2010; Chen et al., 2018; Liu et al., 2019; Monga et al., 2021) showed that learned *matrices* and *thresholds* accelerate ISTA; our theorems isolate the constants of the linear solver and say that those, unlike matrices and thresholds, do not change the class. Readings of graph neural networks as unrolled energy minimizers (Ma et al., 2021; Zhu et al., 2021; Yang et al., 2021) go from optimization to architecture; we go the other way.

The graph-restoration line. The correspondence between linear denoisers, graph filters, and MAP estimates under a Laplacian prior (Milanfar, 2013; Chan, 2019) underlies the interpretability claims of the networks we test (Zeng et al., 2019; Vu et al., 2021; Do et al., 2024; Hosseini et al., 2024; Cai et al., 2026; Yao et al., 2026b). Recent unrollings learn the constants of BiCG inside Douglas–Rachford layers (Zhang et al., 2025), of CG inside proximal-gradient layers (Do et al., 2026), a Lanczos cutoff on signed graphs (Yao et al., 2026a), and relaxations of the Lanczos recurrence that, as its authors note, keep the Krylov space (Eftekhar et al., 2026); one keeps

textbook CG and learns only the prior and the graph (Wei et al., 2025), the design our results recommend. A recent letter (Hosseini, 2026) proved the nested-composition instance of Corollary 4 for one construction; we generalize the collapse to every linear Krylov iteration, add localization, optimality, and the closed-form head, and test inside one unrolled graph network and on sequence layers. The Krylov ingredients are classical (Golub & Van Loan, 2013; Saad, 2003); the contribution is accounting for which parameters can leave the class.

Sequence layers. The identification is being approached from several sides at once. Liu et al. (2026a) show that TTT layers trained on the *key–value binding* objective $-\langle f(\mathbf{k}), \mathbf{v}\rangle$ — LaCT, ViTTT, Titans — remain a learned linear attention operator for any number of steps, with momentum or Muon: that objective is linear in the inner model, so its gradient never touches the state and every step stays at degree one. Theorem 1 is the statement for the squared reconstruction objective, where the gradient applies the context covariance and depth becomes degree; their result is its degree-one case. For test-time updates of a pretrained model's *weights*, Wakayama (2026) reads the step count as a spectral filter matched to the prompt's signal-to-noise ratio; here the filter is exact and we compute its class optimum. Preconditioned updates are the proposed improvement to the delta rule (Tumma et al., 2026), and preconditioning is the operation that leaves the class (Theorem 2(ii)); softmax transformers can implement preconditioned Richardson iteration (Yan et al., 2026); learned matrix-polynomial recurrences are studied for linear algebra itself (Liu et al., 2026b); MesaNet (von Oswald et al., 2026) solves the in-context problem by CG. None states the multi-step class, its reachable set, or the closed-form member.

3 SOLVERS ARE FILTERS

Setting. In a test-time-training layer a solver for the in-context least squares updates the state $\mathbf{W}$: the operator is the key covariance $\mathbf{A} = \mathbf{K}^\top\mathbf{K}$ (ridge-damped) and the input the key–value correlation $\mathbf{b} = \mathbf{K}^\top\mathbf{v}$; in an unrolled restoration network $\mathbf{A}$ is the learned graph operator and $\mathbf{b}$ the observation. Throughout, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a symmetric operator available through matrix–vector products (symmetry is dropped in Proposition 3; the dilemma needs $\mathbf{A} \succ 0$, the head a bound $\mathrm{spec}(\mathbf{A}) \subset [0, \lambda_{\max}]$; in-context covariances and ridge-damped systems satisfy all three, row-normalized attention only the first), $\mathbf{b} \in \mathbb{R}^N$ an input, $\mathcal{K}_T(\mathbf{A}, \mathbf{b}) = \mathrm{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \ldots, \mathbf{A}^{T-1}\mathbf{b}\}$, and $\mathcal{P}_d$ the polynomials of degree at most d; we call $p(\mathbf{A})\mathbf{b}$ a *polynomial filter* with response $p(\lambda)$ on $\mathrm{spec}(\mathbf{A})$ (Shuman et al., 2013; Hammond et al., 2011). An unrolled network is $\Phi_\theta = N_B \circ S_B \circ \cdots \circ N_1 \circ S_1$ (1), each block S_b running T_b steps of a linear solver on an operator $\mathbf{A}_b$ (fixed, or re-estimated from the current iterate, $\mathbf{A}_b = \mathbf{A}(x_{b-1}; \theta_{\mathrm{op}})$) and each N_b a pointwise map. The parameters form three groups: the solver constants θ_{lin} (step sizes, momenta, recurrence coefficients), the pointwise maps θ_{nl} (thresholds, shrinkage), and the operator estimator θ_{op} (the graph learner). A block S_b is a *linear Krylov iteration with constant coefficients* if it keeps a finite state of vectors initialized in $\mathrm{span}\{\mathbf{b}\}$ and updates it as $\mathbf{s}_{t+1} = \Phi_t(\mathbf{s}_t, \mathbf{A}\mathbf{s}_t; \theta_{\mathrm{lin}})$ (2), with Φ_t linear in its vector arguments, coefficients independent of $\mathbf{b}$, and at most one application of $\mathbf{A}$ per state vector per step: gradient descent, heavy-ball and Nesterov momentum, conjugate gradient with learned (α_t, β_t), Chebyshev iteration with learned coefficients, and the unrolled solvers of the graph-restoration line (Hosseini et al., 2024; Do et al., 2024; Cai et al., 2026). Textbook conjugate gradient is *not* (its steps depend on $\mathbf{b}$; Section 3.3). Proofs are in Appendix D.

3.1 DEPTH IS DEGREE

Theorem 1 (Uniform Krylov collapse)**.** *Let S be a linear Krylov iteration with constant coefficients run for T steps on $(\mathbf{A}, \mathbf{b})$. Then for every value of θ_{lin},*

$$\mathbf{x}_T \;=\; p_\theta(\mathbf{A})\,\mathbf{b}, \qquad p_\theta \in \mathcal{P}_T, \tag{3}$$

and $\mathbf{b} \mapsto \mathbf{x}_T$ is a fixed polynomial filter, linear in $\mathbf{b}$ and determined by p_θ on $\mathrm{spec}(\mathbf{A})$.

The mechanism: the Krylov index counts operator applications, not parameters. A learned constant multiplies a vector that already lies in the subspace; only $\mathbf{A}$ raises the index; and since the constants do not depend on $\mathbf{b}$, the composition is one fixed linear map. Three consequences follow. *Instances* (Corollary 3 in the appendix): gradient descent with learned steps, heavy-ball, and conjugate gradient with learned (α_t, β_t) reach subsets of $\mathcal{P}_T$; Chebyshev iteration with learned recurrence

coefficients reaches all of it, because it *is* a direct parameterization; from $\mathbf{x}_0 = 0$ the iterate degree is $T-1$, the "matched degree" of Section 4. *Nesting* (Corollary 4): an inner polynomial of degree K applied by an outer solver for m steps from $\mathbf{x}_0 = 0$ gives degree at most $K(m-1)$ for every setting of both coefficient sets. *Lanczos filters* (Remark 1): an m-step Lanczos filter with a learned response (Yao et al., 2026b) lies in $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$, but its polynomial interpolates the response at Ritz values that depend on $\mathbf{b}$ — the subspace statement holds, the fixed-filter statement does not.

Proposition 1 (Reachable set)**.** *Let the T-step block have P learned constants entering equation 2 polynomially. Its reachable set $\mathcal{R} = \{p_\theta\} \subseteq \mathcal{P}_T$ is the image of a polynomial map $\mathbb{R}^P \to \mathbb{R}^{T+1}$, hence semialgebraic with $\dim \mathcal{R} \le \min(P, T+1)$; if $P < T+1$, or the map's Jacobian has rank below $T+1$ everywhere, $\mathcal{R}$ is a measure-zero strict subset of $\mathcal{P}_T$.*

Learned constants thus choose within a class they cannot enlarge, and usually a thin part of it. The nested denoiser of Hosseini et al. (2024) inverts an order-K truncated expansion by m steps of conjugate gradient with learned (α_t, β_t); its published setting $K{=}10$, $m{=}3$ spends seventeen constants on a class of dimension twenty-one and reaches twelve directions at initialization: the last momentum never touches the output, and the outer polynomial absorbs rescalings of the inner one (Example 1; measured in Section 4.4).

3.2 The same object in sequence layers

In in-context linear regression a transformer reads $(\mathbf{x}_i, y_i)_{i \le n}$ and a query and must output $\hat{y} \approx \langle \mathbf{w}^\star, \mathbf{x}_q \rangle$. With the restricted parameterization of Ahn et al. (2023), L layers of linear self-attention implement L steps of preconditioned gradient descent on the in-context least squares,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{A}_k\big(\hat{\mathbf{\Sigma}}\mathbf{w}_k - \mathbf{b}\big), \qquad \hat{\mathbf{\Sigma}} = \tfrac{1}{n}\mathbf{X}^\top\mathbf{X}, \quad \mathbf{b} = \tfrac{1}{n}\mathbf{X}^\top\mathbf{y}, \tag{4}$$

each layer applying the in-context operator $\hat{\mathbf{\Sigma}}$ once.

Proposition 2 (Depth is degree, in context)**.** *If the learned preconditioners are scalars, $\mathbf{A}_k = a_k\mathbf{I}$ (or commute with $\hat{\mathbf{\Sigma}}$), then $\mathbf{w}_L = p_a(\hat{\mathbf{\Sigma}})\,\mathbf{b}$ for every setting of the a_k, with $1 - \lambda\, p_a(\lambda) = \prod_{k=1}^{L}(1 - a_k\lambda)$: exactly the degree-$L$ polynomials q with $q(0) = 1$ and real roots $1/a_k$, a strict subset of $\mathcal{P}_L$ with nonempty interior within $\{q(0) = 1\}$, containing the Chebyshev minimax polynomial (consistent with the $\sqrt{\kappa}$ depth requirement of Gatmiry et al. (2024)). Unrestricted matrices $\mathbf{A}_k$ leave the polynomial class (Theorem 2(ii)).*

Corollary 1 (Class hierarchy)**.** *Write $\mathcal{C}_{\mathrm{GD}}$, $\mathcal{C}_{\mathrm{HB}}$, $\mathcal{C}_{\mathrm{CG}}$ and $\mathcal{C}_{\mathrm{head}}$ for the sets of responses reachable at L operator applications by learned steps, learned steps and momenta, learned CG constants, and the direct head. Then $\mathcal{C}_{\mathrm{GD}} \subsetneq \mathcal{C}_{\mathrm{HB}} \subseteq \mathcal{C}_{\mathrm{head}} = \mathcal{P}_L$: $\mathcal{C}_{\mathrm{GD}}$ is the real-rooted polynomials (in $q = 1 - \lambda p$); $\mathcal{C}_{\mathrm{HB}}$ contains conjugate-pair roots (already at $L{=}2$) and is full-dimensional (verified for $L \le 16$); $\mathcal{C}_{\mathrm{CG}}$ is measure-zero where Proposition 1 applies (Example 1). Proof in Appendix F.*

The restriction is measurable: fitted at degree 20 on the E4 targets, learned steps stop 42 dB short of the head on a band response and lose 5 dB even on the Wiener denoiser, where momenta recover all but 0.1 dB (Figure 4).

The same object sits inside test-time-training layers. TTT (Sun et al., 2024) updates a linear inner model by gradient descent on the in-context squared reconstruction loss; in its *batch* form (all gradients at the same weights; one step is linear attention) L steps are L applications of the context covariance and the memory is a degree-L polynomial in it. Under the linear binding loss $-\langle f(\mathbf{k}), \mathbf{v}\rangle$ the gradient is state-independent and no step raises the degree (Liu et al., 2026a). TTT's *mini-batch* form applies a different operator per mini-batch, a product of non-commuting factors — the re-estimated-operator mechanism of Theorem 2(ii), outside the fixed-polynomial class — and its per-token learning rate $\eta(x_t)$ is a data-dependent constant; the reported gain from smaller mini-batches is not more steps of one solve but more distinct operators. Titans (Behrouz et al., 2024) is the momentum form of the same update; its linear-memory, gate-constant, batch specialization is the heavy-ball instance of Corollary 3, and as built it leaves the class three ways (data-dependent gates, per-token operators, an MLP memory). MesaNet (von Oswald et al., 2026) solves the gated in-context least squares to optimality by conjugate gradient on the vector system $(\mathbf{H}_t + \Lambda)\,\mathbf{q}^* = \mathbf{q}_t$: adaptive constants, a data-dependent polynomial in $\mathbf{H}_t$, and the per-context optimum of Proposition 4. Learned scalars reach a full-dimensional subset of $\{q(0) = 1\}$ here while unrolled conjugate gradient is rank-deficient: the composition structure, not the number of constants, decides what is reachable.

Proposition 3 (Nonsymmetric operators)**.** *Theorem 1 and Proposition 1 hold verbatim for any square* $\mathbf{A}$ *— a directed-graph Laplacian, a row-normalized attention matrix — since their proofs use only linearity, and Proposition 4 holds in the residual norm with the* T*-th GMRES iterate in place of CG's. No frequency formulation for directed graphs is needed to say what an unrolled directed-graph solver computes.*

3.3 Where capacity enters, and what the layers mean

Theorem 2 (Localization)**.** *Let* Φ_θ *be as in equation 1 with every* S_b *a linear Krylov iteration with constant coefficients (*T_b *steps on* $\mathbf{A}_b$*). (i)* ***Ceiling.*** *Every* Φ_θ *factors as* $N_B \circ p_B(\mathbf{A}_B) \circ \cdots \circ N_1 \circ p_1(\mathbf{A}_1)$ *with* $p_b \in \mathcal{P}_{T_b}$*; replacing each* S_b *by a direct parameterization of* $\mathcal{P}_{T_b}$ *yields a class that contains every* Φ_θ*. (ii)* ***Strictness.*** *If some* N_b *is not affine on the range of its input, or some* $\mathbf{A}_b$ *is re-estimated from* x_{b-1} *by a non-affine map, then for generic* θ *the network* Φ_θ *is not a polynomial filter of any degree.*

Capacity enters through the shrinkage and the operator estimator, and nowhere in the solver constants between them. For quadratic priors every N_b is affine and the whole network is one polynomial filter; for sparsity-promoting priors only the shrinkage grows the class — the mechanism by which learned thresholds help in LISTA-type networks (Gregor & LeCun, 2010; Chen et al., 2018).

Proposition 4 (Optimality dilemma)**.** *Let* $\mathbf{A} \succ 0$ *and* $\mathbf{x}_T^{\mathrm{CG}}$ *the* T*-th iterate of textbook conjugate gradient from* $\mathbf{x}_0 = 0$*. For every linear Krylov iteration with constant coefficients from* $\mathbf{x}_0 = 0$ *and every* θ_{lin}*,* $\|\mathbf{x}_T^\theta - \mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}} \geq \|\mathbf{x}_T^{\mathrm{CG}} - \mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ *for every* $\mathbf{b}$*. Hence either* (Horn 1) *the learned constants do not improve the solve, or* (Horn 2) *they improve an end-task loss by producing a* worse *approximation of* $\mathbf{A}^{-1}\mathbf{b}$ *— in which case the layer-as-iteration reading is forfeited and the block is a task-selected polynomial filter.*

This is a re-attribution, not a debunking: learned constants may help, but as coefficients of a filter chosen for the task, and "each layer is an iteration" then has to be replaced by "the network is this spectral response" — available directly, from a larger class, by parameterizing the polynomial.

3.4 The closed-form polynomial head

Proposition 5 (Conditioning)**.** *Let* J_θ *be the Jacobian of* $\theta \mapsto p_\theta|_{\mathrm{spec}(\mathbf{A})}$*. (i) For a composed parameterization,* $\operatorname{rank} J_\theta \leq \dim \mathcal{R}$ *everywhere, and a constant that never reaches the output contributes a zero column. (ii) For a direct parameterization in the Bernstein basis,* $p_c(\lambda) = \sum_{j=0}^{d} c_j \binom{d}{j} \lambda^j (1-\lambda)^{d-j}$ *on* $\mathrm{spec}(\mathbf{A}) \subset [0,1]$ *(or the Chebyshev basis on* $[-1,1]$*),* $c \mapsto p_c$ *is linear and onto* $\mathcal{P}_d$*,* J_c *has full rank* $d{+}1$ *whenever the spectrum has* $d{+}1$ *distinct points, and a squared-error loss is convex in* c *(Farouki & Rajan, 1987). (iii) If* $c_j \in [0, \gamma]$ *then* $\|p_c(\mathbf{A})\|_2 \leq \gamma$ *and* $p_c(\mathbf{A}) \succeq 0$*: a certificate on demand;* $c_j = j/d$ *is the identity response exactly.*

Closed-form polynomial head. Replace the T-step solver block on $(\mathbf{A}, \mathbf{b})$ by

$$\mathbf{x} \;=\; \mathbf{x}_0 + \sum_{k=0}^{d} a_k \, T_k(\hat{\mathbf{A}})\, \mathbf{r}, \qquad \hat{\mathbf{A}} = 2\mathbf{A}/\lambda_{\max} - \mathbf{I}, \quad \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad d = T - 1,$$

with T_k the Chebyshev polynomials (a three-term recurrence; d operator applications, the same as the solver it replaces). The output is linear in a, so for a fixed network its class optimum under the squared loss is

$$a^\star \;=\; \left(\mathbb{E}\, \mathbf{V}^\top \mathbf{V}\right)^{-1} \mathbb{E}\, \mathbf{V}^\top (\mathbf{x}^* - \mathbf{x}_0), \qquad \mathbf{V} = [T_k(\hat{\mathbf{A}})\mathbf{r}]_{k=0}^{d},$$

a $(d{+}1)$-dimensional solve. Initialize a at the textbook solver's polynomial (the T-step CG polynomial on a calibration batch); during training, take a gradient step on the network and reset a to the solve with expectations kept as moving averages (Appendix B). The gradient-trained variant (Bernstein basis, $c = c^\star + \delta$) is the ablation between the two.

Which coordinates. The Jacobian favours the additive basis; the *size* of the coefficients does not: a solve-like response has additive coefficients growing like $\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k$ with the degree while the step form keeps every step in $[1/\lambda_{\max}, 1/\lambda_{\min}]$ (Appendix E); only the additive coordinates make the output linear in the parameters, so only there is the optimum a solve.

4 EXPERIMENTS

All experiments run on our matrix-free graph-solver package, a matrix-free PyTorch implementation of grid-graph operators and solvers (anonymized; supplied as supplementary material), so every arm shares one operator and differs only in the solver block. Recall results are medians over ten seeds (three where stated), regression and language-model results over three and one, restoration rows one seed unless a spread is given; protocols and per-seed curves are in Appendix I. **Models**: the transformer instance of Proposition 2 on in-context regression (E6); a causal recall layer with the in-context least-squares state (E7) and 42M/125M language models built from it (Appendix G.7); a fixed symmetric operator with prescribed responses (E4); and, for the regime in which the operator is re-estimated in every layer (the ℓ_2 end of Theorem 2), one unrolled solver network we control end to end — image restoration, a learned nonlocal operator rebuilt in each of three layers, CG with $d{=}10$ operator applications per layer, denoising at $\sigma{=}25$ and non-blind deblurring under one protocol (E2) — with GDD (Hosseini et al., 2024) as a second instance; the published networks of that line (Zeng et al., 2019; Do et al., 2024; Cai et al., 2026) are not retrained. **Arms**: the linear solve inside every layer is instantiated at a *matched degree* d as UNROLLED (the published learned-constant CG), CLASSICAL (textbook CG, the solver the network was unrolled from), EXACT (CG to convergence: the dense inverse's answer), BERNSTEIN (the gradient-trained direct head), and the CLOSED-FORM head of Section 3.4; both direct heads start at the textbook solver's polynomial (Appendix B). Everything else — operator learner, thresholds, layer count, loss, optimizer — is identical across arms. Appendix G.1 checks Theorem 1 to machine precision.

4.1 E6: DEPTH IS DEGREE, IN CONTEXT

The transformer instance of Proposition 2 runs on in-context linear regression ($d{=}8$, $n{=}32$, condition number κ): attention with learned scalar steps, the direct Chebyshev head of degree $L{-}1$, and attention with learned $d{\times}d$ preconditioners (outside the class), against L steps of textbook CG and the exact solution, spectrally normalized per prompt (Appendix I).

Figure 5b shows the fixed-constant arms, scalar steps, momentum, the direct head, and even the unrestricted matrices at small depth on the *same* curve, with the closed-form head as that class's optimum. Adaptive CG beats the class by a factor that grows with the condition number and depth, Horn 1 of Proposition 4 as a number. At $\kappa{=}100$, $L{=}16$ the Adam-trained head fails while the step forms and the closed form do not (Table 3).

When the task is not the solve. Give each task a dominant nuisance mode: the top eigen-direction of its covariance couples to the context labels but not to the query label. The optimal update now solves the signal band and notches the top mode — a high-pass response. At $\kappa{=}100$ the exact solution and adaptive CG fit the nuisance mode and pay for it (test MSE 68.5 at both depths). Every fixed-constant rule does better. The closed-form head is the class optimum (29.0 at $L{=}4$, 25.3 at $L{=}8$), and learned step sizes fall 8% behind it at $L{=}4$ and 29% at $L{=}8$: the real-rooted restriction of Corollary 1, opening with degree (Table 4).

4.2 E7: DEPTH IS DEGREE ON A SEQUENCE TASK

Multi-query associative recall (MQAR) (Arora et al., 2024): n key–value pairs are presented, keys are queried later, accuracy is reported against load n. One causal layer keeps the in-context least-squares state $(\mathbf{S}_t, \mathbf{U}_t)$ and reads out $\mathbf{q}_t^\top p(\mathbf{S}_t)\mathbf{U}_t$ with exactly T applications of $\mathbf{S}_t$; p is each rule of Section 3.2, the closed-form head, its gradient-trained variant, degree zero, and the exact solve; two layers ($d{=}64$) train at load 16 and test at 4–64 (Appendix I).

Figure 2 shows three things (Table 11; Figure 7). *Degree buys recall under load, and the rule does not*: at degree 4 the head, learned step sizes, momentum, and conjugate gradient tie at the median (86–87 at load 32, 56–57 at load 64). Degree 8 adds two points at most, and linear attention trails by $12/17$. *The gradient-trained members are the unreliable ones*: learned step sizes fail on three seeds of ten (59–70 at load 32), the gradient-trained head on two of three, and linear attention on five of ten. The head's worst seed is $76/52$: Proposition 5 as seed variance, and the reason the head is computed, not learned. *The solve is not the task under interference* (Proposition 4, Horn 2): with the ridge lowered from $10^{-1}\lambda_{\max}$ to $10^{-3}\lambda_{\max}$ the solve-targeting rules collapse at load while learned

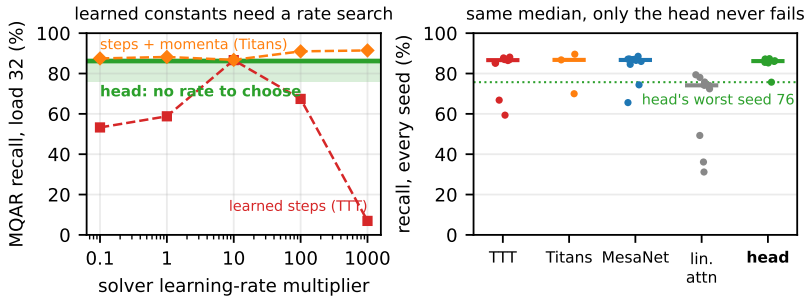


Figure 2: **Learned constants need a rate search and still lose seeds; the head needs neither (E7, MQAR, degree 4).** Left: recall against the solver learning rate — learned step sizes collapse away from $10\times$; the head has no rate. Right: all ten seeds at load 32; the medians tie, and the head's worst seed (76) is above every other rule's worst.

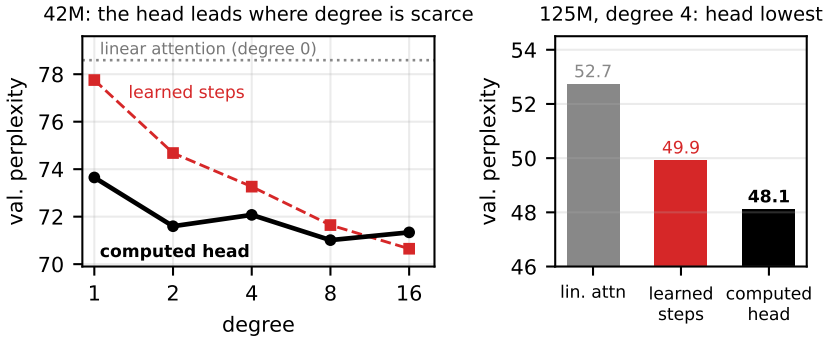


Figure 3: **In language models the head leads where degree is scarce and is lowest at 125M.** Left: 42M model, perplexity against degree; learned step sizes close the gap only by degree 16. Right: 125M, degree 4, same token budget: head 48.1, learned steps 49.9, linear attention 52.7 (Appendix G.7).

steps hold, a damped filter, which is where MesaNet's tuned Λ puts it (Figure 11, one seed). Degree buys most where interference is highest, where the layers at scale report their recall gains (Yang et al., 2024; 2025; von Oswald et al., 2026). At 42M and 125M the ordering repeats (Figure 3, Appendix G.7).

4.3 E4: A FIXED OPERATOR, PRESCRIBED RESPONSES — WHERE THE RESTRICTION COSTS ACCURACY

A thin reachable set costs nothing if it passes near the optimum, and for low-pass targets it should: Proposition 1 is about shape. On a fixed symmetric operator (a grid-graph Laplacian with bilateral weights) at matched degree $d{=}20$ we fit the nested unrolled parameterization, learned step sizes, learned steps with momenta, and the direct head to two tasks and two prescribed responses (Figure 4, Table 6). On denoising and deblurring the restriction costs nothing to 1.4 dB; on a detail band and a bandlimited projection the head reaches the 60 dB cap while the nested arm stops 18 dB short and learned step sizes 42 dB short (Corollary 1); the nested gap widens with the budget.

4.4 E2: THE OPERATOR RE-ESTIMATED IN EVERY LAYER — CONDITIONING, NOT CAPACITY

We train the restoration network under each arm at matched degree (Table 2; equal accuracy predicted; Proposition 5 says which arm gets there reliably). Each arm trains its own operator learner, so arms are compared at their own operators; the class ceiling of the system the layers declare is the EXACT arm (30.11 dB). Over three seeds the textbook solver with no constants lands at 29.30 ± 0.66 dB and the bounded direct head, started at the textbook polynomial, at 29.43 ± 0.10 — a tie under the ceiling, the head's spread a tenth of a decibel against the solver's two thirds. The unrolled solver

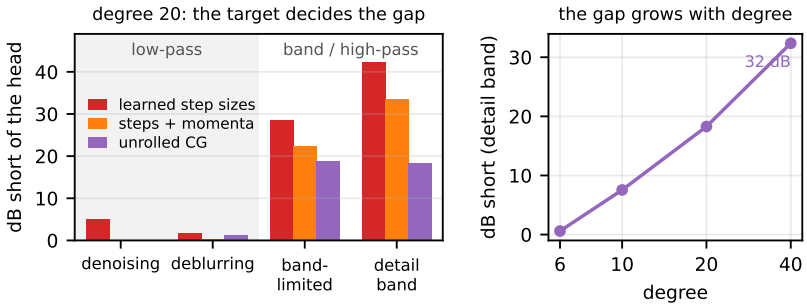


Figure 4: **The target decides the gap (E4).** Left: dB short of the computed head (the class optimum) at degree 20 on a fixed operator: nothing on low-pass targets, 18–42 dB on band targets. Right: the unrolled-CG gap on the detail band grows with degree.

Table 2: E2. Deleting the solver constants inside one unrolled solver network whose operator is re-estimated in every layer (image restoration). One network, one protocol, matched degree d; PSNR on CBSD68 (dB; denoising σ=25, three seeds, median $\pm$ half-range; deblurring seed 0, prior kept on); constants per layer in parentheses; *three seeds at the one rate of a $\{0.01, 0.1, 1, 10\}\times$ grid that does not collapse; †seed 0. In denoising every fixed-constant arm's median sits under the class ceiling (the exact solve of the system the layers declare); the one arm above it is the one whose constants trained jointly with the operator learner. Full table in Appendix G.3.

task	d	class ceiling exact solve	textbook solver PSNR (0)	bounded head PSNR (constants)	least-squares head PSNR (0)	learned constants PSNR (constants)	best rate PSNR
denoising, σ=25	10	$30.11^{\dagger}$	29.30 ± 0.66	29.43 ± 0.10 (11)	20.9 ± 5.7	12.4 ± 9.1 (30)	$\mathbf{30.20 \pm 0.09}^{*}$
deblurring, prior on	10	28.77	**28.83**	28.29 (11)	27.72	28.57 (30)	25.71
GDD denoising	20	—	28.06	28.10 (21)	—	28.10 (17)	—

with learned step sizes collapses on every seed at the shared rate (6.7, 12.4, 25.0). On a four-rate grid three rates collapse and the fourth reaches 30.20 ± 0.09 over three seeds, 0.1 dB *above* the exact solve of the system its own layers declare. No polynomial beats the solve of its system, so those constants did not find a better filter for that operator: they co-adapted with the operator learner into a different operator — Theorem 2(ii) measured — at the price of a rate search the head never pays. The least-squares head decays the same way on two seeds of three. Both unbounded arms, learned constants and the least-squares head, fit the spectra of the 48^2 training crops and amplify modes of full-size images outside that range (Proposition 1). The bounded head, whose response cannot leave $[0, 1]$, holds (Figure 9). In deblurring, with the learned prior on (a floor $\mu \geq 0.03$ for every arm), the exact solve, textbook CG, the bounded head and learned constants at the shared rate land within 0.55 dB (Table 2, seed 0): one class, Theorem 1; the least-squares head sits 1.1 dB lower, and tuning the solver rate per arm buys nothing (Appendix G.3).

5 Discussion and Limitations

What "each layer is an iteration" means. It is a claim about initialization: Proposition 4 makes the drift measurable as the solve residual — on a published network the trained constants leave the declared system by $600\times$ while matching textbook CG, and on a released checkpoint replacing them by the actual solve costs 11 dB (Appendix I); the layer *is* a spectral response on a learned operator. *Limitations:* the theorems take the symmetric, definite setting of Section 3; sequence-layer results are at 42M and 125M, one seed, one protocol, one GPU; the ordering holds at both (Figure 3, right); restoration arms share one network at equal budget.

Reproducibility Statement

All theorems are proved in Appendix A; the Bernstein and closed-form heads are specified in Appendix B and every experimental setting (architectures, budgets, learning-rate grids, seeds, calibration, data splits) in Appendix I. The code is a single package (our matrix-free graph-solver package) with one script per experiment: the collapse check (E1), the restoration arms (E2), the reachable-

set gaps (E4), the breadth survey (E5), the in-context study (E6) and the MQAR study (E7), each writing a JSON record that the table-generating script consumes; the anonymized repository is supplied as supplementary material. Restoration runs use one seed unless a spread is reported (0–4 on deblurring), one shared RTX 3090, and 30–90 minutes per run; E1, E4, E5 and E6 run on CPU in minutes; each MQAR run takes about 20 minutes on the same GPU. Datasets are public (DIV2K and BSDS300 for training, CBSD68 for testing).

USE OF LARGE LANGUAGE MODELS

The authors used a large language model (Claude, Anthropic) to draft and edit text, to write and debug experiment code under their direction, and to search the literature. All theorems and proofs, all reported numbers, and all claims were verified by the authors, who take full responsibility for the content.

REFERENCES

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

Jianghe Cai, Gene Cheung, and Fei Chen. Unrolling plug-and-play gradient graph Laplacian regularizer for image restoration. *IEEE Trans. Image Process.*, 35:1817–1831, 2026.

Stanley H. Chan. Performance analysis of plug-and-play ADMM: A graph signal processing perspective. *IEEE Trans. Comput. Imaging*, 5(2):274–286, 2019. arXiv:1809.00020.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016.

Tam Thuc Do, Parham Eftekhar, Seyed Alireza Hosseini, Gene Cheung, and Philip A. Chou. Interpretable lightweight transformer via unrolling of learned graph smoothness priors. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

Tam Thuc Do, Philip A. Chou, and Gene Cheung. Deep unrolling of sparsity-induced RDO for 3D point cloud attribute coding. *IEEE Transactions on Image Processing*, 2026.

Parham Eftekhar, Gene Cheung, Mingxiao Liu, H. Vicky Zhao, Xuejun Han, and Eugene Wen. Unrolling iterative Lanczos algorithm for ideal low-pass graph filter approximation. *arXiv preprint arXiv:2609.29086*, 2026.

Rida T. Farouki and V. T. Rajan. On the numerical condition of polynomials in Bernstein form. *Comput. Aided Geom. Design*, 4(3):191–216, 1987.

Khashayar Gatmiry, Nikunj Saunshi, Sashank J. Reddi, Stefanie Jegelka, and Sanjiv Kumar. On the role of depth and looping for in-context learning with task diversity. *arXiv preprint arXiv:2410.21698*, 2024.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, 4th edition, 2013.

Anne Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, 1997.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proc. Int. Conf. Machine Learning (ICML)*, 2010.

David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon. Anal.*, 30(2):129–150, 2011.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. BernNet: Learning arbitrary graph spectral filters via Bernstein approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Seyed Alireza Hosseini. Unrolling a graph-Laplacian denoiser realizes only compositions of polynomial graph filters. *arXiv preprint arXiv:2608.09923*, 2026.

Seyed Alireza Hosseini, Tam Thuc Do, Gene Cheung, and Yuichi Tanaka. Constructing an interpretable deep denoiser by unrolling graph Laplacian regularizer. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 1431–1437, 2024.

Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations (ICLR)*, 2019.

Junchen Liu, Sven Elflein, Or Litany, Zan Gojcic, and Ruilong Li. Test-time training with KV binding is secretly linear attention. *arXiv preprint arXiv:2602.21204*, 2026a.

Zihang Liu, Oleg Balabanov, Yaoqing Yang, and Michael W. Mahoney. Learning to discover iterative spectral algorithms. *arXiv preprint arXiv:2602.09530*, 2026b.

George G. Lorentz. *Bernstein Polynomials*. Univ. of Toronto Press, 1953.

Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. A unified view on graph neural networks as graph signal denoising. In *Proc. ACM International Conference on Information and Knowledge Management (CIKM)*, 2021.

Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, 2013.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.*, 38(2):18–44, 2021.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003.

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Neehal Tumma, Noel Loo, and Daniela Rus. Preconditioned DeltaNet: Curvature-aware sequence modeling for linear recurrences. *arXiv preprint arXiv:2604.21100*, 2026.

Johannes von Oswald, Nino Scherrer, Seijin Kobayashi, Luca Versari, Songlin Yang, Sarthak Mittal, Maximilian Schlegel, Kaitlin Maile, Yanick Schimpf, Oliver Sieberling, Alexander Meulemans, Rif A. Saurous, Guillaume Lajoie, Charlotte Frenkel, Razvan Pascanu, Blaise Agüera y Arcas, and João Sacramento. MesaNet: Sequence modeling by locally optimal test-time training. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2506.05233.

Huy Vu, Gene Cheung, and Yonina C. Eldar. Unrolling of deep graph total variation for image denoising. In *Proc. IEEE ICASSP*, pp. 2050–2054, 2021.

Tomoya Wakayama. A decision-theoretic view of test-time training: When, how far, and which directions to adapt. In *International Conference on Machine Learning (ICML)*, 2026.

Songlin Wei, Gene Cheung, Fei Chen, and Ivan Selesnick. Unrolling nonconvex graph total variation for image denoising. In *Proc. IEEE International Conference on Image Processing (ICIP)*, 2025.

Mingsong Yan, Dongyang Li, Charles Kulick, and Sui Tang. Transformers can implement preconditioned Richardson iteration for in-context Gaussian kernel regression. *arXiv preprint arXiv:2605.08475*, 2026.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025.

Yongyi Yang, Tang Liu, Yangkun Wang, Jinjing Zhou, Quan Gan, Zhewei Wei, Zheng Zhang, Zengfeng Huang, and David Wipf. Graph neural networks inspired by classical iterative algorithms. In *International Conference on Machine Learning (ICML)*, 2021.

Junyi Yao, Parham Eftekhar, Gene Cheung, Wei Hu, Chris Liu, and Yao Wang. Importance of balance: Lightweight transformer via signed graph algorithm unrolling for EEG signal denoising. In *Proc. IEEE ICASSP*, 2026a.

Junyi Yao, Parham Eftekhar, Gene Cheung, Xujin Chris Liu, Yao Wang, and Wei Hu. Lightweight transformer for EEG classification via balanced signed graph algorithm unrolling. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2026b.

Jin Zeng, Jiahao Pang, Wenxiu Sun, and Gene Cheung. Deep graph Laplacian regularization for robust denoising of real images. In *Proc. IEEE/CVF CVPR Workshops (CVPRW)*, pp. 1759–1768, 2019.

Xue Zhang, Bingshuo Hu, and Gene Cheung. Unrolling graph-based Douglas–Rachford algorithm for image interpolation with informed initialization. *arXiv preprint arXiv:2509.11926*, 2025.

Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, and Peng Cui. Interpreting and unifying graph neural networks with an optimization framework. In *Proc. The Web Conference (WWW)*, 2021.

A Proofs and closed-form results for the nested instance

This appendix collects the instance-level results quoted in Section 3 for the nested construction of Example 1 (a truncated Taylor expansion of the system matrix inside an unrolled conjugate-gradient solver), with proofs. Throughout, $\mathbf{\Psi} \in \mathbb{R}^{N\times N}$ is the graph-Laplacian denoising operator $\mathbf{\Psi} = (\mathbf{I}+\mu\widehat{\mathcal{L}})^{-1}$ with $\widehat{\mathcal{L}}$ the symmetrically normalized Laplacian and $\mu > 0$, so $\mathrm{spec}(\mathbf{\Psi}) \subset (0, 1]$ and the system matrix is $\mathbf{A} = \mathbf{I}+\mu\widehat{\mathcal{L}} = \mathbf{\Psi}^{-1}$. The TSE module forms $t_K(\mathbf{\Psi}) = \sum_{k=0}^{K} a_k s^{-(k+1)}(\mathbf{\Psi}-s\mathbf{I})^k$ with learned $\{a_k\}$, s (initialized at the truncated Taylor series of $1/\lambda$ about $s = 1$: $a_k = (-1)^k$), and the CG module runs m steps with learned $\{\alpha_k\}, \{\beta_k\}$. Notation: $\mathcal{P}_d$ is the space of polynomials of degree at most d and $\mathcal{K}_j = \mathrm{span}\{\mathbf{y}, \mathbf{A}_K\mathbf{y}, \ldots, \mathbf{A}_K^{j-1}\mathbf{y}\}$.

Lemma 1. *Let $\widehat{\mathcal{L}} = \mathbf{I} - \mathbf{W}$ with $\mathbf{W} = \mathbf{S}^{-1/2}\mathbf{B}\mathbf{S}^{-1/2}$, $\mathbf{B}$ symmetric with non-negative entries and $\mathbf{S} = \mathrm{diag}(\mathbf{B1})$ positive. Then $\mathbf{W}\mathbf{S}^{1/2}\mathbf{1} = \mathbf{S}^{1/2}\mathbf{1}$ and, $\mathbf{W}$ being similar to the row-stochastic $\mathbf{S}^{-1}\mathbf{B}$, $\mathrm{spec}(\mathbf{W}) \subset [-1, 1]$; so $\lambda_{\max}(\mathbf{\Psi}) = 1$ exactly and*

$$\kappa(\mathbf{\Psi}) = 1/\lambda_{\min}(\mathbf{\Psi}) = 1 + \mu\,(1 - w_{\min}), \tag{5}$$

with $w_{\min}$ the smallest eigenvalue of $\mathbf{W}$.

Proposition 6 (TSE module)**.** *For any $a_0, \ldots, a_K \in \mathbb{R}$ and any $s \neq 0$,*

$$t_K(\mathbf{\Psi}) = \sum_{k=0}^{K} \frac{a_k}{s^{k+1}}(\mathbf{\Psi} - s\mathbf{I})^k \tag{6}$$

is a polynomial in $\mathbf{\Psi}$ of degree at most K. Conversely, with s fixed, every polynomial of degree at most K is realized by exactly one choice of $(a_0, \ldots, a_K)$. The module's reachable set is therefore exactly $\mathcal{P}_K$.

Proof. The $K+1$ polynomials $\{(\lambda-s)^k\}_{k=0}^K$ have distinct degrees, hence are independent and span $\mathcal{P}_K$; the rescaling a_k/s^{k+1} is invertible for $s \neq 0$. Polynomials in a fixed matrix form a commutative algebra, so substituting $\mathbf{\Psi}$ preserves this. □

Theorem 3 (Collapse). *Run the m-step CG module on $t_K(\mathbf{\Psi})\mathbf{x} = \mathbf{y}$ from $\mathbf{x}_0 = 0$ with arbitrary step sizes $\{\alpha_k\}_{k<m}$ and $\{\beta_k\}_{k<m-1}$, learned or otherwise. Then the output satisfies*

$$\mathbf{x}_m = P(\mathbf{\Psi})\,\mathbf{y}, \qquad \deg P \leq K(m-1), \tag{7}$$

for every setting of the learned coefficients.

Proof. Write $\mathbf{A}_K = t_K(\mathbf{\Psi})$ and $\mathcal{K}_j = \operatorname{span}\{\mathbf{y}, \mathbf{A}_K\mathbf{y}, \ldots, \mathbf{A}_K^{j-1}\mathbf{y}\}$. We claim $\mathbf{x}_k \in \mathcal{K}_k$, $\mathbf{r}_k \in \mathcal{K}_{k+1}$ and $\mathbf{p}_k \in \mathcal{K}_{k+1}$ for all k. This holds at $k=0$, where $\mathbf{x}_0 = 0$ and $\mathbf{r}_0 = \mathbf{p}_0 = \mathbf{y}$. Assuming it at k, and using $\mathbf{A}_K\mathcal{K}_{k+1} \subseteq \mathcal{K}_{k+2}$,

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k\mathbf{p}_k \in \mathcal{K}_{k+1},\\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\mathbf{A}_K\mathbf{p}_k \in \mathcal{K}_{k+2},\\
\mathbf{p}_{k+1} &= \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k \in \mathcal{K}_{k+2}.
\end{aligned}$$

The coefficients α_k, β_k appear only as scalars multiplying vectors already in the subspace; the sole operation that raises the Krylov index is multiplication by $\mathbf{A}_K$, and the recursion performs exactly one per step whatever their values. Hence $\mathbf{x}_m \in \mathcal{K}_m$, i.e. $\mathbf{x}_m = q_{m-1}(\mathbf{A}_K)\mathbf{y}$ with $\deg q_{m-1} \leq m-1$. By Proposition 6, $\deg t_K \leq K$, so $P = q_{m-1} \circ t_K$ has $\deg P \leq K(m-1)$. □

Proposition 7 (The reachable set is thin). *The set realized by the block is $\mathcal{R} = \{q \circ t : \deg q \leq m-1,\ \deg t \leq K\}$, the image of a polynomial map from $\mathbb{R}^{K+2m+1}$. It is therefore a semialgebraic set of dimension at most $K+2m+1$. Whenever $K(m-2) > 2m$,*

$$\dim \mathcal{R} \;\leq\; K+2m+1 \;<\; K(m-1)+1 \;=\; \dim \mathcal{P}_{K(m-1)}, \tag{8}$$

so $\mathcal{R}$ is a strict subset of $\mathcal{P}_{K(m-1)}$ of Lebesgue measure zero within it.

Proof. With $\tilde{a}_k = a_k/s^{k+1}$ (a bijection of $\mathbb{R}^{K+1}$ for each $s \neq 0$), $\mathcal{R}$ is the image of $(\tilde{a}, s, \alpha, \beta) \in \mathbb{R}^{K+2m+1}$ under a map with polynomial coordinates, hence semialgebraic with dimension at most that of its domain. Inequality equation 8 is $K+2m < K(m-1)$, i.e. $K(m-2) > 2m$. A semialgebraic subset of a vector space of strictly smaller dimension has measure zero. □

Proposition 8 (Spectral response at initialization). *At $a_k = (-1)^k$ and $s = 1$ the TSE module realizes $t_K(\lambda) = \big(1-(1-\lambda)^{K+1}\big)/\lambda$. If the CG module solves exactly, the end-to-end response is*

$$g_K(\lambda) = \frac{\lambda}{1-(1-\lambda)^{K+1}}, \qquad \lambda \in (0,1], \tag{9}$$

which is increasing, satisfies $g_K(1) = 1$ and $g_K(\lambda) > \lambda$ for $\lambda < 1$, and obeys $g_K(\lambda) \to 1/(K+1)$ as $\lambda \to 0^+$. Consequently the deviation from the target response is bounded by

$$\big\|\mathbf{x}_{\text{init}} - \mathbf{\Psi}\mathbf{y}\big\|_2 \;\leq\; \frac{1}{K+1}\,\|\mathbf{y}\|_2, \tag{10}$$

and the bound is approached as $\lambda_{\min} \to 0$.

Proof. At $a_k = (-1)^k$, $s = 1$, equation 6 telescopes to $t_K = \sum_{k=0}^K u^k$ with $u = 1-\lambda$, giving equation 9 on inversion. That sum increases in u and u decreases in λ, so g_K increases; $g_K(1) = 1$; and $0 < 1-u^{K+1} < 1$ for $\lambda < 1$ gives $g_K(\lambda) > \lambda$. Using $1-u^{K+1} = \lambda\sum_{k\leq K} u^k$,

$$g_K(\lambda) - \lambda = \frac{\lambda u^{K+1}}{1-u^{K+1}} = \frac{u^{K+1}}{\sum_{k=0}^K u^k}, \tag{11}$$

which increases in u on $[0,1)$ with limit $1/(K+1)$, so it is bounded by $1/(K+1)$ throughout. Symmetry of $\mathbf{\Psi}$ turns this into equation 10. □

Corollary 2 (Truncation order)**.** *The relative error of t_K as an approximation of $1/\lambda$ is $(1-\lambda)^{K+1}$, so accuracy ε uniformly over $\mathrm{spec}(\mathbf{\Psi})$ requires*

$$K+1 \;\geq\; \frac{\log(1/\varepsilon)}{-\log(1-\lambda_{\min})} \;\gtrsim\; \kappa(\mathbf{\Psi})\log(1/\varepsilon). \tag{12}$$

Proposition 9. *Let the spatial weights be $e^{-a(p-q)^2}$ truncated to $|p-q| \leq r$, and set*

$$\sigma_r(\theta) = 1 + 2\sum_{k=1}^{r} e^{-ak^2}\cos(k\theta). \tag{13}$$

If $\sigma_r(\pi) < 0$ then, for all n large enough that the numerator of equation 14 is negative, $w_{\min} < 0$ and hence $\kappa(\mathbf{\Psi}) > 1+\mu$.

Proof. Here $\mathbf{W} = \mathbf{S}^{-1/2}\mathbf{B}\mathbf{S}^{-1/2}$ is a congruence of $\mathbf{B}$, so by Sylvester's law of inertia both have equally many negative eigenvalues and it suffices to exhibit x with $x^\top\mathbf{B}x < 0$. Absent photometric variation the features reduce to (x, y), so weights and window factor: $\mathbf{B} = \mathbf{T}\otimes\mathbf{T}$ and $\mathbf{W} = \mathbf{V}\otimes\mathbf{V}$ with $\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{T}\mathbf{D}^{-1/2}$, $\mathbf{D} = \mathrm{diag}(\mathbf{T}\mathbf{1})$; as $\lambda_{\max}(\mathbf{V}) = 1$ by Lemma 1, $w_{\min} = \lambda_{\min}(\mathbf{V})$. Take $x_p = (-1)^p$; since $\#\{(p,q) : p-q = k\} = n - |k|$ and $x_p^2 = 1$,

$$\frac{x^\top\mathbf{T}x}{x^\top\mathbf{D}x} = \frac{n\,\sigma_r(\pi) - \sum_{|k|\leq r}|k|\,e^{-ak^2}(-1)^k}{\mathbf{1}^\top\mathbf{T}\mathbf{1}}, \tag{14}$$

whose numerator turns negative once n passes the threshold; it bounds $\lambda_{\min}(\mathbf{V})$ above, and equation 5 gives the claim. □

Proposition 10. *Let $P(\lambda) = \sum_{j=0}^{d} c_j \binom{d}{j}\lambda^j(1-\lambda)^{d-j}$. Then: (i) as $\{c_j\}$ ranges over $\mathbb{R}^{d+1}$, P spans all of $\mathcal{P}_d$; (ii) if $c_j \in [0,1]$ for all j then $P(\mathbf{\Psi})$ is positive semidefinite and $\|P(\mathbf{\Psi})\|_2 \leq 1$, for every $\mathbf{\Psi}$ with $\mathrm{spec}(\mathbf{\Psi}) \subset [0,1]$; and (iii) $c_j = j/d$ gives $P(\lambda) = \lambda$ exactly, i.e. the target $\mathbf{\Psi}\mathbf{y}$ lies in the hypothesis class at zero approximation error.*

Proof. (i) Conversion to the monomial basis is triangular with nonzero diagonal, $b_{j,d}$ having lowest term $\binom{d}{j}\lambda^j$ (Lorentz, 1953). (ii) On $[0,1]$ the $b_{j,d}$ are nonnegative and $\sum_j b_{j,d}(\lambda) = (\lambda+(1-\lambda))^d = 1$, so $P(\lambda)$ is a convex combination of the c_j and lies in $[0,1]$; as $\mathbf{\Psi}$ is symmetric, $\mathrm{spec}(P(\mathbf{\Psi})) = P(\mathrm{spec}(\mathbf{\Psi})) \subset [0,1]$, which is PSD together with $\|P(\mathbf{\Psi})\|_2 \leq 1$. (iii) For $X \sim \mathrm{Binomial}(d,\lambda)$, $\Pr[X = j] = b_{j,d}(\lambda)$, so $\sum_j (j/d)\,b_{j,d}(\lambda) = \mathbb{E}[X]/d = \lambda$. □

B THE BERNSTEIN ARM INSIDE A NETWORK

Construction ($\mathbf{I}+\mu\mathbf{L}$ solves). For a block solving $(\mathbf{I}+\mu\mathbf{L})\mathbf{x} = \mathbf{y}$ with $\mathrm{spec}(\mathbf{L}) \subset [0, \lambda_{\max}]$ ($\lambda_{\max}$ from a 12-step Lanczos bound per batch), set $t = \lambda/\lambda_{\max} \in [0,1]$ and $p_c(\mathbf{L}) = \sum_j c_j\, b_{j,d}(\mathbf{L}/\lambda_{\max})$. The polynomial is evaluated with d matvecs through the three-term Chebyshev recurrence after a stable Bernstein-to-Chebyshev change of basis (values at Chebyshev nodes; Horner on monomial coefficients is unusable beyond $d \approx 20$). Coefficients: $c = c^\star(\mu) + \delta$ where $c^\star(\mu)$ is the least-squares projection of $1/(1+\mu\lambda_{\max}t)$ onto $\mathcal{P}_d$ in the Bernstein basis under the Chebyshev measure (a fixed $(d{+}1)\times(d{+}1)$ Gram solve per μ; batched for DeepGLR's per-sample μ) and $\delta \in \mathbb{R}^{d+1}$ is learned. This keeps the μ-adaptivity of the published block, starts the arm at the best degree-d approximation of the intended solve, and gives it exactly $d+1$ learned constants — the same count as $d{+}1$ CG steps with learned (α_t, β_t) up to the dead final momentum. For the masked interpolation solve of uGLR the operator is $\mathbf{M}\mathbf{L}\mathbf{M}$ on the unknown pixels and the same construction applies with μ absent. Box variant: $c_j = \gamma\,\sigma(\tilde{c}_j)$ with γ fixed at 1 or learned (≥ 1).

Construction (generic PSD solves; deblurring). For $(\mathbf{H}^\top\mathbf{H}+\mu\mathbf{L})\mathbf{x} = \mathbf{H}^\top\mathbf{y}$ the operator is not a shifted Laplacian, so $c^\star$ has no μ-closed form. Instead both learned arms are *calibrated* on one training batch: classical CG is run for T steps, its batch-median (α_t, β_t) initialize the unrolled arm, and the *same* degree-$(T{-}1)$ polynomial (from the CG coefficient recursion) is converted to Bernstein coefficients in $t = \lambda/\lambda_{\max}$, with $\lambda_{\max}$ fixed at $1.1\times$ the batch Lanczos bound; both arms therefore start at the identical filter and differ only in their coordinates. The solve is warm-started at $\mathbf{y}$ and the polynomial applied to the residual.

Closed-form head. The direct head is linear in its coefficients: $\mathbf{x}(a) = \mathbf{x}_0 + \sum_{k=0}^{d} a_k\, T_k(\hat{\mathbf{A}})\, \mathbf{r}$ with $\hat{\mathbf{A}} = 2\mathbf{A}/\lambda_{\max} - \mathbf{I}$ and $\mathbf{r}$ the warm-start residual, so for a fixed network its class optimum under the squared loss is the $(d{+}1)$-dimensional normal-equation solve $a = (\mathbb{E}\,\mathbf{V}^\top\mathbf{V})^{-1}\,\mathbb{E}\,\mathbf{V}^\top(\mathbf{x}^* - \mathbf{x}_0)$, $\mathbf{V} = [T_k(\hat{\mathbf{A}})\mathbf{r}]_k$ (Proposition 5). The "closed-form head" arm trains by alternating minimization: after every gradient step on the network, the expectations are updated as exponential moving averages over training batches (decay 0.99, Tikhonov ridge 10^{-6} of the mean diagonal) and a is reset to the solve; no gradient reaches a, and the network receives gradient through the basis $\mathbf{V}$ at the current a. The head starts at the calibrated CG polynomial, exactly as the gradient-trained Bernstein head does, and carries the same $d{+}1$ constants.

C Corollaries, remarks and the worked example

The following statements are referenced from Section 3.

Corollary 3 (Instances)**.** *For T steps: gradient descent and Richardson with learned steps realize $\{p \in \mathcal{P}_T : p(0)$ fixed by the initialization$\}$ at most; heavy-ball and Nesterov with learned steps and momenta, and conjugate gradient with learned (α_t, β_t), realize subsets of $\mathcal{P}_T$; Chebyshev iteration with learned recurrence coefficients realizes all of $\mathcal{P}_T$ (it is a direct parameterization in the Chebyshev basis). In every case the reachable set is contained in $\mathcal{P}_T$ and the degree bound cannot be exceeded by any choice of the learned constants. (For the solver iterates themselves the bound is one lower — T steps of gradient descent or conjugate gradient from $\mathbf{x}_0 = 0$ give $\mathbf{x}_T \in \mathcal{K}_T(\mathbf{A}, \mathbf{b})$, degree $T - 1$ — which is the "matched degree" used in Section 4.)*

Corollary 4 (Nested iterations)**.** *If the operator of an outer iteration is itself produced by an inner linear Krylov iteration — an inner polynomial $t_K(\mathbf{A})$ of degree K applied by an outer solver for m steps — then the output is $p(\mathbf{A})\mathbf{b}$ with $\deg p \le K\,(m-1)$ (for a solver whose m-step output lies in $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$), for every setting of both sets of coefficients.*

Remark 1 (Lanczos filters). An m-step Lanczos spectral filter $\|\mathbf{b}\|\,\mathbf{U}_m g(\mathbf{H}_m)\mathbf{e}_1$ with a learned response g (e.g. a learned soft cutoff (Yao et al., 2026b)) also lies in $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ for every value of its parameters, so Theorem 1's *subspace* statement holds. The realizing polynomial, however, interpolates g at the Ritz values of $\mathbf{H}_m$, which depend on $\mathbf{b}$, so the map is a data-dependent polynomial filter rather than a fixed one. Learned parameters still select within a subspace they cannot enlarge, but they cannot make the filter linear.

Example 1 (A nested graph-Laplacian denoiser)**.** *The construction of Hosseini et al. (2024) forms a system matrix from a graph-Laplacian denoiser $\mathbf{\Psi}$ by an order-K truncated Taylor expansion with learned coefficients $\{a_k\}_{k \le K}$ and expansion point s, then inverts it by m steps of conjugate gradient with learned $\{\alpha_t\}_{t<m}$, $\{\beta_t\}_{t<m-1}$. By Corollary 4 its output is $P(\mathbf{\Psi})\mathbf{y}$ with $\deg P \le K(m-1)$; by Proposition 1 with $P_{\text{params}} = K + 2m + 1$, whenever $K(m-2) > 2m$ the reachable set has dimension at most $K + 2m + 1 < K(m-1) + 1 = \dim \mathcal{P}_{K(m-1)}$ and is measure-zero in it. At the published setting $K = 10$, $m = 3$: seventeen constants, a class of dimension twenty-one, and a Jacobian whose numerical rank at initialization is twelve (Section 4; Hosseini, 2026). The last momentum update is dead code — β_{m-1} never reaches the output — and rescalings of the inner polynomial absorbed by the outer one leave the response unchanged, which accounts for the gap between seventeen and twelve.*

D Proofs of the statements in Section 3

Theorem 1 (uniform Krylov collapse).

Proof. Induction on t: every state vector lies in $\mathcal{K}_{t+1}(\mathbf{A}, \mathbf{b})$. At $t = 0$ the state lies in $\operatorname{span}\{\mathbf{b}\} = \mathcal{K}_1(\mathbf{A}, \mathbf{b})$. If every $\mathbf{v}_t^i \in \mathcal{K}_{t+1}(\mathbf{A}, \mathbf{b})$ then $\mathbf{A}\mathbf{v}_t^i \in \mathcal{K}_{t+2}(\mathbf{A}, \mathbf{b})$, and Φ_t forms linear combinations of vectors in $\mathcal{K}_{t+2}(\mathbf{A}, \mathbf{b})$ with scalar coefficients, which stay in $\mathcal{K}_{t+2}(\mathbf{A}, \mathbf{b})$. The coefficients enter only as scalars multiplying vectors already in the subspace; the sole operation that raises the Krylov index is an application of $\mathbf{A}$, and the recursion performs at most one per vector per step whatever the coefficients are. Hence $\mathbf{x}_T \in \mathcal{K}_{T+1}(\mathbf{A}, \mathbf{b})$, i.e. $\mathbf{x}_T = p(\mathbf{A})\mathbf{b}$ with $\deg p \le T$. Since the coefficients do not depend on $\mathbf{b}$, the composition of the linear maps Φ_t is a fixed linear map of $\mathbf{b}$, namely $p_\theta(\mathbf{A})$. □

Corollary 4 (nested iterations).

Proof. Apply Theorem 1 with $\mathbf{A}_K = t_K(\mathbf{A})$ to get $\mathbf{x} = q(\mathbf{A}_K)\mathbf{b}$, $\deg q \le m-1$; then $p = q \circ t_K$. □

Proposition 2 (depth is degree).

Proof. Unrolling equation 4 from $\mathbf{w}_0 = 0$ gives $\mathbf{w}_L = \sum_k a_k \prod_{j>k}(\mathbf{I} - a_j\hat{\mathbf{\Sigma}})\,\mathbf{b}$, a polynomial in $\hat{\mathbf{\Sigma}}$ of degree $L-1$ applied to $\mathbf{b}$; multiplying by λ and telescoping gives the product form in the statement. Every real-rooted q with $q(0) = 1$ is realized by $a_k = 1/\mathrm{root}_k$; the Chebyshev polynomial of degree L shifted to $[\lambda_{\min}, \lambda_{\max}]$ is real-rooted, and real-rootedness is preserved under small perturbations of the coefficients (nonempty interior) while excluding, e.g., polynomials with complex roots (strictness). For matrices $\mathbf{A}_k$ that do not commute with $\hat{\mathbf{\Sigma}}$ the product $\prod_j(\mathbf{I} - \mathbf{A}_j\hat{\mathbf{\Sigma}})$ is not a function of $\hat{\mathbf{\Sigma}}$ alone. □

Proposition 1 (reachable set).

Proof. The coefficients of p_θ are polynomials in θ (products of the step constants along the recursion), so $\mathcal{R}$ is the image of a polynomial map and semialgebraic; its dimension is at most that of its domain and at most the generic rank of the Jacobian. A semialgebraic subset of $\mathbb{R}^{T+1}$ of smaller dimension has measure zero. □

Theorem 2 (localization).

Proof. (i) Apply Theorem 1 block-wise; the direct parameterization reaches all of $\mathcal{P}_{T_b}$ and hence a superset of each block's reachable set, while any composition of the stated form is, by construction, of the stated form. (ii) A polynomial filter is additive: $p(\mathbf{A})(\mathbf{b}_1 + \mathbf{b}_2) = p(\mathbf{A})\mathbf{b}_1 + p(\mathbf{A})\mathbf{b}_2$. Take $\mathbf{b}_1, \mathbf{b}_2$ such that the input of the first non-affine N_b (a fixed linear function of the network input up to that block) lands on the non-affine part of N_b for $\mathbf{b}_1 + \mathbf{b}_2$ but not for $\mathbf{b}_1$, $\mathbf{b}_2$ separately — possible whenever N_b is non-affine on the range; then additivity fails downstream, since all subsequent maps are injective on the relevant set for generic θ. The re-estimated case is identical with the map $x_{b-1} \mapsto \mathbf{A}_b(x_{b-1})\,x_{b-1}$ in the role of N_b. □

Proposition 4 (optimality dilemma).

Proof. CG's T-th iterate minimizes the $\mathbf{A}$-norm error over $\mathcal{K}_T(\mathbf{A}, \mathbf{b})$ (Golub & Van Loan, 2013); by Theorem 1 $\mathbf{x}_T^\theta$ lies in that subspace. The dichotomy follows: an improvement on the end task with no improvement on the solve is an improvement obtained by departing from the solve. □

Proposition 5 (conditioning).

Proof. (i) is the chain rule through a map of rank $\le \dim \mathcal{R}$. (ii) Conversion from the Bernstein to the monomial basis is triangular with nonzero diagonal (Lorentz, 1953); linearity in c gives convexity of any convex loss; the conditioning statement is Farouki & Rajan (1987). (iii) On $[0,1]$ the basis functions are nonnegative and sum to one, so $p_c(\lambda)$ is a convex combination of the c_j and lies in $[0, \gamma]$; symmetry of $\mathbf{A}$ transfers this to the operator; $\sum_j (j/d)\binom{d}{j}\lambda^j(1-\lambda)^{d-j} = \mathbb{E}[X]/d = \lambda$ for $X \sim \mathrm{Binomial}(d, \lambda)$. □

Proposition 3 (nonsymmetric operators).

Proof. The induction of Theorem 1 uses only that Φ_t is linear in its vector arguments with $\mathbf{b}$-independent coefficients and that each step applies $\mathbf{A}$ at most once per state vector; symmetry of $\mathbf{A}$ is not used, so $\mathbf{x}_T \in \mathcal{K}_{T+1}(\mathbf{A}, \mathbf{b})$ and $\mathbf{x}_T = p_\theta(\mathbf{A})\mathbf{b}$ for any square $\mathbf{A}$. The semialgebraic argument of Proposition 1 is likewise independent of $\mathbf{A}$. For the optimality statement, GMRES's T-th iterate minimizes $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$ over $\mathbf{x} \in \mathcal{K}_T(\mathbf{A}, \mathbf{b})$ (Saad, 2003), and every fixed-constant iterate lies in that

subspace; the dichotomy follows as in Proposition 4 with the residual norm in place of the $\mathbf{A}$-norm error. □

The nested instance, in closed form. For the nested construction of Example 1 two further facts are available exactly (Hosseini, 2026); we quote them because Section 4 tests them at scale. At the standard initialization (the truncated Taylor series itself) the end-to-end response is $g_K(\lambda) = \lambda/(1-(1-\lambda)^{K+1})$ on $\mathrm{spec}(\mathbf{\Psi}) \subset (0,1]$: increasing, equal to one at $\lambda = 1$, strictly above the intended response λ elsewhere, and bounded below by $1/(K+1)$ as $\lambda \to 0$ — the initialization is a *leaky* denoiser, and the required order for relative accuracy ε is $K+1 \gtrsim \kappa(\mathbf{\Psi})\log(1/\varepsilon)$. For the symmetrically normalized bilateral graphs used in practice, $\kappa(\mathbf{\Psi}) = 1+\mu(1-w_{\min})$ with $w_{\min} < 0$ whenever the window truncates the kernel, so $\kappa > 1+\mu$ independently of image content; at $\mu = 100$ this puts the required K near 500 against the 10 used. Proofs are in Appendix A.

E Coordinates for the polynomial class

Which coordinates (full text of the remark in Section 3.4). The Jacobian favours the additive basis uniformly; the *size* of the coefficients a task requires does not. A bounded, low-pass response has Bernstein coefficients in $[0,1]$; a solve-like response, $p(\lambda) \approx 1/\lambda$ — interpolation, deblurring, in-context regression — has additive coefficients growing like $\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k$ with the degree, while the product form of Proposition 2 keeps every step size in $[1/\lambda_{\max}, 1/\lambda_{\min}]$: the learned steps are the *roots* of the polynomial, the bounded coordinates for that class (Section 4.1). Conjugate-gradient constants are a third coordinate system and the worst: rank-deficient, and unbounded. The additive coordinates alone make the output linear in the parameters, so the class optimum is a solve and the coefficient-scale problem disappears with the optimizer.

F Proof of the class hierarchy (Corollary 1)

The class $\mathcal{C}_{\mathrm{GD}}$ is Proposition 2: $1-\lambda p(\lambda) = \prod_k (1-a_k\lambda)$ with real a_k, hence every root real, and every real-rooted q with $q(0) = 1$ arises. Taking every momentum zero gives $\mathcal{C}_{\mathrm{GD}} \subseteq \mathcal{C}_{\mathrm{HB}}$. Strictness: the heavy-ball recurrence $q_{t+1} = (1+\beta_t-\eta_t\lambda)\,q_t - \beta_t\,q_{t-1}$, $q_0 = 1$, $q_1 = 1-\eta_0\lambda$, with $\eta_0 = \eta_1 = 1$ and $\beta_1 = -3/2$ gives $q_2(\lambda) = 1-\lambda/2+\lambda^2$, whose discriminant is $-15/4$: a conjugate root pair, unreachable by any learned steps. Full dimensionality: the map $(\eta,\beta) \mapsto (q_L$'s L free coefficients$)$ has Jacobian rank L at random points for every $L \le 16$ (checked in float64; the script is in the supplementary code), so $\mathcal{C}_{\mathrm{HB}}$ has nonempty interior in the $q(0) = 1$ family; whether it is all of that family is open. Because the Chebyshev (or Bernstein) coefficients are a basis, $\mathcal{C}_{\mathrm{head}} = \mathcal{P}_L$. Finally, $\mathcal{C}_{\mathrm{CG}}$ is Proposition 1 with Example 1 (rank 12 of 21 at the published setting).

G ADDITIONAL FIGURES AND TABLES

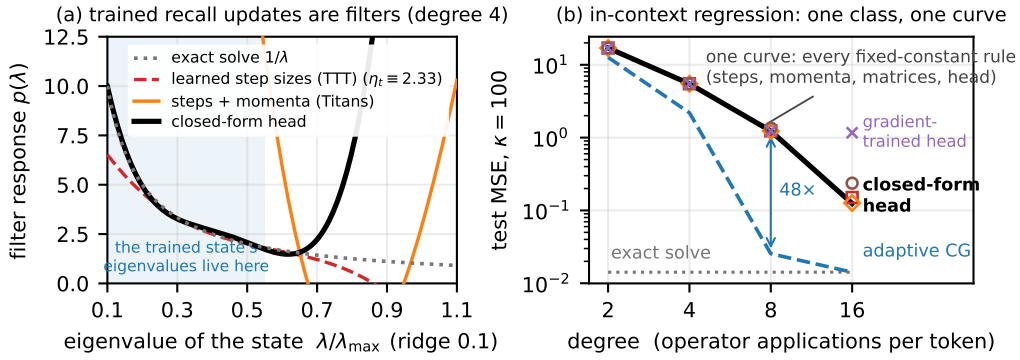


Figure 5: Test-time-training updates are filters, and every fixed-constant rule is one curve. (a) The trained update rules of a recall layer (MQAR, four operator applications) drawn as polynomial filters on the state's spectrum: learned step sizes (TTT; the four steps converged to one value), steps with momenta (Titans) and the closed-form head coincide with the ridge-damped solve $1/\lambda$ where the trained state's eigenvalues live and differ only where none do. (b) In-context regression at condition number 100 against degree (Theorem 1): learned step sizes, momenta, learned $d\times d$ matrices and the gradient-trained head land on one curve; the closed-form head *is* that curve; only conjugate gradient (MesaNet), whose coefficients depend on the context, leaves the class, by $48\times$ at degree 8 (Proposition 4); the gradient-trained head diverges at degree 16 (Proposition 5). Medians of three seeds; Tables 3 and 11.

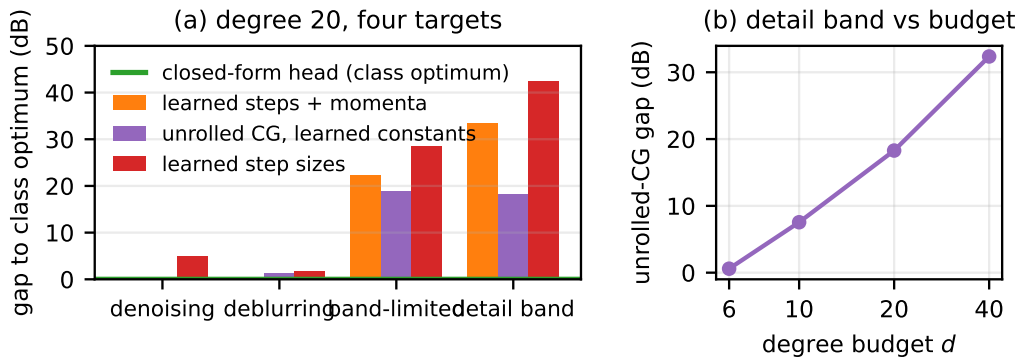


Figure 6: **Where the restriction costs accuracy (E4).** (a) Gap to the class optimum over $\mathcal{P}_{20}$ on a fixed bilateral graph, for the published nested parameterization (unrolled CG with learned constants), learned step sizes, and learned steps with momenta; the closed-form head is the optimum by construction. Low-pass targets cost nothing; a band response costs 18 to 42 dB. (b) The nested gap on the detail band against the degree budget: the reachable set grows more slowly than the class.

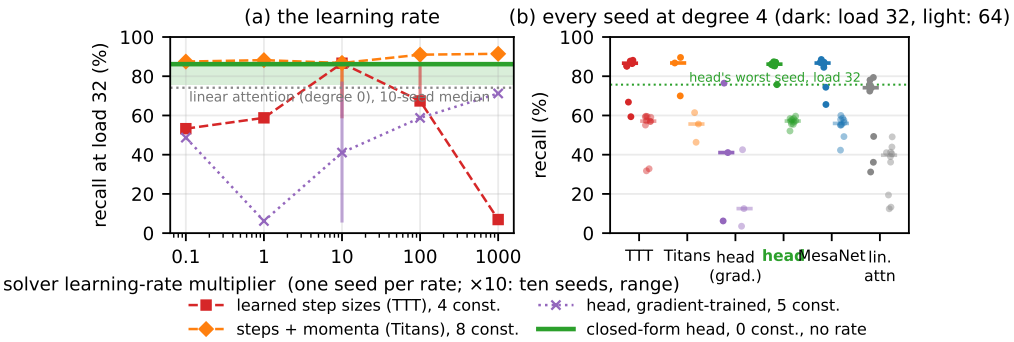


Figure 7: **The knobs a learned solver needs, and the head does not.** Recall at degree 4. (a) Against the solver learning rate, one seed per rate and ten at the shared rate: rules with learned constants move with it, learned step sizes diverging past $100\times$; the head has no rate. (b) Every seed of every rule, loads 32 (dark) and 64 (light): all rules tie at the median; the head's worst seed is $76/52$, learned step sizes fall to $59/32$ on three seeds of ten, linear attention to $31/12$ on five (Table 11).

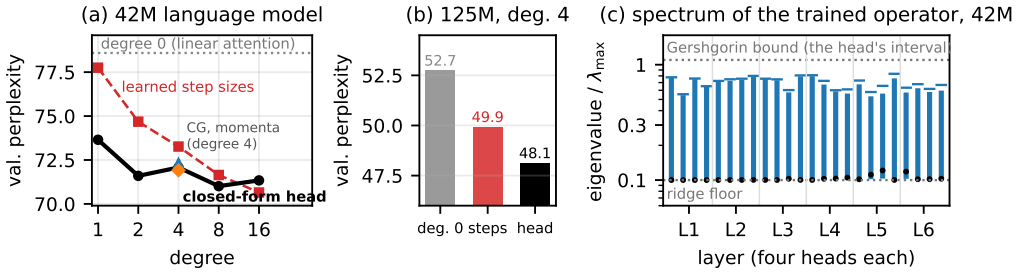


Figure 8: **In a language model, degree moves the perplexity and the rule does not, and the ordering holds at 125M.** (a) A 42M model built from the recall layer: learned step sizes and the closed-form head land within 1.4 at every degree above one, momentum and conjugate gradient sit on the head at degree 4, degree zero trails by six. (b) The same three arms at 125M, degree 4, same token budget: the head is lowest, by 1.8 over learned steps and 4.6 over degree zero (Section 4.2). (c) The measured spectrum of the trained 42M operator (ridged, normalized), per head: 1st to 99th percentile (bar), median (dot), maximum (tick). The median sits at the ridge floor (effective rank below $d/2$) and the top stays below 0.85: condition number under 10, so the best degree-4 polynomial leaves a tenth of the worst direction unsolved and degree 8 under one percent, which is why degree saturates and why the exact solve is a poor filter (Proposition 4).

Table 3: E6. In-context regression test MSE (median over 3 seeds, 4096 test prompts) against depth L and task condition number κ; exact least squares $= 0.014$. Each learned arm uses the best of four learning rates, selected by training loss; "> 10" marks divergence at every rate.

	$\kappa = 1$				$\kappa = 10$				$\kappa = 100$			
L	2	4	8	16	2	4	8	16	2	4	8	16
MesaNet-style CG (L steps, adaptive; no constants)	0.364	0.023	0.014	0.014	2.272	0.227	0.014	0.014	> 10	2.194	0.025	0.014
closed-form head (no learned constants)	0.432	0.040	0.015	0.014	2.846	0.512	0.041	0.014	> 10	5.485	1.233	0.125
TTT-style GD (learned step sizes)	0.432	0.040	0.015	0.014	2.846	0.512	0.041	0.015	> 10	5.490	1.239	0.151
Titans-style GD + momentum (learned)	0.432	0.040	0.015	0.014	2.846	0.512	0.041	0.015	> 10	5.488	1.234	0.127
direct head, gradient-trained (Adam)	0.432	0.040	0.018	0.014	2.846	0.512	0.041	0.024	> 10	5.492	1.232	1.167
linear attention, learned $d \times d$ preconditioners	0.433	0.041	0.015	0.014	2.853	0.515	0.045	0.020	> 10	5.510	1.352	0.237

Table 4: E6, nuisance mode. As Table 3 at κ=100, but the top eigen-direction of each task couples to the context labels with a per-task $\mathcal{N}(0,1)$ coefficient and not to the query label, so the optimal update must notch it (a high-pass response). Test MSE, median of three seeds (min–max in parentheses); label-noise floor 0.01.

	L=4	L=8
exact least squares	68.5	68.5
MesaNet-style CG (L steps, adaptive)	70.6	68.5
linear attention, learned $d\times d$ preconditioners	35.0 (34.8–35.6)	34.1 (32.7–34.3)
TTT-style GD (learned step sizes)	31.4 (31.2–31.8)	32.6 (32.6–32.6)
Titans-style GD + momentum (learned)	31.2 (31.2–31.2)	26.4 (25.9–26.5)
direct head, gradient-trained (Adam)	29.2 (29.0–29.2)	25.2 (25.2–25.4)
closed-form head (no learned constants)	**29.0**	**25.3**

Table 5: E1. Theorem 1 on vectors. Bilateral graphs on 64×64 CBSD68 patches, T=15 steps, fp64; worst case over 100 random draws of the learned constants and 32 inputs. Relative residual after projection onto an orthonormal basis of $\mathcal{K}_{T+1}(\mathbf{A},\mathbf{y})$, and additivity defect $\|\Phi(\mathbf{y}_1+\mathbf{y}_2) - \Phi(\mathbf{y}_1) - \Phi(\mathbf{y}_2)\|/\|\Phi(\mathbf{y}_1+\mathbf{y}_2)\|$. Every learned-constant iteration is a fixed polynomial filter to machine precision; the Lanczos filter lies in the subspace but is not linear in $\mathbf{y}$ (Remark 1).

	Gradient descent	Heavy-ball	CG (learned α,β)	Chebyshev	Lanczos (m=15)
Krylov residual (max)	6×10^{-16}	5×10^{-16}	6×10^{-16}	2×10^{-15}	8×10^{-16}
Additivity defect (max)	2×10^{-16}	3×10^{-16}	4×10^{-16}	3×10^{-15}	1×10^{-2}

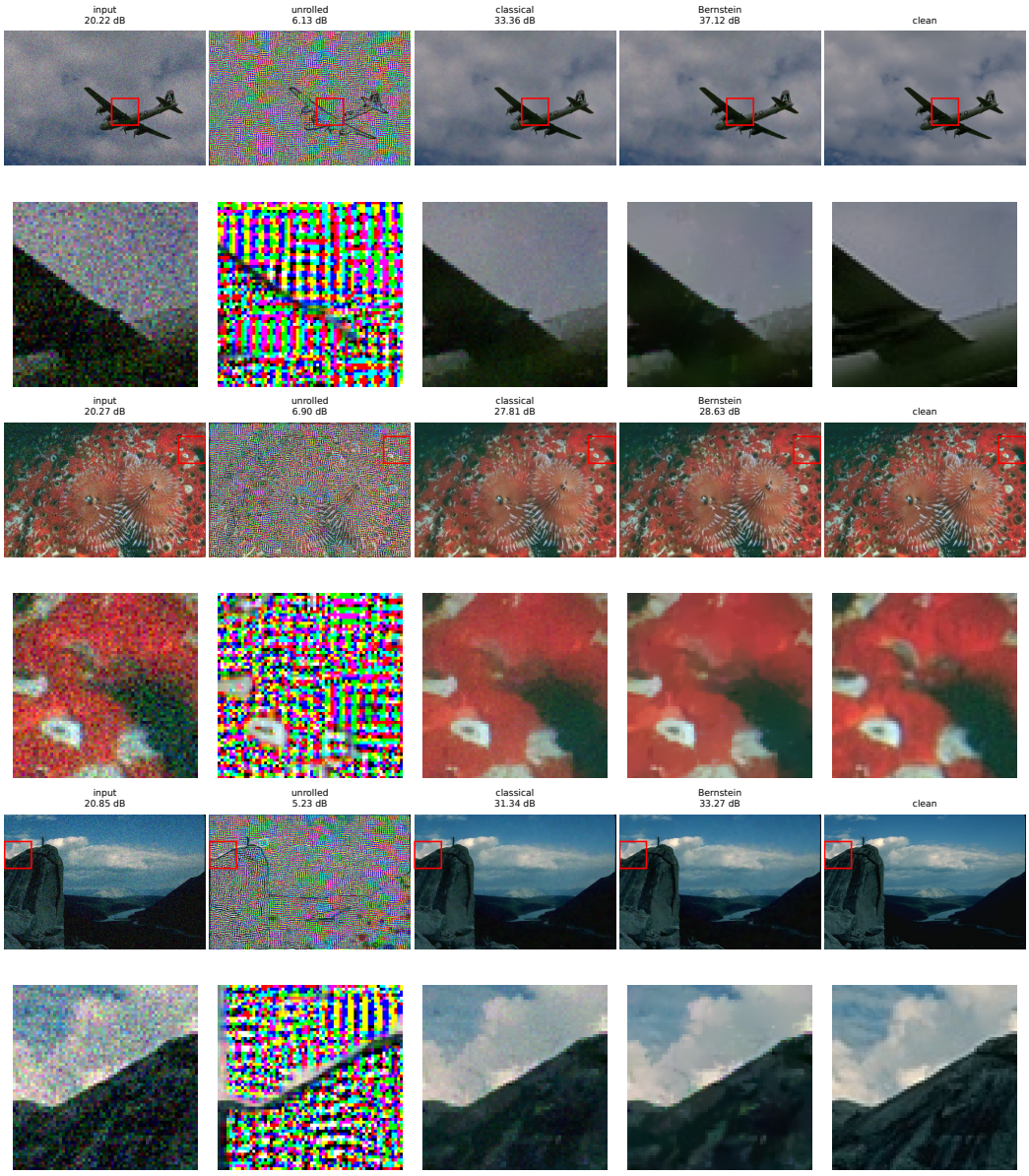


Figure 9: The graph network at σ=25, the three solve arms at matched degree 10 (seed 0), full CBSD68 images with a zoom on the boxed region. Classical CG and the bounded Bernstein filter are close, the Bernstein head sharper by 1–4 dB per image; the unrolled learned-constant arm shows the checkerboard of the highest-frequency Laplacian eigenmode — fixed step sizes amplify modes on samples whose spectrum lies outside the range the constants were fitted for ($|1 - \alpha\lambda| > 1$), the per-sample-operator failure discussed in the text. PSNR of each image in the panel titles.

G.1 E1: THE CEILING IS EXACT

Theorem 1 can be checked to machine precision. For a graph Laplacian on 64×64 patches we run gradient descent, heavy-ball, learned-constant CG, Chebyshev and a Lanczos filter for T steps with random constants and project the output onto the Krylov basis (Table 5): the first four leave a residual at the orthogonalization error (10^{-16}–10^{-15}) and are additive to the same precision — fixed polynomial filters, whatever the constants; the Lanczos filter is in the subspace to 10^{-16} but fails additivity by 10^{-2}, a data-dependent polynomial (Remark 1).

G.2 TABLE FOR E4

Table 6: E4. Spectral-domain fits at matched degree d=20 on a fixed bilateral graph (32×32 patches of CBSD68; signal energy measured from natural patches; median over 3 graphs). The first two rows are *tasks* with a noise term — denoising (σ=0.1) and non-blind deblurring of a graph heat-kernel blur (Wiener objective) — reported as the gap in dB to the optimum over $\mathcal{P}_d$. The last two are prescribed responses — a detail band (spectral-wavelet / multilayer-Laplacian band; on a signed graph) and a near-ideal low-pass projection (bandlimited reconstruction) — reported as approximation PSNR in dB, capped at 60. Bernstein arms: unconstrained; box $c \in [0, 1]$ (non-expansive certificate); gain budget $c \in [0, 2]$. Median ± half-range over the three graphs; the nested arm is the best of eight restarts, with the restart median in parentheses. Lower block: learned step sizes (real-rooted residual, $\mathcal{C}_{\mathrm{GD}}$) and learned steps with momenta ($\mathcal{C}_{\mathrm{HB}}$), d+1 steps each, fitted by the same restart procedure (Corollary 1; `e4_realrooted.py`).

Target	unrolled (nested)	Bernstein	box $[0, 1]$	box $[0, 2]$
denoising task [gap, dB]	0.06 ± 0.01	0	0.01	0.01
deblurring task [gap, dB]	1.37 ± 0.66	0	5.32 ± 1.66	1.23 ± 0.76
detail band, signed graph [PSNR, dB]	41.7 ± 8.3 (29.4 restart-median)	60.0	21.5 ± 4.1	24.2 ± 4.4
bandlimited reconstruction [PSNR, dB]	38.0 ± 0.5 (33.6 restart-median)	56.9 ± 0.3	36.9 ± 1.1	39.8 ± 1.0
detail band, nested gap vs. budget $d = 6/10/20/40$ [dB]	0.6 ± 1.2 / 7.6 ± 1.3 / 18.3 ± 8.1 / 32.4 ± 7.8			

	learned steps ($\mathcal{C}_{\mathrm{GD}}$)	steps + momenta ($\mathcal{C}_{\mathrm{HB}}$)
denoising task [gap, dB]	5.0 ± 1.4	0.12 ± 0.06
deblurring task [gap, dB]	1.8 ± 0.3	0.4 ± 0.4
detail band, signed graph [PSNR, dB]	17.6 ± 3.2	26.4 ± 6.2
bandlimited reconstruction [PSNR, dB]	28.4 ± 0.4	34.5 ± 0.7

G.3 E2: THE FULL RESTORATION TABLE

The class ceiling. On this network every arm trains its own operator learner, so the arms are compared at their own operators; the exact solve of the system the layers declare (30.11 denoising, 28.77 deblurring) is the ceiling of every fixed-constant rule for *that* system (the exact arm was run on seed 0; the seed-2 textbook solver, 30.12, is within 0.01 dB of it, and in deblurring, one run each, textbook CG at 28.83 is 0.06 dB above it), and in denoising every fixed-constant arm's median in Table 7 sits under it. The one arm above it, unrolled CG at ten times the shared rate, is the one whose constants trained jointly with the operator learner: by Theorem 2(ii) the gain is the operator's. The fixed-operator version of this table (operator learner frozen, arms retrained) is the direct test and is left to future work.

Table 7: E2, full version of Table 2. The graph network (three blocks, re-estimated graph, CG-10) and GDD, the arms at matched degree d. PSNR on CBSD68 (denoising, σ=25), $\pm$ spread where several seeds were run; steps to 95% of final PSNR (— where the validation curve peaked and decayed); Jacobian rank / parameter count of the solver block. Deblurring: non-blind, Gaussian blur σ_b=1.6, noise $2.55/255$, each block solving $(\mathbf{H}^\top\mathbf{H} + \mu\mathbf{L})\mathbf{x} = \mathbf{H}^\top\mathbf{y}$; *each learned arm at its own best solver-parameter learning rate (a $\{1, 10, 100\}\times$ grid; the head's coefficients and the CNN weights live on different scales); †the direct head's coefficients fit by least squares after every step (its class optimum given the network; App. B), the network by gradient through the basis; ‡seed 0, with the learned prior weight floored at $\mu \geq 0.03$ for every arm (Appendix H); ranks where the Jacobian was computed. On this network the operator is re-estimated in every block; the least-squares head, exact on fixed-operator blocks (E6, E7), fits the training crops here and decays on full images (Section 4.4).

Model	Arm	d	PSNR	Steps to 95%	Rank / params
graph network, denoising (3 seeds)	exact (converged), seed 0	—	30.11	—	—
	unrolled	10	12.4 ± 9.1 (6.7, 12.4, 25.0)	—	—
	unrolled, 10× rate*	10	30.20 ± 0.09 (30.2, 30.1, 30.3; 0.1×: 6.9, 0.01×: 26.9)	3400	—
	classical	10	29.30 ± 0.66 (28.8, 29.3, 30.1)	4600	—
	least-squares head†	10	20.9 ± 5.7 (18.5, 20.9, 30.0)	—	0 / 0
	Bernstein	10	29.43 ± 0.10 (29.4, 29.4, 29.6)	2400	11 / 11
GDD	unrolled	20	28.10	1100	12 / 17
	classical	20	28.06	1700	—
	Bernstein	20	28.10	900	21 / 21
graph network, deblurring‡	exact (converged)	10	28.77	—	—
	classical CG	10	28.83	—	—
	unrolled, shared rate	10	28.57	—	—
	unrolled, 100× rate	10	25.71	—	—
	Bernstein, shared rate	10	28.29	—	11 / 11
	Bernstein, 10× rate	10	28.47	—	11 / 11
	least-squares head†	10	27.72	—	0 / 0

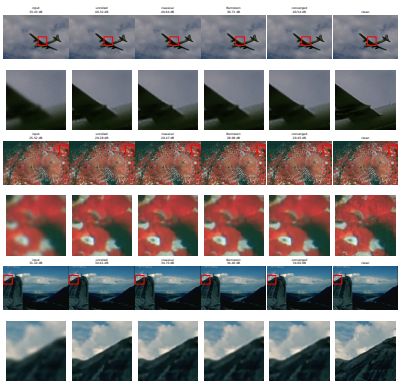


Figure 10: **Solvers are filters.** Non-blind deblurring (Gaussian blur σ_b=1.6, noise $2.55/255$) with the graph network, BSDS300-trained, CBSD68 test images near the set medians; the arms share the network, data, budget and initialization and differ only in the solver block (seed 0, prior kept on). The four arms — learned constants, textbook CG, the bounded head and the converged solve — are visually indistinguishable and within 0.6 dB of one another on every image (40.3–40.6, 29.0–29.5, 34.4–34.7 dB): one class, Theorem 1; the converged solve sits with the truncated ones because the truncation regularizes little once the prior is on.

G.4 E7: THE RECALL TABLE

Table 8 lists the recall of every state update at every load, with the seed ranges in the caption; Figure 7 draws the same runs.

Table 8: E7. MQAR recall (%) against load at T=4, d=64, ridge $10^{-1}\lambda_{\max}$ for every arm (Appendix I); constants per layer; medians over ten seeds (momentum, gradient-trained head and exact solve: three). Seed ranges at loads $32/64$: learned steps 59–88 / 32–60, closed-form head 76–87 / 52–60, CG 66–89 / 42–60, linear attention 31–79 / 12–49.

state update	constants	load 4	8	16	32	64	steps to 95%
linear attention (degree 0)	0	99	99	98	74	40	3000
delta rule (degree 1, learned step)	1	100	99	99	79	44	2375
TTT-style GD (learned steps)	4	100	100	100	87	57	1500
Titans-style GD + momentum	8	100	100	100	87	56	1250
MesaNet-style CG (adaptive)	0	100	100	100	87	56	1750
closed-form head	**0**	100	100	100	**86**	**57**	1500
direct head, gradient-trained	5	97	93	84	41	13	2750
exact solve	0	100	100	100	71	48	1500

G.5 E7: SENSITIVITY TO THE SOLVER LEARNING RATE

Figure 7(a) in numbers. The E7 regime at T=4, seed 0 (ten seeds at the shared multiplier $10\times$, marked *), recall (%) at loads $32/64$ against the multiplier on the learning rate of the solver constants; the closed-form head has no such rate and gives $86/57$ (ten-seed range 76–87 / 52–60).

solver learning-rate multiplier	0.1	1	10^*	100^*	1000
learned step sizes (TTT)	53/33	59/36	67/33	67/46	7/5
learned steps + momenta (Titans)	87/60	88/58	87/56	91/64	91/65
head, gradient-trained	49/24	6/3	41/13	59/31	71/43
closed-form head (no rate)			85/55		

G.6 E7D: LENGTH GENERALIZATION

Trained at length 256 and load 16 with no absolute positional embedding (the causal convolution supplies relative positions; vocabulary 256 so that load 128 has distinct keys; otherwise the E7 protocol, T=4, ridge $10^{-1}\lambda_{\max}$, seed 0), every arm is evaluated at length 1024 (Table 9). Every arm transfers to four times the training length without loss at loads 16–64; every arm drops to 43–52% at load 128, which exceeds the 64-dimensional state. The state size is the limit. In this regime (twice the keys of the main table, so lower interference at a given load) linear attention sits closer to the degree-4 arms than in Table 8: degree buys most where interference is highest.

Table 9: E7d. MQAR recall (%) at the training length (256) and at $4\times$ the training length (1024), trained at load 16.

	length 256, load			length 1024, load			
state update	16	32	64	16	32	64	128
linear attention (degree 0)	99	91	61	96	91	77	52
TTT-style GD (learned steps)	100	96	68	100	97	79	48
Titans-style GD + momentum	100	97	71	100	98	82	52
MesaNet-style CG (adaptive)	100	93	58	100	96	76	43
closed-form head	100	96	67	100	97	80	48

G.7 E9: THE ARMS INSIDE A SMALL LANGUAGE MODEL

The E7 block, made multi-head and stacked into a causal language model: 6 layers of [causal convolution, 4 solver heads of d=64, MLP], width 512, tied embeddings, 42.1M parameters; GPT-2 tokens of WikiText-103, sequence length 1024, batch 8, 8000 steps (65M tokens), AdamW at 6×10^{-4} with cosine decay ($10\times$ for solver constants), ridge $10^{-1}\lambda_{\max}$, T=4, seed 0, one RTX 4090; validation perplexity on the held-out split (Table 10). At this budget the four degree-4 update rules

land within 1.4 perplexity of each other and degree zero trails the best by 6.7; the closed-form head, with no learned solver constants and no non-finite step, sits between momentum and textbook CG, and the three gradient-free curves (CG, head, momentum) lie on top of each other throughout training. One seed and one budget: the row is a believability check, not a scaling result. At 125M (12 layers, width 768, 12 heads of d=64, the same tokens, batch and schedule, gradient accumulation 4 to 8), the three arms we could afford rank in the same order with the head lowest: linear attention 52.74, learned step sizes 49.92, closed-form head 48.12 (Table 10).

Table 10: E9. Validation perplexity of 42M- and 125M-parameter language models whose sequence layer is the E7 solver block, one arm per row, at a fixed 65M-token budget (WikiText-103, seed 0).

state update	learned solver constants	val. perplexity
linear attention (degree 0)	0	78.59
TTT-style GD (learned steps)	96	73.26
Titans-style GD + momentum	192	71.91
MesaNet-style CG (adaptive)	0	72.32
closed-form head	0	72.07
125*M parameters, same budget*		
linear attention (degree 0)	0	52.74
TTT-style GD (learned steps)	576	49.92
closed-form head	0	48.12

G.8 Performance versus degree

Theorem 1 makes depth a degree. Table 11 reads the E7 recall task and the E9 language model along that axis, one arm per column, with a spectral column: the scale-invariant unsolved fraction $\tilde{\rho} = \min_s \max_x |1 - s\,x\,p(x)|$ of the layer's fitted polynomial over the empirical spectrum band $[q_{0.01}, q_{0.99}]$ of its regularized operator (the recall readout is an argmax, so only the shape of the filter matters; $\tilde{\rho} = 0$ is the exact solve, $1 - x$ at degree zero). Three things are visible, now over ten seeds. (i) Degree pays, for every arm, up to degree 4: the head's $\tilde{\rho}$ falls from 0.42 to 0.07 between degrees 1 and 4 and its recall rises from 83/55 to 86/57; learned step sizes rise from 79/44 to 87/57; CG is at 86/56 from degree 2. Beyond degree 4 the medians move by at most two points, at every rule: the interval of Figure 8c is covered. (ii) At the median the rules tie at every degree from 4 up; what separates them is the seed distribution, the worst seeds being 76–80 for the head at degrees 4–8 against 59–70 for learned step sizes and 65–66 for CG, and 10 for momentum at degree 16, Proposition 5 as seed variance. An earlier three-seed version of this table showed the solve-targeting rules dropping to the exact solve's 71/48 at degree 16; with ten seeds that drop is absent, and the exact solve's own three seeds span 59–89 at load 32, so E7 does not separate the exact solve from the truncated ones; the evidence for Horn 2 of Proposition 4 on this task is the low-ridge run of Figure 11, one seed, and E4 and E6. (iii) In the language model every degree helps learned step sizes (77.8 to 70.6 perplexity from degree 1 to 16); the head is ahead at every degree but 16, reaches its best value, 71.0, at degree 8, and never takes a non-finite step at any degree. Momentum never solves ($\tilde{\rho} \geq 0.4$ at every degree) and recalls at 87–88 at degrees 4–8: the solve is not the task.

Table 11: Performance versus degree. MQAR recall (%) at loads 32/64, median over ten seeds with the range at load 32 in brackets (momentum: three seeds), E9 validation perplexity of the 42M language model, and the unsolved fraction $\tilde{\rho}$ (median of three probed seeds), for the closed-form head, learned step sizes (TTT), textbook CG (MesaNet) and learned steps with momenta (Titans), at matched degree. Degree 0 is linear attention (74/40 [31–79], perplexity 78.59); the exact ridge solve recalls 71/48 [59–89] over three seeds.

	recall 32/64 [range at 32]				perplexity		$\tilde{\rho}$		
degree	head	steps	CG	momentum	head	steps	head	steps	momentum
1	83/55 [57–88]	79/44 [66–82]	—	70/33 [64–78]	73.65	77.75	0.42	—	0.73
2	84/53 [58–86]	84/56 [55–87]	86/56 [56–88]	82/49 [69–83]	71.60	74.68	0.31	0.40	0.39
4	86/57 [76–87]	87/57 [59–88]	87/56 [66–89]	87/56 [70–90]	72.07	73.26	0.07	0.21	0.52
8	87/58 [80–89]	87/58 [70–88]	86/56 [65–89]	88/55 [78–90]	71.01	71.65	0.01	0.09	0.56
16	86/57 [61–89]	87/58 [75–89]	86/56 [60–89]	76/43 [10–90]	71.34	70.65	0.02	0.01	0.42

H THE TWO BASINS OF THE DEBLURRING TASK

The arms share the network, the data, the budget (d=10 operator applications per block) and the calibration, and differ only in the solver block. In the DeepGLR-style block of the previous version of this study two basins existed for every arm: one with the learned prior on ($\mu \approx 0.08$), where the arms tied, and one where training drove μ to its floor, the system degenerated to $\mathbf{H}^\top \mathbf{H}$ and every arm lost 1.2 dB — the trivial-system escape of Proposition 4 as a basin. The graph network is trained with the prior floored at $\mu \geq 0.03$ for every arm; on seed 0 no arm leaves the prior-on basin (every stable arm lands at 27.7–28.8 dB, Table 7); further seeds are reported in the final version.

I EXPERIMENTAL DETAILS

Restoration network (E2, Figure 10). Training crops are drawn from the BSDS300 training set (200 images; the E5 survey uses the same); test PSNR is on CBSD68 at full size (RGB; GDD in grayscale, as published). Noise is additive Gaussian at $\sigma \in \{15, 25, 50\}/255$; interpolation uses a 2× regular grid mask; deblurring uses a 15×15 Gaussian kernel with σ_b=1.6 and noise $2.55/255$. Budgets are fixed per model and identical across arms, and are shorter than the published training schedules (which run to 10^5–10^6 iterations on larger corpora), so absolute PSNRs sit below converged schedules, and every claim is relative between arms at equal budget: the graph network (3 blocks, 10 CG steps, 48-px patches, batch 16, 8000 steps, for denoising and deblurring alike; four graph heads per block, 12-dimensional features, 3×3 window), GDD (K=10, m=3, batch 32, 3000 steps, learning rate 10^{-3}); Adam with learning rate 10^{-4} unless stated, cosine schedule, gradient-norm clip 1, validation every 100 steps on a held-out crop set, one seed per row unless a spread is given. Every arm is *calibrated* on one batch before training: the direct heads are initialized at the Chebyshev (closed form) or Bernstein (gradient-trained) coefficients of the T-step CG polynomial on that batch, so all arms start at the same filter. For deblurring the solver-parameter learning rate is chosen per arm from a $\{1, 10, 100\}\times$ grid (10× for the Bernstein head, 100× for the unrolled constants), and for denoising the unrolled constants get a $\{0.01, 0.1, 1, 10\}\times$ grid; the closed-form head has no learning rate: its coefficients are re-fit after every step by least squares with the normal equations accumulated as exponential moving averages (ρ=0.99) in float64, diagonally scaled, and solved by a truncated pseudo-inverse (relative cutoff 10^{-6}). Steps-to-95% is the first validation step at which the PSNR gain reaches 95% of its final value. The graph learner is a small CNN per head emitting features whose Mahalanobis distances set the edge weights on a constant-offset stencil, row-normalized, with the head weights and the prior weight μ learned; GDD's is the published one. Restoration runs take 30–90 minutes each on one RTX 3090 and share the GPU with other runs, so wall-clock is not reported.

In-context regression, nuisance mode (E6). As E6, with the true weight projected off the top eigen-direction of each task's covariance and a spurious coupling $c\,(\mathbf{u}^\top \mathbf{x})$, $c \sim \mathcal{N}(0, 1)$ per task, added to the n=32 context labels only; κ=100, three seeds, the same learning-rate grid, arms and normalization as E6 (`icl_krylov.py --nuisance 1.0`). *Released checkpoint.* The published uGTV checkpoint (Do et al., 2024) (epoch 29 of the released training run), loaded into our stencil re-implementation (which reproduces the released code to 5×10^{-7}), gives 38.2 dB on six Urban100 images (Bayer demosaicking, 768-px crops) with its learned (α_t, β_t); with those constants replaced by textbook conjugate gradient on the same operator and the same ten steps, 26.5 dB. The trained constants and the surrounding weights have co-adapted into a filter that is not the solve of the declared system, the trained-state reading of Section 5.

E4 and E5 (spectral domain). Bilateral graphs are built on 32×32 (E4) and 64×64 (E5) grayscale patches with features $(x, y, \text{intensity}, \nabla_x, \nabla_y)$ scaled by $(1/3, 1/3, 10, 5, 5)$, radius 1 or 3, symmetric normalization, and a dense eigendecomposition; fits are on the spectrum with the natural-patch signal energy as weight. E4 uses three graphs, degree d=20, Adam for the nested arm with eight restarts. E5 (Table 12) measures the nested instance of Appendix A across two datasets (BSDS300 train, CBSD68), $\mu \in \{10, 100, 1000\}$, both radii, and $(K, m) \in \{(6, 3), (10, 3), (10, 5), (20, 3)\}$: the condition number and its bound, the initialization response at $\lambda_{\min}$ against its floor, the Jacobian rank, and a spectral-domain ablation in which the unrolled and Bernstein constants are trained directly on 24 train patches per (dataset, σ) and tested on 24 others (K=10, m=3, radius 3, μ=100; 3000 Adam steps, learning rate from $\{0.05, 0.2\}$, 3 seeds).

Table 12: E5. The nested instance across settings. Top: condition number of $\mathbf{I} + \mu\mathbf{L}$ (median over patches), required order K_{req} for 10^{-2} accuracy, and the initialization response $g_K(\lambda_{\min})$ against its floor, per (radius, μ) (the two datasets agree to three digits and are pooled). Middle: numerical rank of the constants-to-response Jacobian at initialization. Bottom: spectral-domain ablation, test PSNR (median of 3 seeds) of the unrolled and Bernstein parameterizations at matched degree 20.

radius, μ	κ	K_{req}	$g_{10}(\lambda_{\min})$ / floor	$g_{20}(\lambda_{\min})$ / floor
1, 10	13.8	60	0.129 / 0.091	0.091 / 0.048
1, 100	129	589	0.094 / 0.091	0.051 / 0.048
1, 1000	1277	5879	0.091 / 0.091	0.048 / 0.048
3, 10	11.6	50	0.137 / 0.091	0.102 / 0.048
3, 100	107	488	0.095 / 0.091	0.052 / 0.048
3, 1000	1057	4865	0.091 / 0.091	0.048 / 0.048
(K, m)	(6, 3)	(10, 3)	(10, 5)	(20, 3)
rank / constants	8 / 13	12 / 17	19 / 21	22 / 27
ablation, σ=15/25/50	BSDS300: unrolled 29.47 / 27.31 / 24.98 Bernstein 30.33 / 28.26 / 25.15		CBSD68: unrolled 28.76 / 26.55 / 24.15 Bernstein 29.64 / 27.47 / 24.25	

E7 (MQAR). Sequences of length 256 over a vocabulary of 128, 16 key–value pairs presented and queried at training time, evaluated at loads 4–64 with the sequence length fixed. The model is two layers of [causal depthwise convolution (kernel 4) $\rightarrow$ projections to d=64 keys, values and queries $\rightarrow$ solver layer $\rightarrow$ MLP], model width 128, residual connections and layer norm. The solver layer keeps the prefix sums $\mathbf{S}_t = \sum_{s \le t} \mathbf{k}_s \mathbf{k}_s^\top$ and $\mathbf{U}_t = \sum_{s \le t} \mathbf{k}_s \mathbf{v}_s^\top$, regularizes $\mathbf{S}_t$ by a ridge of $0.1\,\lambda_{\max}$ (validated on the training load against 10^{-3}; every arm uses the same ridge), and reads out $\mathbf{q}_t^\top p(\mathbf{S}_t)\mathbf{U}_t$ with T applications of $\mathbf{S}_t$; the arms differ only in p: learned steps (GD), learned steps and momenta (heavy-ball), textbook CG on the vector system, the closed-form head (Chebyshev coefficients re-fit by least squares after every step), its gradient-trained variant, degree zero (linear attention) and the exact solve. Adam, learning rate 10^{-3} ($10\times$ for solver constants), batch 32, 6000 steps, ten seeds for the head, learned steps, CG and linear attention, three for the rest; medians reported. Seed ranges at loads $32/64$: learned steps 59–88 / 32–60 (median $87/57$), gradient-trained head 6–76 / 4–43 (median $41/13$), closed-form head 76–87 / 52–60 (median $86/57$), CG 66–89 / 42–60 (median $87/56$), linear attention 31–79 / 12–49 (median $74/40$). At the lower ridge $10^{-3}\lambda_{\max}$ (seed 0): CG $57/17$, closed-form head $57/35$, learned steps $89/59$, momentum $90/62$, exact $81/47$ (Figure 11). Forward wall-clock at batch 32, length 256, one RTX 3090, T=4: GD 35 ms, momentum 38, CG 40, head 46, exact solve 4–5$\times$ GD.

E6 (in-context regression). Dimension d=8, n=32 examples per prompt, $\mathbf{w}^\star \sim \mathcal{N}(0, \mathbf{I})$, inputs $\mathcal{N}(0, \mathbf{\Sigma})$ with a random rotation and eigenvalues log-uniform in $[1, \kappa]$, $\kappa \in \{1, 10, 100\}$, label noise 0.1. Depths $L \in \{2, 4, 8, 16\}$; every arm normalized per prompt by $\lambda_{\max}(\hat{\mathbf{\Sigma}})$; Adam, 3000 steps, batch 256, the best of a learning-rate grid $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$ per arm, three seeds, test MSE on 4096 fresh prompts.

E1. Bilateral graphs on 64×64 patches; each solver is run for $T \le 20$ steps with constants drawn uniformly at random, and the output is projected onto an orthonormalized Krylov basis in float64; additivity is tested on random pairs of inputs.

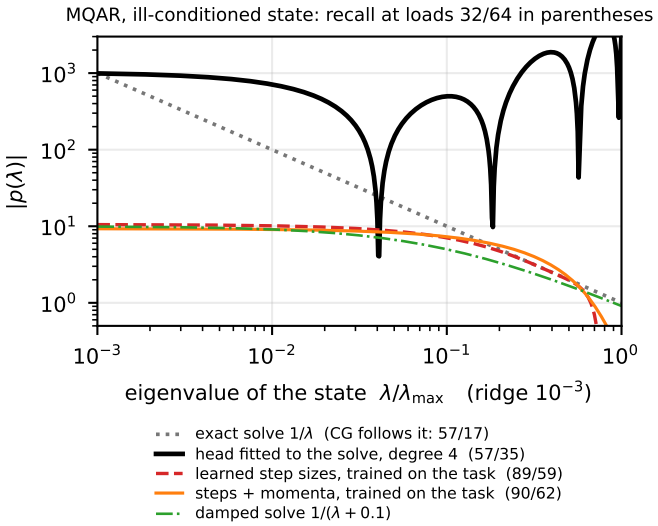


Figure 11: **The truncated-series failure, and what the task does about it.** The trained degree-4 update rules of E7 drawn as filters when the state is ill-conditioned (ridge $10^{-3}\lambda_{\max}$, condition number about 10^3; seed 0, recall at loads $32/64$ in parentheses). The closed-form head is fitted to the *solve*: a degree-4 polynomial cannot follow $1/\lambda$ over three decades, its coefficients reach 10^3 (Proposition 5) and it oscillates across the band where the informative eigenvalues live; conjugate gradient follows $1/\lambda$ exactly and does worse still. The rules trained on the *task*, learned step sizes and momenta, choose the damped solve $1/(\lambda + 0.1)$ almost exactly, an implicit ridge of 0.1 that neither was given: Horn 2 of Proposition 4. In restoration, where the solve is the answer, the same truncation is a failure; in the layer it is the regularizer, and the fix for the head is the ridge every rule in Table 8 is run with.