BLOCKWISE STABILIZED ADAPTIVE CUBIC REGULARIZATION WITH TRUST-CONTROLLED φ_1 SUBSOLVERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Overparameterized networks often have ill-conditioned, saddle rich loss landscapes. They are usually trained with first-order methods that struggle on those landscapes, while many second-order methods stay infeasible at scale. Dense Hessian variants in the cubic Newton family stop scaling near 15k parameters. Some scalable cubic methods replace the Hessian by a cheap approximation and drop the exact block curvature. This paper introduces ARC-block, a blockwise adaptive regularization with cubics (ARC) optimizer. For each parameter tensor it minimizes a cubic model over the true block Hessian, with a per-block adaptive cubic constant and a monotone guard on the full loss. Large tensors are handled matrix-free in a Lanczos-built Krylov subspace. ARC-block's CubicKrylov subsolver's descent is monotone with a per-block cubic decrease bound. For exact steps, and for Krylov steps that meet the Cartis–Gould–Toint residual test, we provide an $\mathcal{O}(\sqrt{B}\varepsilon^{-3/2})$ sweep bound above a cross-block coupling floor. A second subsolver, ARC-φ_1, uses a trust-controlled exponential relaxation step: Newton on high curvature modes, gradient descent on flat modes, and a clamped exponential escape on negative modes, at the same per-trial cost. Among the cubic Newton methods evaluated here, these are the only ones whose steps stay exact on every parameter block (within the Krylov subspace), including on a 91.4M-parameter implicit neural representation (INR). Stopped after 400 sweeps without a 0.05 dB gain on FINER 2D image-fitting INR network, ARC-φ_1 reaches 133.5 dB peak signal-to-noise ratio (PSNR), with no post peak collapse. Tuned Adam peaks at 78.2 dB at 2,629 s (step $\sim$625k, about 44 minutes), further steps degrade its PSNR. ARC-φ_1 reaches that 78.2 dB sooner, at 1,927 s, and by Adam's peak time is already at 84.2 dB. By Adam's 10^6-step budget (about 70 minutes) ARC-φ_1 reaches 95.6 dB. On the same benchmark, SOAP, L-BFGS, and the scalable diagonal cubic method AdaCubic stay at or below Adam's 78.2 dB peak. Over the benchmark's 20 images, run to a plateau rule, ARC-φ_1's per-image PSNR has median 132.4 and mean 131.5 ± 3.3 dB against tuned Adam's 72.8 ± 4.5 dB. The paper provides the FINER network loss landscape analysis explaining the behavior of these optimizers.

1 INTRODUCTION

First-order optimizers (Adam, AdamW, Muon: Kingma and Ba, 2015; Loshchilov and Hutter, 2019; Jordan et al., 2024) use the gradient alone, with no curvature, and therefore exhibit *spectral bias*. Low frequency components of the target are fit quickly, while high frequency components lag by orders of magnitude (Rahaman et al., 2019). In neural tangent kernel terms, error along each kernel eigenmode decays at a rate proportional to its eigenvalue, so the kernel spectrum determines the order in which frequencies converge (Jacot et al., 2018; Lee et al., 2019) (Appendix P). This work develops a second-order optimizer that mitigates spectral bias in training implicit neural representations (INRs), although it can be applied to other kinds of neural networks.

Spectral bias mitigation. The INR literature describes approaches to mitigate spectral bias architecturally, through Fourier feature embeddings (Tancik et al., 2020), wavelet activations (WIRE: Saragadam et al., 2023), variable periodic activations (FINER: Liu et al., 2024), and embeddings whose

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.

initial spectrum is aligned with the target's (Kania et al., 2025), and via custom training through diagonal preconditioners (Chng et al., 2025), tangent kernel gradient transformations (Shi et al., 2025), and stochastic preconditioning (Ling et al., 2025). Most use first-order or diagonal curvature models, as does AdaCubic (Tsingalis et al., 2026), a scalable cubic regularized optimizer with a Hutchinson diagonal Hessian approximation.

Comparative analysis of Adam's performance. Adam (Kingma and Ba, 2015) trains many networks across problem domains, image processing INRs included, with success, and their authors often report that the more computationally expensive second-order methods cannot compete with it. Our results on networks such as FINER (Liu et al., 2024), whose loss landscape is more complex, show otherwise. One line of research on the reasons for Adam's performance attributes its advantage over stochastic gradient descent (SGD) to coordinate-aligned landscape structure (Xie et al., 2024; Zhang et al., 2024; Dong et al., 2025; Jiang et al., 2023; Cohen et al., 2022; Zhang et al., 2025), and on quadratics Das et al. (2024) show that Adam performs well on near diagonal Hessians and can lose to gradient descent for non-diagonal ones. Section 6.2 relates our benchmark results to this line of work.

Cubic Newton methods are often not scalable. Cubic regularized Newton methods (Griewank, 1981; Nesterov and Polyak, 2006), including the adaptive regularization with cubics (ARC) of Cartis et al. (2011a;b), achieve the optimal $\mathcal{O}(\varepsilon^{-3/2})$ rate, finding approximate stationary points of non-convex objectives, but each step usually needs a full eigendecomposition or factorization at $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ storage (Nesterov and Polyak, 2006; Cartis et al., 2011a; Doikov and Grapiglia, 2023; Doikov, 2021). The refinements of this family (Doikov et al., 2023; Doikov and Grapiglia, 2023; Doikov et al., 2024; Mishchenko, 2023; Agafonov et al., 2024; Chayti et al., 2025; 2024) mitigate this cubic cost through subspace methods, low-rank approximations, or approximate Hessians, while keeping a full space linear solve as the basic step, and are often validated at only $n \lesssim 10^3$ scale. In our implementations and on our hardware, all of them and the subspace cubic Newton SSCN (Zhao et al., 2025) remained impractical beyond roughly 15k–20k parameters (Appendix M). The original ARC with its Lanczos subsolver (Cartis et al., 2011a) is matrix-free, yet it stalls at 4.0 dB peak signal-to-noise ratio (PSNR) on the 91.4M ViSIR while placing first on the 15k SIREN-like replica (Sitzmann et al., 2020), evidence that a Krylov subspace subsolver without the blockwise organization may not scale.

Inspiration for ARC-block: Chebyshev second kind acceleration and related work. ARC-block descends from the second-order Chebyshev second kind accelerated optimizers such as the scaled up Dynamic Spectral Optimizer (DSO) of Podorozhny (2026), whose construction is restated in full in Appendix C. The ARC-block outer scheme admits that recurrence as a subsolver (CubicCheby-DSO, Appendix N), and the Lanczos basis is the preferred one for the reasons of Section 3.2. The Krylov cubic step of Gould et al. (1999); Cartis et al. (2011a); Carmon and Duchi (2018); Jiang et al. (2024) reaches a similar subsolver in the trust region line of research. Most Chebyshev accelerated optimizers are first-order. The schemes surveyed by d'Aspremont et al. (2021) and the polynomial preconditioning of Doikov and Rodomanov (2023) are gradient recurrences with no explicit curvature. One of the few second-order Chebyshev optimizers is the spectral preconditioning of Doikov et al. (2024), which keeps the top τ Hessian eigenvectors $V_k \in \mathbb{R}^{n\times\tau}$ (τn floats), refreshes them each iteration by T_k orthogonal iteration steps (τ HVPs and an $\mathcal{O}(\tau^2 n)$ QR each), and applies the preconditioner through the Woodbury identity at $\mathcal{O}(\tau^2 n)$. DSO instead applies Chebyshev polynomials of the *second kind* U_n through a three term recurrence of Hessian–vector products (HVPs), L HVPs and three vectors per block with no stored matrix, with a worst case residual of $\approx$0.22 per sweep at degree $L \approx 1.28\sqrt{\kappa}$ (Appendix C). The recurrence replaces the eigenvector refresh and the Woodbury solve: $\mathcal{O}(Ln)$ computational time and $3n$ storage per sweep against $\mathcal{O}(\tau^2 n)$ and τn per iteration for the Chebyshev first kind optimizer, a factor L against τ^2 and 3 against τ. The larger difference is in the step. The rank τ preconditioner inverts the Hessian on τ directions and takes a gradient step on the other $n-\tau$, while the degree L polynomial contracts every direction in the spectral interval. DSO trains the 91.4M parameter ViSIR (Appendix N), and on the 15k SIREN-like replica, whose spectrum has no drop after its first ten eigenvalues, the first kind optimizer at $\tau = 10$ saturates at 27.7 ± 3.3 dB against 30.0–33.1 dB for DSO (Tables 28, 30). The drawbacks are that the polynomial degree L grows with κ and can become too large, especially when the estimate of the Hessian condition number is inaccurate, and that the magnitude of the search step on indefinite curvature needs clamping, because U_n grows hyperbolically for $\lambda < 0$ (the mechanism of its saddle escape). In ARC-block, the cubic shift mitigates both of these drawbacks (Section 3).

Approach. The proposed ARC-block outer scheme is blockwise adaptive cubic regularization. Parameters are partitioned by tensor, each block minimizes a cubic model with its own adapted cubic constant M_b, and each block step is accepted against a monotone guard on the full loss (Proposition A.10). The subsolver is interchangeable. The cubic shift $\lambda^\star = \frac{M}{2}\|s\|$ bounds the effective condition number of the shifted Hessian operator (Proposition A.3) and makes it positive semidefinite *before* the polynomial processes it (Lemma A.2). The full space cubic step has the global $\mathcal{O}(\varepsilon^{-3/2})$ rate to approximate second-order stationary points (Theorem 3.1, Corollary A.9). The blockwise sweep keeps the exponent, $\mathcal{O}(\sqrt{B}\,\varepsilon^{-3/2})$ sweeps down to a coupling floor determined by the off-block Hessian, with curvature certified per block on its Krylov subspace (Theorem B.10). A second subsolver keeps the outer scheme and replaces the cubic minimizer by the exponential relaxation (φ_1) step: Newton-like on high curvature modes, gradient-like on flat ones, an exponential escape on negative ones (Section 5).

1.1 Contributions

- **A blockwise outer scheme with per-block adaptive regularization (ARC-block).** Each parameter tensor has its own regularization constant, ratio test, and acceptance decision against a monotone guard on the full loss (Algorithm 1, Proposition A.10).
- **Proofs.** Proposition A.3 bounds the subsolver's effective condition number by the cubic shift. Proposition A.10 gives the per-block cubic decrease, the Cartis–Gould–Toint acceptance argument applied per block. Theorems B.8–B.10 give the Phase A, Phase B, and joint sweep rates $\mathcal{O}(\sqrt{B}\,\varepsilon^{-3/2})$ down to a measurable coupling floor (Appendix B). The full space results are restated with proofs in our notation (Lemma A.4 for the Gould–Lucidi–Roma–Toint reduction, Theorem 3.1 and Corollary A.9 for Nesterov and Polyak's rate). The algebraic cores of the blockwise, CubicKrylov, and φ_1 arguments were checked by the Lean 4 kernel (Appendix F), with the cited arguments and Assumption 2.1 as hypotheses.
- **A trust-controlled φ_1 subsolver (ARC-φ_1).** The cubic minimizer is replaced by the exponential relaxation (φ_1) step rule (Proposition 5.1): Newton on high curvature modes, gradient descent on flat modes, and a clamped exponential escape along negative modes (Lemma A.11, Section 5).
- **Empirical results and a dispatch rule.** The 91.4M parameter ViSIR network is trained with the exact block Hessian through HVPs (Section 6). On FINER, ARC-φ_1, run to its plateau, reaches 133.5 dB against tuned Adam's 78.2 dB on the reference image and a mean 131.5 ± 3.3 against 72.8 ± 4.5 dB over 20 images (Section 6.3), and the landscape fingerprint (Section 6.2) distinguishes axis-aligned ill-conditioning, where tuned first-order methods perform well, from cross-coupled or saddle dominated landscapes, where the blockwise cubic step's cost is justified.

2 Background

This section fixes the notation and recalls the cubic regularized step that ARC-block applies per block.

2.1 Problem setting

We minimize a twice-differentiable, generally non-convex training loss

$$\min_{\theta\in\mathbb{R}^n} f(\theta), \qquad f(\theta) = \frac{1}{N}\sum_{i=1}^{N} \tfrac{1}{2}\,\|\Phi_\theta(x_i) - y_i\|^2\,, \tag{1}$$

where Φ_θ is a neural network with weights θ. We write $g = \nabla f(\theta)$, $\mathbf{H} = \nabla^2 f(\theta)$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ and orthonormal eigenvectors $u_1, \ldots, u_n$. Following Doikov et al. (2024) we assume:

Assumption 2.1 (Lipschitz–Hessian)**.** There exists $L_H \geq 0$ such that $\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| \leq L_H\,\|x - y\|$ for all x, y.

Assumption 2.1 yields the standard cubic upper bound

$$f(x+s) \;\leq\; f(x) + \langle g,\, s\rangle + \tfrac{1}{2}\,\langle \mathbf{H}s,\, s\rangle + \tfrac{L_H}{6}\,\|s\|^3\,. \tag{2}$$

2.2 Cubic regularization as a shifted linear solve

The cubic regularized Newton step (Nesterov and Polyak, 2006) minimizes

$$m(s) \;=\; \langle g,\, s\rangle + \tfrac{1}{2}\,\langle \mathbf{H}s,\, s\rangle + \tfrac{M}{6}\,\|s\|^3 . \tag{3}$$

Its global minimizer is characterized (Lemma A.1) by the stationarity system

$$\Big(\mathbf{H} + \tfrac{Mr}{2}\,\mathbf{I}\Big)s = -g, \qquad r = \|s\|\,, \qquad \mathbf{H} + \tfrac{Mr}{2}\mathbf{I} \succeq 0, \tag{4}$$

a damped Newton solve whose damping $\lambda^\star = Mr/2$ is determined by a one-dimensional secular equation. Usually, the solver diagonalizes $\mathbf{H}$ (Nesterov and Polyak, 2006; Doikov and Grapiglia, 2023). On large blocks we build a Lanczos basis instead (Section 3.2). The Krylov cubic Newton method of Jiang et al. (2024) is a different approach, one Krylov subspace of the *whole* parameter space with a dimension-free rate for *convex* objectives. Here the objective is non-convex, the subspace is built and regularized *per block* with its own M_b (Appendix M), and the depth is tied to the cubic shift by Proposition A.3.

3 The Per-Block Cubic Step and its Krylov Subsolver

This section develops the theory of the per-block step, with all proofs in Appendix A. The results concern the cubic step on a single block (identically, on the whole parameter vector), hold for a generic symmetric $\mathbf{H}$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$, $g \neq 0$, and $M > 0$, and for any subsolver.

3.1 The secular equation, the shift, and the degree bound

Lemma A.1 (Appendix A) restates Section 5 of Nesterov and Polyak (2006) (see also Cartis et al., 2011a) in our notation. If $\langle u_n,\, g\rangle \neq 0$ or $\lambda_n \geq 0$, the secular equation $\lambda = \frac{M}{2}\varphi(\lambda)$ with $\varphi(\lambda) = \big\|(\mathbf{H}+\lambda\mathbf{I})^{-1}g\big\|$ has a unique solution $\lambda^\star > \max\{0, -\lambda_n\}$, and $s^\star = -(\mathbf{H}+\lambda^\star\mathbf{I})^{-1}g$ is the global minimizer of (3). Lemma A.2 (Appendix A) adds the floor $r^\star \geq 2\max\{0,-\lambda_n\}/M$, so the shifted operator $\mathbf{H} + \frac{Mr^\star}{2}\mathbf{I}$ is always positive semidefinite, and positive definite in the non-degenerate case. Negative curvature is thus handled *before* the polynomial operates on the system.

Proposition A.3 (Appendix A) makes the self-adaptive depth schedule precise. The shift obeys $\lambda^\star \geq \lambda_{\mathrm{lo}} := \frac{1}{2}\big(\sqrt{\lambda_1^2 + 2M\,\|g\|} - \lambda_1\big)$, and whenever $\lambda_{\mathrm{lo}} \geq 2\max\{0,-\lambda_n\}$ the shifted operator has effective condition number $\kappa_{\mathrm{eff}} := \frac{\lambda_1+\lambda^\star}{\lambda_n+\lambda^\star} \leq \frac{2(\lambda_1+\lambda_{\mathrm{lo}})}{\max\{\lambda_n,0\}+\lambda_{\mathrm{lo}}}$, so a Chebyshev or Lanczos solver needs depth $L = \mathcal{O}\big(\sqrt{\kappa_{\mathrm{eff}}}\,\ln\frac{1}{\varepsilon_{\mathrm{rel}}}\big)$ for a relative accuracy $\varepsilon_{\mathrm{rel}}$. Far from stationarity ($2M\,\|g\| \geq \lambda_1^2$) $\kappa_{\mathrm{eff}} \leq 2 + \mathcal{O}(\lambda_1/\sqrt{M\,\|g\|})$, a constant independent of $\kappa = \lambda_1/\lambda_n$. Near a strict second-order stationary point the shift vanishes and $\kappa_{\mathrm{eff}} \to \lambda_1/\lambda_n$. The Krylov subproblem rates of Carmon and Duchi (2018) also depend on the shifted spectrum.

3.2 Krylov subspace solver and its optimality

We solve the cubic subproblem over the Krylov subspace, following the Lanczos-based trust region and cubic subproblem solvers of Gould et al. (1999); Cartis et al. (2011a). Run L steps of the Lanczos process (Golub and Van Loan, 2013) on $(\mathbf{H}, g)$ with full reorthogonalization (one HVP per step), producing $\mathbf{Q}_L \in \mathbb{R}^{n\times L}$ whose columns span $\mathcal{K}_L(\mathbf{H}, g) = \mathrm{span}\{g, \mathbf{H}g, \ldots, \mathbf{H}^{L-1}g\}$, with $\mathbf{Q}_L^\top \mathbf{H}\,\mathbf{Q}_L = \mathbf{T}_L$ tridiagonal ($L \times L$) and $\mathbf{Q}_L^\top g = \|g\|\,e_1$. The Lanczos depth L grows until the Cartis–Gould–Toint test (19) is met, $\|\nabla m(s_L)\| \leq \kappa_\theta \min(1, \|s_L\|)\,\|g\|$, which is the inexactness condition that transfers the $\mathcal{O}(\varepsilon^{-3/2})$ rate to a subspace step (Remark B.3, Lemma B.6). Proposition A.3 gives a worst case ceiling on that depth. The Lanczos process on $(\mathbf{H}_b, g_b)$ builds a basis of the Krylov subspace by a three term recurrence whose coefficients are computed from that pair, where the Chebyshev recurrence uses coefficients set in advance for a spectral interval. It evaluates the g_b-weighted spectrum of the tensor by Gauss quadrature (the Ritz values of $\mathbf{T}_L$, Remark A.6), and the secular solve then takes the exact cubic minimizer in that space (Lemma A.4). Modes orthogonal to g_b are not in the subspace (Remark B.2). Corollary A.5 adds that the depth-L Krylov step attains, at most, the model value of the Chebyshev second kind recurrence step (the step that motivated the ARC-block design) in the same space, at one HVP per basis vector (memory in

Appendix N). Carmon and Duchi (2019) describe gradient-based (first-order) subsolvers for the same subproblem, and Zhu and Chang (2022) bound the Lanczos projection error through the condition number of the shifted Hessian.

3.3 One-step progress and global complexity

The exact cubic step, and any Krylov step meeting (19), has the standard global guarantee. The theorem below mirrors Lemma 4.1 and Theorem 4.2 of Doikov et al. (2024) and the telescoping argument of Nesterov and Polyak (2006), Section 4, with Lemmas A.7 and A.8 in Appendix A.

Theorem 3.1 (Global complexity)**.** *Let f satisfy Assumption 2.1 with $f^\star := \inf f > -\infty$, and let $\{x_k\}$ be generated by $x_{k+1} = x_k + s_k^\star$ where $s_k^\star$ minimizes the cubic model at x_k with constant $M \geq L_H$. Then for any $\varepsilon > 0$, $\min_{1\leq i\leq K} \|\nabla f(x_i)\| \leq \varepsilon$ holds once the iteration count reaches*

$$K = \left\lceil \frac{12\,(f(x_0)-f^\star)}{M} \left(\frac{L_H+M}{2}\right)^{3/2} \varepsilon^{-3/2} \right\rceil = \mathcal{O}\Big(\frac{\sqrt{M}\,(f(x_0)-f^\star)}{\varepsilon^{3/2}}\Big).$$

Corollary A.9 (Appendix A) extends Theorem 3.1 to approximate second-order stationarity. For any $\varepsilon > 0$, within the same iteration count K some iterate x_i satisfies both $\|\nabla f(x_i)\| \leq \varepsilon$ and $\lambda_n(\nabla^2 f(x_i)) \geq -\frac{M+2L_H}{\sqrt{2(L_H+M)}}\sqrt{\varepsilon}$, so for small ε the curvature condition rules out strict saddles as limit points. Inexactness of the subproblem solve and the adaptivity of M are discussed in Remark B.3 (Appendix B). Remark B.4 delimits the scope of the guarantee.

3.4 Block-diagonal decomposition at scale

We now partition $\theta = (\theta_1, \ldots, \theta_B)$ by tensors (each block a single weight or bias tensor), let $\mathbf{H}_b$ denote the diagonal blocks, and apply the cubic step *per block*, sequentially, with fresh gradients. Block b minimizes $m_b(s_b) = \langle g_b,\, s_b\rangle + \frac{1}{2}\langle \mathbf{H}_b s_b,\, s_b\rangle + \frac{M_b}{6}\|s_b\|^3$ where g_b is the gradient evaluated *after* blocks $1, \ldots, b-1$ have been updated (a fresh gradient, which the theorems below assume for every block). Proposition A.10 (Appendix A) supplies the guarantee. Under the per-block ratio test of Algorithm 1 the iteration is monotone, and every accepted block step, exact or Krylov at any depth, decreases f by at least $\eta_1 \frac{M_b}{12}\|s_b\|^3$. Remark B.5 explains why joint acceptance over many blocks fails at scale. The cross-block coupling, absent from every block model, appears only in the gradient at a sweep's end. Theorem B.8 bounds the small block gradient after Phase A and Theorem B.9 bounds the large block gradient after Phase B. Theorem B.10 (Appendix B) assembles both phases of one sweep, Phase B is retried until accept, and gives $\mathcal{O}(\sqrt{B}\,\varepsilon^{-3/2})$ sweeps to an ε-second-order point for every ε above a floor $\varepsilon_\star = 2B\delta^2/C_1$ determined by the off-block curvature δ, which is measured by the loss landscape fingerprint. A Krylov step is successful when it meets the termination test of Cartis et al. (2011b), checked per trial (Appendix S).

4 ARC-block Algorithm

We now assemble the step of Section 3 into the blockwise optimizer, *ARC-block*. In the experiments, the large block step is called *CubicKrylov*. The name is also used as an optimizer label for the blockwise scheme before the two per-block stabilizations of Section 6.1. The Chebyshev second kind recurrence of Podorozhny (2026) is one of the interchangeable subsolvers (CubicCheby-DSO, Appendix N). The Lanczos basis is used for the reasons of Section 3.2.

4.1 The CubicKrylov step (single block)

Given an HVP oracle $v \mapsto \mathbf{H}v$ for one block: *(i)* a Lanczos build (L HVPs, float64 accumulation, full reorthogonalization) gives $\mathbf{Q}_L, \mathbf{T}_L$ with $\mathbf{Q}_L^\top g = \|g\|\, e_1$; *(ii)* the reduced secular equation (10) is solved on the $L \times L$ tridiagonal $\mathbf{T}_L = \mathbf{V}\,\mathrm{diag}(\nu)\,\mathbf{V}^\top$ by safeguarded bisection or Newton for the shift $\lambda > \max\{0, -\nu_{\min}\}$ (Lemma A.2), at $\mathcal{O}(L^3)$ cost, negligible for $L \ll n_b$; *(iii)* the step is lifted, $s_L = \mathbf{Q}_L y^\star$. Every trial λ reuses the cached $\mathbf{Q}_L, \mathbf{T}_L$, so the whole root finding loop costs L HVPs in total, independent of the number of trial shifts, with no $\mathcal{O}(n_b^3)$ factorization and no $\mathcal{O}(n_b^2)$ storage (Gould et al., 1999).

Algorithm 1 ARC-BLOCK: blockwise stabilized adaptive cubic regularization. The large block subsolver CUBICKRYLOV is interchangeable with the Chebyshev second kind recurrence or the φ_1 step (Appendix N, Section 5).

Require: parameters $\theta = (\theta_1, \dots, \theta_B)$; size threshold $n_{\max}$; Lanczos depth L; laziness p_b; Lipschitz estimate L_H; floor $M_{\min}$ and cap $M_{\max}$; initialization $M_{\text{init}} = 6L_H$; ratio thresholds $\eta_1 = 0.1$, $\eta_2 = 0.9$; factor $\gamma = 2$; tolerance $t = \tau_{\text{rel}}|\ell_{\text{cur}}| + \tau_{\text{abs}}$; rejection cap n_{rej}

partition blocks: $\mathcal{S} = \{b : n_b \le n_{\max}\}$ (small), $\mathcal{L} = \{b : n_b > n_{\max}\}$ (large)
for $k = 0, 1, 2, \dots$ **do**
  $g \leftarrow \nabla f(\theta)$; $\ell_{\text{cur}} \leftarrow f(\theta)$ ▷ sweep start closure; 1 gradient, 1 forward
  Phase A (small blocks; lazy exact cubic from the sweep start gradient, guarded):
  for $b \in \mathcal{S}$ **do**
    if $k \equiv 0 \pmod{p_b}$ or the last visit of b was rejected **then** rebuild $\mathbf{H}_b$ exactly; cache $\text{eigh}(\mathbf{H}_b)$
    $s_b \leftarrow$ exact cubic step for $(g_b, \mathbf{H}_b)$ from the cached eigendecomposition with $M = 6\, p_b\, L_H$ ▷ Doikov et al., 2023; $s_b = 0$ if $g_b = 0$
    $\theta_b \mathrel{+}= s_b$; $\ell_{\text{new}} \leftarrow f(\theta)$ ▷ 1 forward
    if $\ell_{\text{new}} > \ell_{\text{cur}}$ or not finite **then** revert θ_b; mark b rejected ▷ descent guard
    else $\ell_{\text{cur}} \leftarrow \ell_{\text{new}}$
  Phase B (large blocks; per-block CubicKrylov, retry; acceptance against ℓ_{cur} after Phase A):
  for $b \in \mathcal{L}$ **do**
    $g_b \leftarrow \nabla_{\theta_b} f(\theta)$ with graph ▷ *fresh* gradient, post Phase A / prior blocks
    if $\|g_b\| = 0$ **then**
      next b ▷ skip; $r_b = 0$
    else
      for trial $= 1, \dots, n_{\text{rej}} + 1$ **do**
        $s_b \leftarrow \text{CUBICKRYLOV}(\text{hvp}_b,\, g_b,\, M_b,\, L)$ ▷ Lemma A.4; $L{+}3$ gevals per trial
        $\theta_b \mathrel{+}= s_b$; $\ell_{\text{new}} \leftarrow f(\theta)$ ▷ 1 forward
        $\rho_b \leftarrow (\ell_{\text{cur}} - \ell_{\text{new}} + t)/(-m_b(s_b) + t)$ ▷ model decrease from one HVP
        if $\rho_b < \eta_1$ or $\ell_{\text{new}} > \ell_{\text{cur}} + t$ or not finite **then**
          revert θ_b; $M_b \leftarrow \min(\gamma M_b,\, M_{\max})$ ▷ reject; same g_b, larger M_b
        else
          $\ell_{\text{cur}} \leftarrow \ell_{\text{new}}$; **if** $\rho_b \ge \eta_2$: $M_b \leftarrow \max(M_b/\gamma,\, M_{\min})$; **break**

4.2 THE BLOCKWISE ALGORITHM AND ITS DESIGN DECISIONS

Algorithm 1 embeds the step into the block optimizer. Three of its design decisions, each a response to an observed failure, deserve emphasis here, with the full narratives and the computational cost table in Appendix D. *(a) Two phase update with guarded small blocks.* Small blocks (biases and norms, $n_b \le 512$) take lazy exact cubic steps (Doikov et al., 2023) from the sweep start gradient, each kept only if the loss does not rise (a descent guard, with no ratio test and no M adaptation). Phase B acceptance is measured against the loss after Phase A. *(b) Per-block acceptance and per-block M_b* (Remark B.5). A joint test over large tensors may reject even when every block step is good, so per-block testing is used, at one forward pass per large block. A rejected Phase B trial restores block b, raises M_b, and is retried in that sweep until acceptance, which Proposition A.10 guarantees once $M_b \ge L_{H,b}$. *(c) A nearly zero floor for M_b.* A floor $M_{\min} = 6L_H$ over-regularizes blocks whose local Hessian Lipschitz constant is far below the global estimate (accepted but microscopic steps). So we set $M_{\text{init}} = 6L_H$ and let the adaptation of Algorithm 1 ($M_b\gamma$ on a rejected trial, M_b/γ on a very successful one, $\rho_b \ge \eta_2$) take M_b down to a nearly zero $M_{\min}$. Phase B gradients are recomputed after Phase A and after each preceding large block. Phase A reuses the gradient computed at the sweep start. Its Hessians use lazy computation. Reusing that gradient in Phase B too lowers the final PSNR by $\approx$8.5 dB on ViSIR-Nano (Table 2). The CubicKrylov step (e.g. with $L \approx 10$) is the only entry of Table 1 whose oracle cost is independent of n_b, which makes cubic steps feasible at large scale, e.g. on the 88.5M parameter ViSIR decoder weight.

5 THE EXPONENTIAL RELAXATION (φ_1) STEP SUBSOLVER

The subsolver's *step rule* is an interchangeable component of the ARC-block outer scheme. This section introduces the *exponential relaxation* step, known in the exponential integrator literature as a φ_1 step (Hochbruck and Lubich, 1997; Hochbruck and Ostermann, 2010),

$$s \;=\; -H(G,h)\, g, \qquad H(G,h) \;=\; \int_0^h e^{-G\tau}\, d\tau \;=\; h\, \varphi_1(-hG), \tag{5}$$

the exact solution at time h of the gradient flow $\dot{x} = -g - G\,(x - x_0)$ of the same local quadratic model, integrated in closed form up to a finite horizon h (Remark B.12, Appendix B).

Here G plays the role of the block Hessian $\mathbf{H}_b$ and g the block gradient g_b of Section 4. The outer scheme is reused unchanged: the per-block trust constant σ_b (with $M_b = 2\sigma_b$ as implemented), the ratio test, and the proportional offset monotone guard of the stabilized block-σ ARC ablation (Appendix M),

$$f(x + s_b) \;\le\; f(x) + t, \qquad t \;=\; \tau_{\mathrm{rel}}\,|f(x)| + \tau_{\mathrm{abs}}, \tag{6}$$

of Section 6.1. We write $h_b = c_h/\sigma_b$ for the relaxation horizon, so the trust adaptation is also the φ_1 step size control. Rejection contracts the horizon, and sustained success extends it toward the Newton limit. The resulting method, **ARC-φ_1**, shares the outer scheme with ARC-block, so any performance difference is due to the step rule.

5.1 The φ_1 step rule

Proposition 5.1 (φ_1 step rule)**.** *Let G be symmetric with eigendecomposition $G = \mathbf{V}\Lambda\mathbf{V}^\top$. Then $H(G,h) = \mathbf{V}\,\mathrm{diag}\big(\eta_h(\lambda_i)\big)\,\mathbf{V}^\top$ with*

$$\eta_h(\lambda) \;=\; \frac{1 - e^{-h\lambda}}{\lambda} \quad (\lambda \neq 0), \qquad \eta_h(0) = h, \tag{7}$$

and: (i) $\eta_h(\lambda) \to 1/\lambda$ as $h\lambda \to \infty$ (Newton limit); (ii) $\eta_h(\lambda) = h\,(1 - h\lambda/2 + \mathcal{O}(h^2\lambda^2))$ as $\lambda \to 0$ (gradient limit); (iii) for $\lambda < 0$, $\eta_h(\lambda) = (e^{h|\lambda|} - 1)/|\lambda|$ grows exponentially in h (saddle amplification); (iv) the doubling identity $H(G,2h) = H(G,h)\,(2\mathbf{I} - G\,H(G,h))$ holds, so a geometric horizon search costs one application per candidate; (v) for every $h > 0$ and $g \neq 0$ the step $s = -H(G,h)\,g$ decreases the quadratic model, $m(s) < 0$, whatever the signs of the spectrum (equation (9)). Under Assumption 2.1 the decrease transfers to f once the cubic remainder $L_H\,\|s\|^3/6$ is below the model decrease. For $G \succeq 0$ and $h \le 0.1/\,\|G\|$ that condition is $L_H h^2\,\|g\|/6 < \varphi_1(-1/5)$, so that the horizon bound alone does not cancel L_H (Appendix F).

The contrast with the CubicKrylov step of Section 2.2 is most noticeable on the flat modes, where no regularization is needed and the cubic multiplier $1/(\lambda + M\,\|s\|/2)$ (written $1/(\lambda + \sigma\,\|s\|)$ with $M = 2\sigma$) suppresses progress. The φ_1 multiplier (7) decouples the three cases per mode with no shift estimation and no secular solve. A drawback is that the negative mode amplifier must be bounded explicitly (Lemma A.11) and that no analogue of Lemma A.1 is claimed. The search step magnitude is safeguarded only by the outer acceptance test, whose rejections contract the horizon $h_b = c_h/\sigma_b$.

5.2 Krylov evaluation, decrease law, and the negative mode clamp

Proposition 5.1 defines the step through the eigenpairs of G, which are not available on a large block, so the multiplier (7) is applied on the CubicKrylov step's Lanczos basis, and the two subsolvers differ only in the function of the Ritz values.

For a block with gradient g_b and oracle $\mathbf{H}_b$, run L steps of the Lanczos process with full reorthogonalization from $q_1 = g_b/\,\|g_b\|$, producing $\mathbf{Q}_L \in \mathbb{R}^{n_b \times L}$ and tridiagonal $\mathbf{T}_L = \mathbf{V}\,\mathrm{diag}(\nu)\,\mathbf{V}^\top$ with Ritz values ν_i, exactly as in Section 3.2. The search step is the subspace φ_1 step

$$s_b \;=\; -\mathbf{Q}_L\,\mathbf{V}\,\mathrm{diag}\big(\tilde{\eta}_{h_b}(\nu_i)\big)\,\mathbf{V}^\top\,\mathbf{Q}_L^\top g_b \;=\; -\,\|g_b\|\;\mathbf{Q}_L\mathbf{V}\,\mathrm{diag}\big(\tilde{\eta}_{h_b}(\nu_i)\big)\,\mathbf{V}^\top e_1, \tag{8}$$

where $\tilde{\eta}$ is the clamped multiplier of Lemma A.11. A trial costs $L + 3$ gradient equivalents (one gradient, L Lanczos HVPs, one HVP for the model decrease, one function evaluation), the cost of the CubicKrylov step at equal L (Table 1), so any difference in the results of Section 6 is due to the step rule. The Lanczos approximation converges superlinearly once L exceeds $h\,\|\mathbf{H}_b\|$ (Saad, 1992). The truncation error is caught by the ratio test, which rejects a trial whose measured decrease is less than η_1 (the acceptance threshold of Algorithm 1) times the predicted one and contracts the horizon.

Diagonalizing G and summing the per-mode multipliers of Proposition 5.1 gives an exact single-step decrease law on the frozen quadratic $m(u) = \langle g,\,u\rangle + \frac{1}{2}\langle u,\,Gu\rangle$:

$$m(0) - m(s_h) \;=\; h\;\langle g,\,\varphi_1(-2hG)\,g\rangle \;>\; 0, \tag{9}$$

strictly positive for every sign pattern of the spectrum. The hard case of Lemma A.1 has no analogue here. It is bounded below by the Cauchy type estimate $m(0) - m(s_h) \ge \frac{1}{4}\,\|g\|^2\,\min(h, 1/\,\|G\|)$,

and an accepted trial decreases f by at least η_1 times the model decrease $m(0) - m(s_h)$, up to the tolerance. Under a uniform floor $\underline{h} > 0$ on $\min(h_b, 1/\|\mathbf{H}_b\|)$ over accepted steps, and the transfer of the model decrease to f of Proposition 5.1(v), ARC-φ_1 reaches an ε-first-order point within $4(f(x_0) - f^\star)/(\eta_1 \underline{h}\, \varepsilon^2)$ accepted steps, the $\mathcal{O}(\varepsilon^{-2})$ class of steepest descent (Cartis et al., 2010). The $\mathcal{O}(\varepsilon^{-3/2})$ cubic rate does not transfer, and second-order stationarity (Corollary A.9) transfers only generically because the clamped escape of Lemma A.11 is proportional to $\langle u_n, g\rangle$. In a Newton limit basin the complexity improves to $\mathcal{O}(\varepsilon^{-1})$ sweeps, with a Q-linear (quotient-linear) loss contraction, $f(x_{k+1}) - f^\star \le \vartheta\,(f(x_k) - f^\star)$ with $\vartheta < 1$ (Appendix T; Nocedal and Wright, 2006, Appendix A.2).

The negative mode clamp. Lemma A.11 (Appendix A) clamps the unbounded negative mode amplifier $(e^{h|\lambda|} - 1)/|\lambda|$ so that, for $M_{\text{amp}} \ge 2$, one step amplifies any gradient component by at most $(M_{\text{amp}} - 1)/|\lambda|$ and the step is finite in fp32 for any horizon and spectrum.

Trust-controlled horizons. The composite scheme sets $h_b = c_h/\sigma_b$ and evaluates the acceptance ratio against the subspace quadratic model (one extra HVP). Rejection contracts h_b geometrically, and by Proposition 5.1(v) a small enough horizon brings the cubic remainder below the model decrease, so acceptance follows after finitely many rejections (Algorithm 2, Appendix E).

6 Experimental Results

We evaluate ARC-block and ARC-φ_1 at 91.4M parameters (Section 6.1), measure the landscape features that decide where they win (Section 6.2), and run FINER to full convergence (Section 6.3). All runs use PyTorch on one RTX PRO 6000 GPU (96 GB). The full scale model is ViSIR (Zeraatkar et al., 2025) (91,383,042 parameters), a vision transformer encoder with SIREN feed-forward blocks and decoder whose final layer holds 97% of the model. ViSIR-Nano (Appendix H, Table 2) is the same architecture at 15,235 parameters. We report PSNR in dB, and cost in HVPs, gradient equivalents (gevals, one per gradient, backward pass, or function evaluation), and wall clock seconds.

6.1 The 91.4M parameter single image ViSIR Benchmark

The main experiment extends the Chebyshev second kind ablation (ESM image #1, $60 \times 60 \to 240 \times 240$ image super-resolution task, 100 steps) with Algorithm 1, $L = 10$, $p_b = n_b$ (lazy), $L_H = 10$ (Table 3, Appendix I). Scaling to the full model exposed the two failure modes of Section 3.4. With the per-block corrections of Section 4, the CubicKrylov optimizer reaches **51.65 dB** at 36,836 gevals in 148 s, +9.90 dB over the best Chebyshev-ON result and +29.4 dB over damped Newton large block steps. With the two per-block stabilizations it reaches **64.4 dB** best PSNR over 150 sweeps (**64.2 dB** with the Chebyshev three term recurrence, Appendix N.1).

6.2 Landscape Fingerprints Along the Adam Trajectory

Adam divides each gradient coordinate by $\sqrt{\hat{v}_t} + \epsilon$, so the landscape it sees has the curvature of $D^{-1/2}\mathbf{H}D^{-1/2}$ with $D = \text{diag}(\sqrt{\hat{v}_t} + \epsilon)$. At log-spaced checkpoints of an Adam run we measure matrix-free the raw and Adam preconditioned conditionings κ_{raw} and κ_{Adam}, the diagonal mass $\rho = \|\text{diag}\,\mathbf{H}\|^2/\|\mathbf{H}\|_F^2$, the negative spectral mass (Ghorbani et al., 2019; Dauphin et al., 2014), and the gradient energy split over high curvature, flat, and negative curvature classes (Gur-Ari et al., 2018). When ρ is small, high curvature and flat directions are mixtures of coordinates, and Jacobi scaling barely changes κ. Since for a positive definite matrix no diagonal scaling improves on Jacobi scaling by more than a factor n (van der Sluis, 1969), the per-coordinate class, Adam included, faces the raw ill-conditioning, and second-order methods are more effective there.

Table 4 (Appendix I) distinguishes three cases. *(i) Axis-aligned* (35k SIREN-like, SIREN 2D and signed distance function (SDF), WIRE 2D). The rescaling leaves a small κ_{Adam}, the gradient never enters flat or negative directions, and tuned Adam reaches parity with second-order methods (Appendix J). *(ii) Coupled.* Multisaddle and Chebyshev–Rosenbrock (Jarre, 2013) retain κ_{Adam} up to 1.5×10^6 at $\rho \le 0.04$, the landscapes where the first-order methods evaluated here stall at any tuning (ViSIR is in this class). *(iii) Saddle dominated.* FINER (2D and SDF) has small κ_{Adam} (61 and 6.9), so conditioning does *not* limit Adam, but its negative spectral mass is 0.81 at initialization (0.57–0.89 over the 20 Earth System Model (ESM) images). Adam converges to a saddle adjacent

point (Sagun et al., 2018) at 65 dB with a low frequency dominated residual, while the cubic shift keeps every shifted subproblem positive semidefinite and ARC-block continues to 120–129 dB with a band uniform residual. The same loss landscape fingerprint appears in 3D (SDF-FINER, negative mass 0.68, Adam chamfer 4.6–5.5 times worse) and on small hard landscapes (Appendix K).

6.3 FINER 2D Image Fitting with the φ_1 Step at Full Convergence

FINER (Liu et al., 2024), the saddle dominated case of Section 6.2, is a coordinate multilayer perceptron (MLP) with the variable periodic activation $\sin((|x|+1)x)$ in place of SIREN's fixed period $\sin(\omega_0 x)$. Three hidden layers of width 256 ($\sim$199k parameters) map pixel coordinates to RGB on the 20 240×240 ESM images, the hardest initialization of our benchmarks.

Run to full convergence. We run the sweep winner ($c_h = 1.0$, $\sigma_0 = 0.3$, $L_{\max} = 163$, Table 5) and tuned Adam on image 0 to their plateaus. Adam peaks at $\mathbf{78.2}$ dB at 2,629 s and then degrades. ARC-φ_1 reaches $\mathbf{133.5}$ dB at 17,173 s. On all 20 ESM images, run to a plateau (Appendix I.5), ARC-φ_1's per-image best has median 132.4 and mean 131.5 ± 3.3 dB against Adam's 72.8 ± 4.5 dB and AdaCubic's 46.7 ± 0.3 dB, a lead over Adam of 42.1–66.4 dB on every image.

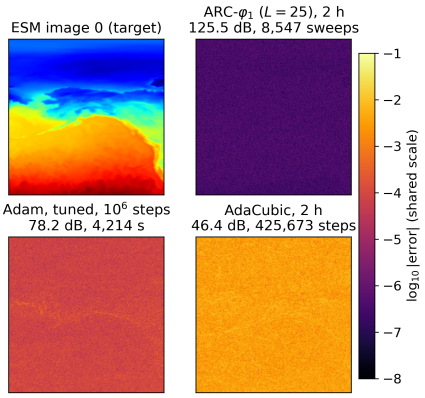

Figure 1: FINER on ESM image 0 in equal wall clock time of 2 h, $\log_{10}$ absolute error on a shared scale. ARC-φ_1 at Lanczos depth L=25 reaches 125.5 dB in 8,547 sweeps. Tuned Adam converges to 78.2 dB (10^6 steps, 4,214 s). AdaCubic (Tsingalis et al., 2026) with its authors' universal hyperparameters reaches 46.4 dB at 6,328 s and then oscillates (Appendix I.5). SOAP (Vyas et al., 2024) converges at 77.65 dB and L-BFGS stays at 7.3 dB at the initialization saddle (Table 6). At its tuned depth L=163, ARC-φ_1 converges to 133.5 dB (2,824 sweeps, 17,173 s). The faint bright filament along the warm/cold front boundary in the Adam panel is signal dependent error.

Iso-time comparison. At 277 gevals per ARC-φ_1 sweep at L=25 against one per Adam step, Table 6 and Figure 1 compare at equal wall clock on ESM image 0. ARC-φ_1 passes Adam's 78.2 dB at 1,927 s, reads 84.2 dB at Adam's peak time (2,629 s) and $\mathbf{95.6}$ dB at 4,207 s, Adam's 10^6-step time. AdaCubic (Tsingalis et al., 2026), whose authors call it "the first optimizer to leverage cubic regularization in scalable deep learning applications", reaches 46.4 dB in 2 h, 32 dB below Adam.

Practical consequence. At 6.02 dB per bit, Adam's 73–78 dB ($\approx$12–13 bits) would discard 3–4 bits of a 16-bit scientific raster as a spatially coherent bias field that does not average out over regions and biases climate statistics. ARC-φ_1's 131–134 dB ($\approx$22 bits) lies below 16-bit quantization noise (107 dB), so the continuous representation replaces the raster without measurable error.

7 Conclusion and Limitations

We introduced ARC-block, a blockwise stabilized ARC optimizer with an independent cubic constant and acceptance test per tensor and an interchangeable subsolver. Its φ_1 subsolver, at the CubicKrylov step's per-trial cost, reaches 133.5 dB against tuned Adam's 78.2 dB at full convergence. We proved approximate second-order stationarity (Remark B.4), an $\mathcal{O}(\sqrt{B}\varepsilon^{-3/2})$ sweep bound above the cross-block coupling floor (Theorem B.10), and the Cauchy type $\mathcal{O}(\varepsilon^{-2})$ guarantee of the φ_1 rule (Section 5.2). On the landscapes with the features where Adam and its variants struggle (cross-coupled ill-conditioning and persistent negative spectral mass), ARC-block outperforms every optimizer evaluated here. A major limitation is that the cross-block curvature is lost in the blockwise scheme. Also, the 20 image experiment uses only one seed. There is room for tighter theoretical guarantees as well. Directions for future work are as follows: (i) per-block probes of the tensor-block mass β along a run, which bound the cross-block coupling δ and so the floor $\varepsilon_\star$ (Remark B.11), toward removing the floor from the $\mathcal{O}(\varepsilon^{-3/2})$ rate; (ii) a per-block fingerprint with the Gauss–Newton decomposition of the Hessian, whose residual term holds all negative curvature and whose kernel spectrum, the neural tangent kernel (NTK), predicts the first-order loss curve (Appendix P; Jacot et al., 2018; for Newton's method Riedl et al., 2026); and (iii) other step rules and larger networks in other domains.

AI USE STATEMENT

Generative AI tools (LLM coding agents) were used throughout this work under the authors' direction, and the authors take responsibility for the final content, including text, claims, and artifacts produced with their aid. Among the tasks the ICLR policy lists as required, the agents implemented the PyTorch versions of the optimizers based on the NumPy versions written and extensively tested manually by the authors. The search steps of the optimizers introduced in the paper were tested extensively by hand on many convex and non-convex landscapes, using their search trajectories, including the test cases in which the minima could be calculated analytically, so as to verify the minima found by the NumPy versions implemented by hand. The current manuscript describes the results of a year of intensive work. The agents generated the experiment drivers, and the analysis scripts from the formal specifications written by the authors, following disciplined software engineering processes (e.g. an in-house quantified variant of the Module of Thought (MoT) development process) and program analysis methods for verification and validation. The implementations of the critical functions, including the search steps and the polynomial calculations, were verified by Floyd–Hoare proofs against the pre- and post-conditions given by the search step and polynomial formulas. The dynamic behavior was verified by the agreement of the search trajectories on the same test functions, both those specified analytically and those approximated by regression. This checked the agents' PyTorch implementations against the NumPy implementations. The agents also assisted in composing the proofs of Appendix A, including the rewrites made after an audit of each argument against its cited source paper, and they wrote the Lean library of Appendix F. The proofs were largely adapted to the blockwise case from the proofs in the referenced papers. The proofs adapted to the blockwise case were checked by hand as well, in addition to Lean 4. Among the recommended tasks, the agents drafted and edited the text for readability against a controlled vocabulary built from the cited literature based on repeatable experimental results in JSON files, and verified the bibliography against the cited papers in PDFs. The auto-generated text in response to the authors' outline and JSON files was heavily edited by hand to ensure the authors' vision, emphasis, and consistent use of terms. All the cited papers that had an accessible PDF were downloaded and processed. Some of the references were obtained by the authors as hard copies (books). No synthetic data were generated. Every reported number is read from a logged run. The algebraic cores of the blockwise and φ_1 proof arguments were checked by the Lean 4 kernel (Appendix F, with the hypotheses left outside the library listed there). Each proof was re-derived and its provenance compared with the cited papers. The 20 ESM image experiment means, medians, and paired gains were recomputed from the run logs. Every bibliography entry was compared with the title page of the cited document.

REPRODUCIBILITY STATEMENT

Appendix A restates every theorem with a full proof and Appendix F lists the machine-checked cores and the hypotheses outside them. Appendix I.5 describes the 20 ESM image experiment (architecture, seeds, stopping rules, per-image runs and wall clock), Appendix I describes the remaining experimental details and iso-time tables, and Appendix N describes the computational cost evaluation in gradient equivalents. The optimizers, sweep drivers, and analysis scripts will be released upon publication, and the ESM images are the public ViSIR benchmark.

REFERENCES

Y. Nesterov and B. T. Polyak. Cubic regularization of the Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

C. Cartis, N. I. M. Gould, and P. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *SIAM Journal on Optimization*, 20(6):2833–2852, 2010.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, second edition, 2006.

C. Cartis, N. I. M. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

C. Cartis, N. I. M. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Mathematical Programming*, 130(2):295–319, 2011.

C. Cartis, Z. Shao, and E. Tansley. Random subspace cubic-regularization methods, with applications to low-rank functions. arXiv preprint arXiv:2501.09734, 2025.

S. Bellavia, D. Palitta, M. Porcelli, and V. Simoncini. Regularized methods via cubic model subspace minimization for nonconvex optimization. *Computational Optimization and Applications*, 90(3):801–837, 2025. doi:10.1007/s10589-025-00655-2.

H. Al Daas and N. I. M. Gould. Extended-Krylov-subspace methods for trust-region and norm-regularization subproblems. arXiv preprint arXiv:2511.11135, 2025.

A. Cristofari. Block cubic Newton with greedy selection. *Optimization Methods and Software*, 2026. Published online 28 June 2026, doi:10.1080/10556788.2026.2692465; arXiv:2407.18150.

P. Wolinski. Gathering and exploiting higher-order information when training large structured models. arXiv preprint arXiv:2312.03885, 2025.

A. Ranganath, M. Singhal, and R. Marcia. Symmetric rank-one quasi-Newton methods for deep learning using cubic regularization. arXiv preprint arXiv:2502.12298, 2025.

J. Forristal, J. Griffin, W. Zhou, and S. A. Yektamaram. A novel fast exact subproblem solver for stochastic quasi-Newton cubic regularized optimization. arXiv preprint arXiv:2204.09116, 2022.

D. Pasechnyuk-Vilensky and M. Takáč. Chebyshev-exact acceleration under Hessian variation, I: Sine-Jacobi method. arXiv preprint arXiv:2606.16671, 2026.

E. Tansley and C. Cartis. Scalable second-order optimization algorithms for minimizing low-rank functions. In *OPT 2024: Workshop on Optimization for Machine Learning*, 2024. arXiv:2501.03718.

Y. Zhou, J. Xu, B. Li, C. Bao, C. Ding, and J. Zhu. A regularized Newton method for nonconvex optimization with global and local complexity guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 38:81676–81731, 2025.

C. Cartis, S. Jerad, and K. Welzel. On global rates for regularization methods based on secant derivative approximations. arXiv preprint arXiv:2509.07580, 2025.

N. Marumo and A. Takeda. A general recipe for parameter-free nonconvex optimization via higher-order regularization. arXiv preprint arXiv:2605.30891, 2026.

S. Fang, N. Marumo, and A. Takeda. Parameter-free cubic-regularized Newton method: Sharp complexity and generalized smoothness. arXiv preprint arXiv:2607.10741, 2026.

A. Shestakov, N. Bashirov, A. Semenov, A. Gasnikov, M. Takáč, A. Beznosikov, and D. Kamzolov. Adaptive regularized Newton method with inexact Hessian. arXiv preprint arXiv:2512.08775, 2025.

D. Pasechnyuk-Vilensky, D. Kamzolov, and M. Takáč. Cubic regularized Newton method with variance reduction for finite-sum non-convex problems. arXiv preprint arXiv:2510.08714, 2025.

K. Scheinberg and M. Xie. First- and second-order stochastic adaptive regularization with cubics: High probability iteration and sample complexity. arXiv preprint arXiv:2308.13161, 2023.

L. Chen, C. Liu, L. Luo, and J. Zhang. Faster Newton methods for convex and nonconvex optimization in gradient complexity. In *Conference on Learning Theory (COLT)*, PMLR, 2026. arXiv:2501.17488.

T. Fuji, P.-L. Poirion, and A. Takeda. Theoretical analysis of the randomized subspace regularized Newton method for non-convex optimization. *Open Journal of Mathematical Optimization*, 6:article no. 8, 2025.

R. Higuchi, P.-L. Poirion, and A. Takeda. Improving convergence guarantees of random subspace second-order algorithm for nonconvex optimization. In *International Conference on Learning Representations (ICLR)*, 2025.

N. Tsipinakis, P. Tigas, and P. Parpas. A multilevel low-rank Newton method with super-linear convergence rate and its application to non-convex problems. *Transactions on Machine Learning Research*, 2026.

Z. Zeng, J. Zhang, and C. He. Adaptive Newton-CG methods with global and local analysis for unconstrained optimization with Hölder continuous Hessian. arXiv preprint arXiv:2604.02763, 2026.

H. Zeng, Y. Liu, W. Ouyang, and A. Milzarek. A MINRES-based linesearch algorithm for nonconvex optimization with non-positive curvature detection. arXiv preprint arXiv:2601.01575, 2026.

N. Abreu, N. Vyas, S. M. Kakade, and D. Morwani. The potential of second-order optimization for LLMs: A study with full Gauss-Newton. arXiv preprint arXiv:2510.09378, 2026.

D. Martins Gomes. Towards practical second-order optimizers in deep learning: Insights from Fisher information analysis. M.A.Sc. thesis, Concordia University, 2025. arXiv:2504.20096.

F. Hamad and O. Hinder. A simple and practical adaptive trust-region method. arXiv preprint arXiv:2412.02079, 2024.

Y. Ha, S. Shashaani, and Q. Tran-Dinh. Adaptive regularization within trust region methods for stochastic nonconvex optimization. arXiv preprint arXiv:2604.15457, 2026.

C. Zhang and R. Jiang. Riemannian adaptive regularized Newton methods with Hölder continuous Hessians. *Computational Optimization and Applications*, 92(1):29–79, 2025. doi:10.1007/s10589-025-00692-x.

Y. Pei, Y. Lin, S. Shao, M. Silva Louzeiro, and D. Zhu. A scalable sequential adaptive cubic regularization algorithm for optimization with general equality constraints. arXiv preprint arXiv:2503.11254, 2025.

J.-L. Wang, Z. Xu, and H.-L. Zhang. A fully parameter-free second-order algorithm for convex-concave minimax problems. *Journal of Machine Learning Research*, 2025. arXiv:2407.03571.

W. Huo, C. Liu, K. Ding, K. H. Johansson, and L. Shi. Federated cubic regularized Newton learning with sparsification-amplified differential privacy. arXiv preprint arXiv:2408.04315, 2024.

A. d'Aspremont, D. Scieur, and A. Taylor. Acceleration methods. *Foundations and Trends in Optimization*, 5(1–2):1–245, 2021.

J. M. Kohler and A. Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning (ICML)*, PMLR 70:1895–1904, 2017.

J.-P. Dussault, T. Migot, and D. Orban. Scalable adaptive cubic regularization methods. *Mathematical Programming*, 207(1–2):191–225, 2023. doi:10.1007/s10107-023-02007-6.

I. Tsingalis, C. Kotropoulos, and C. Briat. AdaCubic: an adaptive cubic regularization optimizer for deep learning. *Transactions on Machine Learning Research*, 2026.

N. Tripuraneni, M. Stern, C. Jin, J. Regier, and M. I. Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 2899–2908, 2018.

P. Xu, F. Roosta, and M. W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *Mathematical Programming*, 184(1–2):35–70, 2019. doi:10.1007/s10107-019-01405-z.

Y. Carmon and J. C. Duchi. Gradient descent finds the cubic-regularized nonconvex Newton step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2019.

Y. Carmon and J. C. Duchi. Analysis of Krylov subspace solutions of regularized non-convex quadratic problems. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.

Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1):71–120, 2020.

A. S. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

S. Masiha, S. Salehkaleybar, N. He, N. Kiyavash, and P. Thiran. Stochastic second-order methods improve best-known sample complexity of SGD for gradient-dominated functions. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022.

N. Doikov, S. U. Stich, and M. Jaggi. Spectral preconditioning for gradient methods on graded non-convex functions. In *International Conference on Machine Learning (ICML)*, 2024.

N. Doikov, E. M. Chayti, and M. Jaggi. Second-order optimization with lazy Hessians. In *International Conference on Machine Learning (ICML)*, 2023.

N. Doikov and Y. Nesterov. Universal reduced-operator method and high-order global curvature bounds. *arXiv preprint arXiv:2511.07341*, 2025.

A. Semenov, M. Jaggi, and N. Doikov. Gradient-normalized smoothness for optimization with approximate Hessians. *arXiv preprint arXiv:2506.13710*, 2025.

N. Doikov and P. Richtárik. Randomized block cubic Newton method. In *International Conference on Machine Learning (ICML)*, 2018.

N. Doikov and G. N. Grapiglia. First and zeroth-order implementations of the regularized Newton method with lazy approximated Hessians. *arXiv preprint arXiv:2309.02412*, 2023.

N. Doikov and A. Rodomanov. Polynomial preconditioning for gradient methods. In *International Conference on Machine Learning (ICML)*, 2023.

N. Doikov. *New Second-Order and Tensor Methods in Convex Optimization*. PhD thesis, Université catholique de Louvain (UCLouvain), 2021.

E. M. Chayti, N. Doikov, and M. Jaggi. Improving stochastic cubic Newton with momentum. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR 258, 2025.

E. M. Chayti, M. Jaggi, and N. Doikov. Unified convergence theory of stochastic and variance-reduced cubic Newton methods. *Transactions on Machine Learning Research (TMLR)*, 2024.

A. Agafonov, D. Kamzolov, P. Dvurechensky, A. Gasnikov, and M. Takáč. Inexact tensor methods and their application to stochastic convex optimization. To appear in *Optimization Methods and Software*.

Z. Zhu and J. Chang. Solving the adaptive cubic regularization sub-problem using the Lanczos method. *Symmetry*, 14(10):2191, 2022.

J. Zhao, A. Lucchi, and N. Doikov. Cubic regularized subspace Newton for non-convex optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2025.

F. Jarre. On Nesterov's smooth Chebyshev–Rosenbrock function. *Optimization Methods and Software*, 28(3):478–500, 2013. (Preprint, University of Düsseldorf, 2011.)

R. M. Podorozhny. Curvature-aware optimization via Chebyshev polynomials of the second kind for deep learning. In *SIAM Conference on Optimization (OP26), Book of Abstracts*, session CP4, Edinburgh, June 2026. Talk abstract, https://www.siam.org/media/r0be0xtr/op26_abstracts_v3.pdf.

V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Oral presentation.

E. Zeraatkar, S. A. Faroughi, and J. Tešić. ViSIR: Vision transformer single image reconstruction method for Earth System Models. *IEEE Pulse*, 16(1):23–25, 2025.

N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

V. Saragadam, D. LeJeune, J. Tan, G. Balakrishnan, A. Veeraraghavan, and R. G. Baraniuk. WIRE: wavelet implicit neural representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Z. Liu, H. Zhu, Q. Zhang, J. Fu, W. Deng, Z. Ma, Y. Guo, and X. Cao. FINER: flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

A. Kania, M. Mihajlović, S. Prokudin, J. Tabor, and P. Spurek. FreSh: frequency shifting for accelerated neural representation learning. In *International Conference on Learning Representations (ICLR)*, 2025.

S.-F. Chng, H. Saratchandran, and S. Lucey. Preconditioners for the stochastic training of neural fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27222–27232, 2025.

K. Shi, H. Chen, L. Zhang, and S. Gu. Inductive gradient adjustment for spectral bias in implicit neural representations. In *International Conference on Machine Learning (ICML)*, PMLR 267:54864–54891, 2025.

S. Ling, M. Nimier-David, A. Jacobson, and N. Sharp. Stochastic preconditioning for neural field optimization. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 44(4), 2025.

A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)*, PMLR 119:3789–3799, 2020.

D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

J. Jordan, K. Keller, Y. Jin, V. Boza, J. You, F. Cesista, L. Newhouse, and J. Bernstein. Muon: An Optimizer for Hidden Layers in Neural Networks. https://kellerjordan.github.io/posts/muon/, 2024.

R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

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

G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, 4th edition, 2013.

K. Mishchenko. Regularized Newton method with global $\mathcal{O}(1/k^2)$ convergence. *SIAM Journal on Optimization*, 33(3):1440–1462, 2023.

N. Doikov and Y. Nesterov. Gradient regularization of Newton method with Bregman distances. *Mathematical Programming*, 204(1–2):1–25, 2023. doi:10.1007/s10107-023-01943-7.

S. Gratton, S. Jerad, and P. L. Toint. Yet another fast variant of Newton's method for nonconvex optimization. arXiv preprint arXiv:2302.10065, 2023.

R. Jiang, P. Raman, S. Sabach, A. Mokhtari, M. Hong, and V. Cevher. Krylov cubic regularized Newton: a subspace second-order method with dimension-free convergence rate. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR 238, 2024.

R. Das, N. Agarwal, S. Sanghavi, and I. S. Dhillon. Towards quantifying the preconditioning effect of Adam. *arXiv preprint arXiv:2402.07114*, 2024.

K. Jiang, D. Malik, and Y. Li. How does adaptive optimization impact local neural network geometry? In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

T. H. Zhang, L. Maes, A. Milligan, A. Jolicoeur-Martineau, I. Mitliagkas, D. Scieur, S. Lacoste-Julien, and C. Guille-Escuret. Understanding Adam requires better rotation dependent assumptions. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2025.

S. Xie, M. A. Mohamadi, and Z. Li. Adam exploits ℓ_∞-geometry of loss landscape via coordinate-wise adaptivity. *arXiv preprint arXiv:2410.08198*, 2024.

Y. Zhang, C. Chen, T. Ding, Z. Li, R. Sun, and Z.-Q. Luo. Why transformers need Adam: a Hessian perspective. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Z. Dong, Y. Zhang, J. Yao, and R. Sun. Towards quantifying the Hessian structure of neural networks. *arXiv preprint arXiv:2505.02809*, 2025.

Sagun, L., Evci, U., Güney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the Hessian of over-parametrized neural networks. *ICLR Workshop*, 2018.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *NeurIPS*, 2018.

G. Gur-Ari, D. A. Roberts, and E. Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

B. Ghorbani, S. Krishnan, and Y. Xiao. An investigation into neural net optimization via Hessian eigenvalue density. In *International Conference on Machine Learning (ICML)*, PMLR 97:2232–2241, 2019.

Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*, 2014.

J. Cohen, B. Ghorbani, S. Krishnan, N. Agarwal, S. Medapati, M. Badura, D. Suo, D. Cardoze, Z. Nado, G. E. Dahl, and J. Gilmer. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022.

A. van der Sluis. Condition numbers and equilibration of matrices. *Numerische Mathematik*, 14(1):14–23, 1969.

A. Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. *Technical Report NA/12*, DAMTP, University of Cambridge, 1981.

M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 34(5):1911–1925, 1997.

M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.

N. Vyas, D. Morwani, R. Zhao, M. Kwun, I. Shapira, D. Brandfonbrener, L. Janson, and S. Kakade. SOAP: Improving and stabilizing Shampoo using Adam. *arXiv preprint arXiv:2409.11321*, 2024.

H. Zhu, Z. Liu, Q. Zhang, J. Fu, W. Deng, Z. Ma, Y. Guo, and X. Cao. FINER++: building a family of variable-periodic functions for activating implicit neural representation. *arXiv preprint arXiv:2407.19434*, 2024.

N. Doikov. Primal acceleration of Newton's method. *arXiv preprint arXiv:2608.21359*, 2026.

V. S. Amaral, R. Andreani, E. G. Birgin, D. S. Marcondes, and J. M. Martínez. On complexity and convergence of high-order coordinate descent algorithms for smooth nonconvex box-constrained minimization. *Journal of Global Optimization*, 84(3):527–561, 2022.

J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

L. Zancato, A. Achille, A. Ravichandran, R. Bhotika, and S. Soatto. Predicting training time without training. arXiv preprint arXiv:2008.12478, 2020.

S. Fort, G. K. Dziugaite, M. Paul, S. Kharaghani, D. M. Roy, and S. Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

K. Luo, H. Wen, S. Hu, Z. Sun, Z. Liu, M. Sun, K. Lyu, and W. Chen. A multi-power law for loss curve prediction across learning rate schedules. In *International Conference on Learning Representations (ICLR)*, 2025.

A. Pourali, B. Boukani, and H. Khazaei. CAPE: generalized convergence prediction across architectures without full training. *Transactions on Machine Learning Research*, 2026.

S. Wang, X. Yu, and P. Perdikaris. When and why PINNs fail to train: a neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.

K. Riedl, K. Spiliopoulos, and J. Sirignano. Convergence analysis of Newton's method for neural networks in the overparameterized limit. *arXiv preprint arXiv:2605.08352*, 2026.

S. Jiang, A. Voronin, E. C. Cyr, and B. Southworth. On the convergence behavior of preconditioned gradient descent toward the rich learning regime. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2601.03162.

L. de Moura and S. Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, Lecture Notes in Computer Science, pages 625–635, 2021. doi:10.1007/978-3-030-79876-5_37.

The mathlib Community. The Lean mathematical library. In *Certified Programs and Proofs (CPP)*, 2020.

Appendix

Supplementary Analyses and Experimental Evaluations

A Deferred Proofs

Statements summarized in prose in Sections 3 and 5 are restated here in full, each followed by its proof. Algebraic cores of the *introduced* blockwise arguments (Proposition A.10, Theorems B.8–B.10, Proposition 5.1, and the Cauchy estimate of Section 5.2, not the restated full space cubic lemmas) were checked by the Lean 4 kernel as described in Appendix F.

Lemma A.1 (Secular equation)**.** *Define, for* $\lambda > \max\{0, -\lambda_n\}$,

$$s(\lambda) := -(\mathbf{H} + \lambda \mathbf{I})^{-1} g, \qquad \varphi(\lambda) := \|s(\lambda)\| = \Big(\sum_{i=1}^{n} \frac{c_i^2}{(\lambda_i + \lambda)^2}\Big)^{1/2}, \quad c_i := \langle u_i,\, g \rangle\,.$$

Then φ *is continuous, strictly decreasing, and* $\varphi(\lambda) \to 0$ *as* $\lambda \to \infty$*. If* g *has a nonzero component in the eigenspace of* λ_n *(for a simple* λ_n*,* $c_n \neq 0$*) or* $\lambda_n \geq 0$*, the equation*

$$\lambda = \tfrac{M}{2}\, \varphi(\lambda) \tag{10}$$

has a unique solution $\lambda^\star > \max\{0, -\lambda_n\}$*, and* $s^\star = s(\lambda^\star)$ *with* $r^\star = \varphi(\lambda^\star)$ *is the global minimizer of the cubic model Eq. (3).*

Proof. Each term $c_i^2/(\lambda_i + \lambda)^2$ with $c_i \neq 0$ is positive and strictly decreasing in λ on the admissible domain, and at least one such term exists since $g \neq 0$. Hence φ is strictly decreasing, and continuity and the limit are immediate. Consider $\psi(\lambda) := \frac{M}{2}\varphi(\lambda) - \lambda$, which is strictly decreasing with $\psi(\lambda) \to -\infty$. If $\lambda_n < 0$ and g has a nonzero component in its eigenspace then $\varphi(\lambda) \to \infty$ as $\lambda \downarrow -\lambda_n$, so $\psi \to +\infty$ and a unique root exists by the intermediate value theorem. If $\lambda_n \geq 0$, then at $\lambda \downarrow 0$ either $\varphi(0^+) > 0$ (so $\psi(0^+) > 0$, giving a root) or $g = 0$, excluded by assumption. Global optimality of the resulting pair $(s^\star, \lambda^\star)$ for Eq. (3) is the characterization in Section 5 in Nesterov and Polyak (2006), that any s with $g + \mathbf{H}s + \frac{M}{2}\|s\|\, s = 0$ and $\mathbf{H} + \frac{M}{2}\|s\|\, \mathbf{I} \succeq 0$ globally minimizes m. Our $\lambda^\star > -\lambda_n$ ensures the second condition with strict inequality. □

Lemma A.2 (Positive definiteness before the polynomial)**.** *The solution of Eq. (10) satisfies*

$$r^\star \;\geq\; \frac{2\max\{0, -\lambda_n\}}{M}, \tag{11}$$

and consequently, the shifted operator $\mathbf{H} + \frac{Mr^\star}{2}\mathbf{I}$ *is positive semidefinite. It is positive definite whenever* $\lambda^\star > -\lambda_n$*, which holds in the non-degenerate case of Lemma A.1.*

Proof. By Lemma A.1, $\lambda^\star = \frac{M}{2} r^\star$ lies in the admissible domain $\lambda^\star > \max\{0, -\lambda_n\}$, which is Eq. (11) after rearranging. The smallest eigenvalue of the shifted operator is $\lambda_n + \lambda^\star > 0$. □

Proposition A.3 (Degree bound)**.** *Let* $\lambda^\star = \frac{M}{2} r^\star$ *solve Eq. (10). Then*

$$\lambda^\star \;\geq\; \lambda_{\mathrm{lo}} := \frac{\sqrt{\lambda_1^2 + 2M\,\|g\|} - \lambda_1}{2}, \tag{12}$$

and, whenever the gradient driven shift also dominates the negative curvature, in the sense that

$$\lambda_{\mathrm{lo}} \;\geq\; 2\max\{0, -\lambda_n\}, \tag{13}$$

the effective condition number of the shifted operator obeys

$$\kappa_{\mathrm{eff}} := \frac{\lambda_1 + \lambda^\star}{\lambda_n + \lambda^\star} \;\leq\; \frac{2\big(\lambda_1 + \lambda_{\mathrm{lo}}\big)}{\max\{\lambda_n, 0\} + \lambda_{\mathrm{lo}}} \qquad \text{(the factor 2 is not needed when } \lambda_n \geq 0\text{)}. \tag{14}$$

In particular, a Chebyshev (or Lanczos) solver needs a degree $L = \mathcal{O}\big(\sqrt{\kappa_{\mathrm{eff}}}\, \ln \frac{1}{\varepsilon_{\mathrm{rel}}}\big)$ *for a relative-accuracy* $\varepsilon_{\mathrm{rel}}$ *solve, and:*

1. *(far from stationarity) if* $2M\|g\| \geq \lambda_1^2$ *and Eq. (13) holds, for graded non-convex* f *the latter is implied by* $\sqrt{M\|g\|/2} \geq \frac{2\sigma_\tau}{\sqrt{2}-1}$*, since* $-\lambda_n \leq \sigma_\tau$ *(Remark B.1), then* $\lambda_{\mathrm{lo}} \geq (\sqrt{2}-1)\sqrt{M\|g\|/2}$ *and* $\kappa_{\mathrm{eff}} \leq 2 + 2\lambda_1/\lambda_{\mathrm{lo}} = 2 + \mathcal{O}\big(\lambda_1/\sqrt{M\|g\|}\big)$*, so the required degree is* $\mathcal{O}(1)$ *independent of* $\kappa = \lambda_1/\lambda_n$*;*

2. *(near stationarity) if* $\lambda_n > 0$ *(a neighborhood of a strict second-order stationary point), then as* $\|g\| \to 0$ *the shift vanishes and* $\kappa_{\mathrm{eff}} \to \lambda_1/\lambda_n$*, recovering the undamped Chebyshev–Newton case (the pure Newton step computed by a Chebyshev inner solve) with the Chebyshev polynomial decay bound* $L \approx 1.28\sqrt{\kappa}$ *and its fast local convergence rate (residual contraction* ≤ 0.22 *per application of the degree-*L *polynomial, Remark C.1).*

Proof. From Eq. (4), $\|g\| = \|(\mathbf{H} + \lambda^\star \mathbf{I})s^\star\| \leq (\lambda_1 + \lambda^\star)\, r^\star = (\lambda_1 + \lambda^\star) \cdot \frac{2\lambda^\star}{M}$. Rearranging gives the quadratic inequality $2(\lambda^\star)^2 + 2\lambda_1\lambda^\star - M\|g\| \geq 0$, whose positive root yields $\lambda^\star \geq \frac{-\lambda_1 + \sqrt{\lambda_1^2 + 2M\|g\|}}{2} = \lambda_{\mathrm{lo}}$, which is Eq. (12). This estimation argument follows Section 5 in Nesterov and Polyak (2006), where it bounds the step norm. Here we use it to bound the shift. The map $t \mapsto (\lambda_1 + t)/(\lambda_n + t)$ is non-increasing for $t > -\lambda_n$ (its derivative has the sign of $\lambda_n - \lambda_1 \leq 0$), and $\lambda^\star \geq \lambda_{\mathrm{lo}}$ with $\lambda^\star > -\lambda_n$ (Lemma A.2), so

$$\kappa_{\mathrm{eff}} \;\leq\; \frac{\lambda_1 + \lambda_{\mathrm{lo}}}{\lambda_n + \lambda_{\mathrm{lo}}} \qquad \text{whenever } \lambda_{\mathrm{lo}} > -\lambda_n.$$

If $\lambda_n \geq 0$ the denominator equals $\max\{\lambda_n, 0\} + \lambda_{\mathrm{lo}}$, which is Eq. (14) without the factor 2. If $\lambda_n < 0$, Eq. (13) gives $-\lambda_n \leq \lambda_{\mathrm{lo}}/2$, hence $\lambda_n + \lambda_{\mathrm{lo}} \geq \lambda_{\mathrm{lo}}/2 = \frac{1}{2}\big(\max\{\lambda_n, 0\} + \lambda_{\mathrm{lo}}\big)$, which is Eq. (14). Some domination condition of this kind is necessary. With $\lambda_n < 0$ fixed and $c_n \to 0$, the secular solution approaches $-\lambda_n$ from above and $\kappa_{\mathrm{eff}} \to \infty$, the near-hard case, in which Krylov subsolvers fall back to the shift-independent sublinear case analyzed by Carmon and Duchi (2018) and thus the augmentation of Remark B.2 applies. For item 1, write $t := 2M\|g\|$. The assumption $t \geq \lambda_1^2$ gives

$$\begin{aligned}\lambda_{\mathrm{lo}} = \frac{\sqrt{\lambda_1^2 + t} - \lambda_1}{2} = \frac{t}{2\big(\sqrt{\lambda_1^2 + t} + \lambda_1\big)} \;\geq\; \frac{t}{2\big(\sqrt{2t} + \sqrt{t}\big)} = \frac{\sqrt{t}}{2(\sqrt{2}+1)} \\ = \frac{\sqrt{2}-1}{2}\sqrt{2M\|g\|} \;=\; (\sqrt{2}-1)\sqrt{M\|g\|/2},\end{aligned}$$

so $\lambda_{\mathrm{lo}} = \Theta(\sqrt{M\|g\|})$. The sufficient condition quoted in item 1 follows because $-\lambda_n \leq \sigma_\tau$ and $\lambda_{\mathrm{lo}} \geq (\sqrt{2}-1)\sqrt{M\|g\|/2} \geq 2\sigma_\tau$ together imply Eq. (13). Dropping the non-negative term $\max\{\lambda_n, 0\}$ in Eq. (14) gives $\kappa_{\mathrm{eff}} \leq 2 + 2\lambda_1/\lambda_{\mathrm{lo}}$. The degree bound for Chebyshev iteration on a positive definite system follows from the extremal property of Chebyshev polynomials. Among degree-L polynomials with $p(0) = 1$, the shifted Chebyshev polynomial minimizes the maximal residual on $[\lambda_{\min}, \lambda_{\max}]$, giving $\|e_L\| \leq 2\big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\big)^L \|e_0\|$, so degree $L \geq \frac{\sqrt{\kappa_{\mathrm{eff}}}+1}{2}\ln\frac{2}{\varepsilon}$ reduces the shifted residual by the factor ε (Saad, 2003). Item 2 concerns the local phase near a strict second-order stationary point, where $\lambda_n > 0$. Since $\varphi(\lambda^\star) = \big\|(\mathbf{H} + \lambda^\star \mathbf{I})^{-1} g\big\| \leq \|g\|/\lambda^\star$, the secular equation Eq. (10) gives $(\lambda^\star)^2 \leq \frac{M}{2}\|g\|$, so $\lambda^\star \to 0$ as $g \to 0$, while $\lambda_{\mathrm{lo}} \to 0$ by Eq. (12). Hence $\kappa_{\mathrm{eff}} \to \lambda_1/\lambda_n$. If $\lambda_n < 0$ persists as $g \to 0$, the shift remains bounded below by $-\lambda_n$ and the method does not reduce to the undamped Chebyshev–Newton case of item 2.

This is the intended behavior. The shift then keeps the shifted operator positive definite in the non-degenerate case. Lemma A.2 gives positive semidefiniteness unconditionally. Strict definiteness can fail only in the hard case of Lemma A.1, where $\lambda_n < 0$ and the gradient has no component along the bottom eigenvector, $c_n = \langle u_n, g\rangle = 0$, so that $\lambda^\star$ may equal $-\lambda_n$ and the shifted operator becomes singular exactly on the eigenspace of λ_n.

□

Lemma A.4 (Exact subspace reduction)**.** *For any* $y \in \mathbb{R}^L$ *and* $s = \mathbf{Q}_L y$*,*

$$m(\mathbf{Q}_L y) \;=\; \hat{m}(y) := \|g\|\, e_1^\top y + \tfrac{1}{2}\, y^\top \mathbf{T}_L\, y + \tfrac{M}{6}\,\|y\|^3 . \tag{15}$$

Consequently, if $y^\star = \arg\min_y \hat{m}(y)$ (computable by the secular solver of Lemma A.1 on the $L \times L$ eigendecomposition of $\mathbf{T}_L$), then $s_L := \mathbf{Q}_L y^\star$ is the exact *minimizer of the cubic model m over the Krylov subspace $\mathcal{K}_L(\mathbf{H}, g)$:*

$$m(s_L) \;=\; \min_{s \in \mathcal{K}_L(\mathbf{H},g)} m(s).$$

Proof. Using $\mathbf{Q}_L^\top \mathbf{Q}_L = \mathbf{I}$ we have $\langle g,\, \mathbf{Q}_L y\rangle = (\mathbf{Q}_L^\top g)^\top y = \|g\|\, e_1^\top y$; $\langle \mathbf{H}\mathbf{Q}_L y,\, \mathbf{Q}_L y\rangle = y^\top(\mathbf{Q}_L^\top \mathbf{H}\mathbf{Q}_L)y = y^\top \mathbf{T}_L y$; and $\|\mathbf{Q}_L y\| = \|y\|$. Substituting into Eq. (3) gives Eq. (15) exactly (no truncation error appears because all three terms involve $\mathbf{H}$ at most once, and the Lanczos relation $\mathbf{Q}_L^\top \mathbf{H}\mathbf{Q}_L = \mathbf{T}_L$ is exact regardless of the residual term in the three term recurrence). Since every $s \in \mathcal{K}_L$ is $\mathbf{Q}_L y$ for a unique y, minimizing $\hat{m}$ over y minimizes m over $\mathcal{K}_L$. □

This is the reduction of Gould et al. (1999) for the trust region subproblem, extended to the cubic model by Cartis et al. (2011a). It is restated here because Corollary A.5 and Proposition A.10 use it per block.

Corollary A.5 (CubicKrylov step dominates the fixed coefficient U_n step)**.** *Let d_L be any step produced by a polynomial of degree at most $L-1$ in $\mathbf{H}$ applied to g, in particular the depth-L Chebyshev second kind recurrence Eq. (22), whose polynomial has degree $L-2$. Then $d_L \in \mathcal{K}_L(\mathbf{H}, g)$ and*

$$m(s_L) \;\le\; m(d_L).$$

Proof. A polynomial step of degree at most $L-1$ is by definition $q(\mathbf{H})g \in \mathcal{K}_L(\mathbf{H}, g) = \mathrm{span}\{g, \ldots, \mathbf{H}^{L-1}g\}$, and s_L minimizes m over that set by Lemma A.4. (The shifted operator $\mathbf{H} + \lambda\mathbf{I}$ generates the same Krylov space as $\mathbf{H}$.) □

Remark A.6 (Lanczos as Gauss quadrature). Lanczos on $(\mathbf{H}_b, g_b)$ is a three term implementation of the same Krylov sequence the Chebyshev U_n recurrence generates, with coefficients computed from $(\mathbf{H}_b, g_b)$ rather than set in advance for a spectral interval. The Ritz values of $\mathbf{T}_L$ are the Gauss quadrature nodes for the g_b-weighted spectral measure $\mathrm{d}\mu(\lambda) = \sum_i (u_i^\top g_b)^2 \delta(\lambda - \lambda_i)$ (Golub and Van Loan, 2013), so they integrate polynomials of degree less than $2L$ against $\mathrm{d}\mu$ exactly. The secular solve on $\mathbf{T}_L$ then takes the exact cubic minimizer in that space (Lemma A.4). The U_n recurrence is the minimax residual polynomial for an assumed interval (Remark C.1). Lanczos is the sequence of orthogonal polynomials for this block's measured $\mathrm{d}\mu$. Modes orthogonal to g_b have mass zero in $\mathrm{d}\mu$ and are not in the subspace (Remark B.2). This is optimality over $\mathcal{K}_L(\mathbf{H}_b, g_b)$, not among L-dimensional subspaces, and it is a statement about the block model, not a trajectory of f.

Two auxiliary lemmas support Theorem 3.1. Both are the one-step estimates of Nesterov and Polyak (2006), Section 4, written in our notation.

Lemma A.7 (Model decrease)**.** *Let $s^\star$ be the global minimizer of Eq. (3) with $r^\star = \|s^\star\|$. Then*

$$m(s^\star) \;\le\; -\tfrac{M}{12}\,(r^\star)^3. \tag{16}$$

If in addition $M \ge L_H$, then $f(x + s^\star) \le f(x) - \frac{M}{12}(r^\star)^3$.

Proof. Stationarity Eq. (4) gives $\langle g,\, s^\star\rangle = -\langle \mathbf{H}s^\star,\, s^\star\rangle - \frac{M}{2}(r^\star)^3$. Substituting,

$$m(s^\star) = -\tfrac{1}{2}\langle \mathbf{H}s^\star,\, s^\star\rangle - \tfrac{M}{2}(r^\star)^3 + \tfrac{M}{6}(r^\star)^3 = -\tfrac{1}{2}\langle \mathbf{H}s^\star,\, s^\star\rangle - \tfrac{M}{3}(r^\star)^3.$$

The second-order condition $\mathbf{H} + \frac{Mr^\star}{2}\mathbf{I} \succeq 0$ implies $\langle \mathbf{H}s^\star,\, s^\star\rangle \ge -\frac{M}{2}(r^\star)^3$, hence $m(s^\star) \le \frac{M}{4}(r^\star)^3 - \frac{M}{3}(r^\star)^3 = -\frac{M}{12}(r^\star)^3$. For the second claim, by Eq. (2) and $M \ge L_H$, $f(x + s^\star) \le f(x) + m(s^\star) + \frac{L_H - M}{6}(r^\star)^3 \le f(x) + m(s^\star)$. □

Lemma A.8 (New gradient bound)**.** *Under Assumption 2.1, the point $x^+ = x + s^\star$ satisfies*

$$\big\|\nabla f(x^+)\big\| \;\le\; \frac{L_H + M}{2}\,(r^\star)^2. \tag{17}$$

Proof. Using $g + \mathbf{H}s^\star = -\frac{Mr^\star}{2}s^\star$ from Eq. (4),

$$\nabla f(x^+) = \underbrace{\nabla f(x^+) - g - \mathbf{H}s^\star}_{\text{Taylor remainder}} - \tfrac{Mr^\star}{2}\, s^\star,$$

and Assumption 2.1 bounds the remainder by $\frac{L_H}{2}(r^\star)^2$ while the second term has norm $\frac{M}{2}(r^\star)^2$. □

Proof of Theorem 3.1. The argument is the telescoping count of Nesterov and Polyak (2006), Section 4. Fix K and suppose $\|\nabla f(x_i)\| > \varepsilon$ for all $1 \le i \le K$. By Lemma A.8, $r_i := \|s_i^\star\| \ge \left(\frac{2\|\nabla f(x_{i+1})\|}{L_H+M}\right)^{1/2} > \left(\frac{2\varepsilon}{L_H+M}\right)^{1/2}$ for $0 \le i \le K-1$. By Lemma A.7,

$$f(x_i) - f(x_{i+1}) \;\ge\; \tfrac{M}{12}\, r_i^3 \;>\; \tfrac{M}{12}\Big(\tfrac{2\varepsilon}{L_H+M}\Big)^{3/2}.$$

Telescoping over $i = 0, \dots, K-1$ (the sequence $\{f(x_i)\}$ is monotone by Lemma A.7):

$$f(x_0) - f^\star \;\ge\; K \cdot \tfrac{M}{12}\Big(\tfrac{2\varepsilon}{L_H+M}\Big)^{3/2},$$

so $K < \frac{12(f(x_0)-f^\star)}{M}\left(\frac{L_H+M}{2}\right)^{3/2}\varepsilon^{-3/2}$. Taking the ceiling gives the stated K. □

Corollary A.9 (Approximate second-order stationarity)**.** *Under the assumptions of Theorem 3.1, for any $\varepsilon > 0$ there is an iterate x_i, $1 \le i \le K$, with K exactly as in Theorem 3.1, satisfying* both *(the second-order statement of Nesterov and Polyak, 2006, Section 4, and of Cartis et al., 2011b, in our constants)*

$$\|\nabla f(x_i)\| \le \varepsilon \qquad \textit{and} \qquad \lambda_n\big(\nabla^2 f(x_i)\big) \;\ge\; -\frac{M + 2L_H}{\sqrt{2(L_H+M)}}\,\sqrt{\varepsilon}\,,$$

where λ_n denotes the smallest eigenvalue. With the canonical choice $M = 2L_H$ the curvature threshold is $-\frac{4}{\sqrt{6}}\sqrt{L_H\,\varepsilon}$.

The method therefore converges to approximate second-order *stationary points. At a strict saddle the smallest Hessian eigenvalue stays below some fixed $-c < 0$ that does not shrink as $\varepsilon \to 0$, so for ε small enough the curvature condition above rules the saddle out as a stopping point. The method keeps moving and escapes the saddle instead of stopping just because the gradient norm is small.*

Proof. Set $\rho := \left(\frac{2\varepsilon}{L_H+M}\right)^{1/2}$ and suppose the conclusion fails for every $1 \le i \le K$, that is, either $\|\nabla f(x_i)\| > \varepsilon$ or $\lambda_n(\nabla^2 f(x_i)) < -(\frac{M}{2} + L_H)\rho$, noting that $(\frac{M}{2} + L_H)\rho = \frac{M+2L_H}{\sqrt{2(L_H+M)}}\sqrt{\varepsilon}$. In the first case Lemma A.8 gives $r_{i-1} \ge \left(\frac{2\|\nabla f(x_i)\|}{L_H+M}\right)^{1/2} > \rho$. In the second case, the second-order condition of the cubic subproblem at x_{i-1}, $\nabla^2 f(x_{i-1}) + \frac{Mr_{i-1}}{2}\mathbf{I} \succeq 0$, combined with Assumption 2.1 and Weyl's inequality, yields

$$\lambda_n\big(\nabla^2 f(x_i)\big) \;\ge\; \lambda_n\big(\nabla^2 f(x_{i-1})\big) - L_H\, r_{i-1} \;\ge\; -\Big(\tfrac{M}{2} + L_H\Big) r_{i-1},$$

so again $r_{i-1} > \rho$. In either case Lemma A.7 gives $f(x_{i-1}) - f(x_i) \ge \frac{M}{12}\rho^3$, and telescoping over $i = 1, \dots, K$ as in the proof of Theorem 3.1 forces $K < \frac{12(f(x_0)-f^\star)}{M}\left(\frac{L_H+M}{2}\right)^{3/2}\varepsilon^{-3/2}$, contradicting the definition of K as the ceiling of this quantity. □

Proposition A.10 (Per-block cubic decrease)**.** *Let each block trial be accepted by the ratio test of Algorithm 1, phase B, namely $\rho_b = (f(x^{(b-1)}) - f(x^{(b-1)} + s_b))/(-m_b(s_b)) \ge \eta_1$ with $\eta_1 \in (0,1)$, where $x^{(b-1)}$ is the point after blocks $1, \dots, b-1$ have been updated and m_b is the block cubic model at that point, and let s_b be a global minimizer of m_b over $\mathbb{R}^{n_b}$ (exact step) or over $\mathcal{K}_L(\mathbf{H}_b, g_b)$ (CubicKrylov step, any $L \ge 1$). Then:*

(i) every accepted trial satisfies $f(x^{(b-1)}) - f(x^{(b)}) \ge \eta_1 \frac{M_b}{12}\|s_b\|^3$, so one sweep gives $f(x_{k+1}) \le f(x_k) - \frac{\eta_1}{12}\sum_b M_b \|s_b\|^3$ and the sweep sequence is monotone;

(ii) a trial with $M_b \geq L_{H,b}$ (the Hessian Lipschitz constant of the block restriction $f_b(s) := f(\dots, \theta_b + s, \dots)$) is accepted; hence $M_b \leq M_{\max} := \gamma \max(L_{H,b}, M_b^{\text{init}})$ throughout, and the rejected trials of block b over a run number at most $A_b + \log_\gamma(M_{\max}/M_{\min})$, A_b being its accepted trials.

Proof. (i) The ratio test gives $f(x^{(b-1)}) - f(x^{(b)}) \geq \eta_1(-m_b(s_b))$. The proof of Lemma A.7 uses only the stationarity and second-order conditions of the minimizer, $P(g_b + \mathbf{H}_b s_b + \lambda_b s_b) = 0$ and $P\mathbf{H}_b P + \lambda_b P \succeq 0$ with $\lambda_b = \frac{M_b}{2}\|s_b\|$ and P the projector onto the search space ($P = \mathbf{I}$ for the exact step, the projector onto $\mathcal{K}_L$ for the Krylov step, Lemma A.4). Taking the inner product of the first with $s_b = Ps_b$ and inserting the second gives $m_b(s_b) \leq -\frac{M_b}{12}\|s_b\|^3$ in both cases. Summing over the blocks of a sweep gives the rest. (ii) This is the bounded regularizer and unsuccessful iteration count of Cartis et al. (2011a), applied to the block restriction. The cubic upper bound of Assumption 2.1 for f_b gives $f(x^{(b-1)} + s_b) \leq f(x^{(b-1)}) + m_b(s_b) + \frac{L_{H,b} - M_b}{6}\|s_b\|^3 \leq f(x^{(b-1)}) + m_b(s_b)$ when $M_b \geq L_{H,b}$, so $\rho_b \geq 1 \geq \eta_1$. A rejection therefore happens only at $M_b < L_{H,b}$ and leaves $M_b < \gamma L_{H,b}$, and acceptances never raise M_b. With U_b rejections and A_b acceptances, each rejection multiplying M_b by γ and each acceptance dividing it by at most γ inside $[M_{\min}, M_{\max}]$, $U_b \log\gamma - A_b \log\gamma \leq \log(M_{\max}/M_{\min})$. □

Algorithm 1 evaluates the ratio with the stabilized tolerance t added to numerator and denominator, so an accepted trial decreases f by at least $\eta_1(-m_b(s_b)) - (1-\eta_1)\,t$. The additive term is below the fp32 resolution of the loss and is dropped in (i). Phase A applies the exact cubic step with $M_b = 6p_bL_H$ from the sweep start gradient and keeps it only if the loss does not rise, so Phase A is monotone by its guard. Its decrease bound is a hypothesis. Under the fresh gradient assumption of Section 3.4, and taking the lazy exact decrease of Doikov et al. (2023) as given (Appendix F), (ii) would accept the step and (i) holds with $\eta_1 = 1$. Phase B retries a rejection in the same sweep, so every large block that is not skipped with $g_b = 0$ contributes an accepted summand.

Lemma A.11 (Per-step escape bound)**.** *Let $M_{\text{amp}} \geq 2$ and define the clamped multiplier by replacing $x = h\lambda$ with $\hat{x} = \max\big(h\lambda, -\ln M_{\text{amp}}\big)$ in (7):*

$$\tilde{\eta}_h(\lambda) \;=\; \frac{1 - e^{-\hat{x}}}{\lambda} \quad (\lambda \neq 0), \qquad \tilde{\eta}_h(0) \;=\; h. \tag{18}$$

Then for every mode, $|\tilde{\eta}_h(\lambda)\,\lambda| \leq M_{\text{amp}} - 1$, so the amplification of any gradient component in one step is at most $(M_{\text{amp}} - 1)/|\lambda|$, and the step is finite in fp32 for any h and any spectrum.

Proof. Monotonicity of $x \mapsto 1 - e^{-x}$ and the floor $\hat{x} \geq -\ln M_{\text{amp}}$ give $1 - M_{\text{amp}} \leq 1 - e^{-\hat{x}} \leq 1$, and $1 \leq M_{\text{amp}} - 1$ because $M_{\text{amp}} \geq 2$, so $|1 - e^{-\hat{x}}| \leq M_{\text{amp}} - 1$ on every mode, and $\tilde{\eta}_h(\lambda)\,\lambda = 1 - e^{-\hat{x}}$. The default $M_{\text{amp}} = 10^6$ satisfies the hypothesis. □

The clamp trades per-step escape magnitude for compounded escape across accepted steps. k accepted steps amplify an escape direction by up to $(M_{\text{amp}} - 1)^k$ while every intermediate iterate passes the monotone guard. Empirically (Appendix K) the clamp engages in the first sweeps on saddle rich initializations and disengages once the trust adaptation settles.

B Deferred Remarks

Remark B.1 (Graded non-convexity bounds the curvature part of the shift). If $f \in \mathcal{F}_\tau$ is non-convex of grade τ in the sense of Doikov et al. (2024), then $-\lambda_n(x) \leq \sigma_\tau$ uniformly, so the part of the shift spent on enforcing positive definiteness is at most σ_τ, and the lower bound Eq. (11) costs at most $r \geq 2\sigma_\tau/M$. On problem classes with small spectral tails (large τ-grade), the size of the cubic shift is therefore set by the gradient driven term $\Theta(\sqrt{M\,\|g\|})$ of Proposition A.3 (stated below), and not by the positive definiteness requirement $\lambda^\star > -\lambda_n$, mirroring how σ_τ^+ replaces σ_τ in Theorem 5.1 in Doikov et al. (2024).

Remark B.2 (Hard case). If $g \perp u_n$ and $\lambda_n < 0$ (the *hard case* of the trust region subproblem, Cartis et al., 2011a), no polynomial in $\mathbf{H}$ applied to g contains a component along u_n, so the Krylov solution approaches the hard-case boundary solution rather than the interior escape direction. The

usual remedy is to augment the subspace with an approximate eigenvector of λ_n (one extra power or Lanczos probe, the same $-\lambda_n$ probe that the Chebyshev second kind recurrence solver of Appendix C implements). This restores the saddle escape mechanism. Our implementation exposes this as an optional augmentation vector.

Remark B.3 (Inexactness and adaptivity). (i) When s_k only minimizes m over $\mathcal{K}_L$ (Lemma A.4) instead of $\mathbb{R}^n$, the rate is retained under the ARC termination criteria (Cartis et al., 2011a;b), i.e. the residual test (19), not the Chebyshev degree of Proposition A.3. The implementation grows the Lanczos basis until that test holds or $L_{\max}$ is reached. The U_n bound is a ceiling on how large that L can be when κ_{eff} is controlled by the cubic shift. In the extreme case $L = 1$, $\mathcal{K}_1 \ni$ the Cauchy point already yields a (slower) guaranteed decrease. The same applies to Corollary A.9. Its proof uses the subproblem second-order condition, which the Krylov step certifies only within $\mathcal{K}_L$, and Cartis et al. (2011b) show that the ARC criteria (with a Lanczos-based subproblem solver, as here) preserve the second-order complexity bound as well. (ii) L_H is never known. As in Algorithm 3 in Doikov et al. (2023) and in the adaptive search ensuring Eq. (20) in Doikov et al. (2024), we run a multiplicative adaptation of M. Algorithm 1 (the stabilized code used for the FINER experiments) doubles M_b on a rejected trial and halves it on a very successful one ($\gamma = 2$). The Block-CN-Lazy code of the ViSIR runs (Appendix D) quadruples on a rejected step and halves on every accepted one, with a descent guard in place of the ratio test. In either case the accepted steps satisfy the descent inequality of Lemma A.7 with the current $M_k \le \gamma \max\{L_H, M_{\min}\}$, and the rejected steps cost one function evaluation each, at most doubling the total evaluation count. This multiplicative adaptation, applied *per block*, is the first of the two stabilization mechanisms of the blockwise optimizer. The second, the floor on M_b, is analyzed with the algorithm in Section 4. (iii) For *lazy* Hessians reused over m steps the appropriate constant is $M = 6mL_H$ (Doikov et al., 2023), which we exploit for the small block exact step of Section 4.

Remark B.4 (Global minima and the scope of the guarantee). Theorem 3.1 and Corollary A.9 guarantee approximate second-order stationarity and stop short of global minimality. This scope cannot be improved by a better method. By the information-theoretic lower bound of Nemirovski and Yudin (1983), any method that accesses a general smooth non-convex function through local oracle calls (values, gradients, Hessians, or higher derivatives) needs a number of calls growing exponentially with the dimension to locate an approximate global minimizer. On the class of Assumption 2.1 no method of any order can therefore provide a global-minimum guarantee, and the $\mathcal{O}(\varepsilon^{-3/2})$ rate to approximate stationarity is itself optimal for methods with a second-order oracle (Carmon et al., 2020). Global minimum guarantees exist only under additional structure that removes this obstacle. For gradient dominated objectives (a class containing convex, star-convex, and Polyak–Łojasiewicz functions), every stationary point is a global minimizer, and the same cubic method reaches an ε-global minimum with improved complexity (Chayti et al., 2024; Masiha et al., 2022). Sinusoidal INR landscapes are not gradient dominated. The saddle structure documented in Sections 1 and 6 is exactly what such assumptions exclude. So approximate second-order stationarity is the strongest guarantee any method can offer on the problem class studied here.

Remark B.5 (Failure of joint acceptance at scale). Suppose instead that all B block steps $s_1, \dots, s_B$ are computed from the same base point and accepted or rejected *jointly*. Each block step s_b is bounded by its own trust region ($\|s_b\| = r_b \approx \sqrt{2\,\|g_b\|\,/M_b}$), but the joint step has $\|s\| = (\sum_b r_b^2)^{1/2} \approx \sqrt{B}\,\bar{r}$, and the cross-block second-order coupling terms $\langle \mathbf{H}_{bb'} s_b,\ s_{b'} \rangle$, which are absent from every block model, grow with B as well. The joint cubic decrease guarantee would require the *joint* radius to satisfy the secular equation, i.e. each M_b inflated by $\Theta(\sqrt{B})$. With $B \approx 40$ large tensors (as in the 91.4M ViSIR) a joint test rejects nearly every step, and a shared multiplicative M adaptation then drives $M \to M_{\max}$, resulting in the stall observed experimentally (Section 6). Per-block acceptance with per-block M_b (Proposition A.10) costs one extra forward pass per block and avoids this kind of failure due to joint acceptance.

Proposition A.10 gives the decrease. The sweep end gradient is where the cross-block coupling of Remark B.5 appears. Throughout this section every block gradient g_b is evaluated at $x^{(b-1)}$, the point after blocks $1, \dots, b-1$ have been updated (the assumptions of Section 3.4). Let $\delta \ge 0$ bound the off-block curvature, $\left\|\nabla^2 f - \operatorname{blkdiag} \nabla^2 f\right\| \le \delta$ on the sublevel set $\{f \le f(x_0)\}$, and let $\kappa_H \ge \|\mathbf{H}_b\|$ there. Write $r_b = \|s_b\|$ for accepted steps ($r_b = 0$ otherwise) and $R_k = (\sum_b r_b^2)^{1/2}$ for the joint step norm of one sweep. A claim on ∇f needs every visited block to be accepted or skipped with $g_b = 0$. Phase A applies an exact step or skips at $g_b = 0$. Phase B does the same

once a rejection is retried until accept. Theorems B.8 and B.9 bound $g_{\mathcal{S}}$ and $g_{\mathcal{L}}$ separately, and Theorem B.10 assembles both at the end of the sweep.

Lemma B.6 (Block gradient after its own step)**.** *For an accepted exact step, $\|g_b(x^{(b)})\| \le c_b r_b^2$ with $c_b = \frac{L_{H,b}+M_b}{2}$. For an accepted Krylov step that satisfies the termination criterion*

$$\|\nabla m_b(s_b)\| \le \kappa_\theta \min(1, r_b) \left\|g_b(x^{(b-1)})\right\|, \qquad \kappa_\theta \in (0,1), \tag{19}$$

the same holds with $c_b = \frac{L_{H,b}+M_b}{2} + \frac{\kappa_\theta}{1-\kappa_\theta}\left(\kappa_H + \frac{M_b}{2}\right)$.

Proof. Write $g = g_b(x^{(b-1)})$, $\mathbf{H} = \mathbf{H}_b(x^{(b-1)})$, $s = s_b$. By Assumption 2.1 on f_b, $\left\|g_b(x^{(b)}) - g - \mathbf{H}s\right\| \le \frac{L_{H,b}}{2} r_b^2$, and $\nabla m_b(s) = g + \mathbf{H}s + \lambda_b s$ with $\|\lambda_b s\| = \frac{M_b}{2} r_b^2$, so $\left\|g_b(x^{(b)})\right\| \le \frac{L_{H,b}+M_b}{2} r_b^2 + \|\nabla m_b(s)\|$. The exact step has $\nabla m_b(s) = 0$ (Lemma A.8). For the Krylov step, (19) and $\|g\| \le \|\nabla m_b(s)\| + \kappa_H r_b + \frac{M_b}{2} r_b^2$ give $\|g\| \le (\kappa_H r_b + \frac{M_b}{2} r_b^2)/(1-\kappa_\theta)$, and $\min(1, r_b) \le r_b$ against the first summand, $\min(1, r_b) \le 1$ against the second, give $\|\nabla m_b(s)\| \le \frac{\kappa_\theta}{1-\kappa_\theta}(\kappa_H + \frac{M_b}{2}) r_b^2$. □

Criterion (19) is the step form of the termination test of Cartis et al. (2011b), and the Krylov case of the lemma is their new gradient estimate under that test, written for the block restriction. Its left side is free for the Lanczos step, $\|\nabla m_b(s_L)\| = \beta_{L+1} |e_L^\top y^\star|$, and the implementation records whether each trial met it (Appendix S).

Lemma B.7 (Sweep end gradient)**.** *Let $\mathcal{J}$ be the blocks of a phase (or of a full sweep), $R_{\mathcal{J}} = (\sum_{b\in\mathcal{J}} r_b^2)^{1/2}$, and $g_{\mathcal{J}}(x)$ the concatenation of $g_b(x)$ over $b \in \mathcal{J}$. If every accepted step in $\mathcal{J}$ satisfies Lemma B.6 with $c_b \le \bar{C}$ and every skipped block in $\mathcal{J}$ has $\|g_b\| \le \tau$ at its visit, then at the point x^+ after those visits*

$$\left\|g_{\mathcal{J}}(x^+)\right\| \le C_1^{\mathcal{J}} R_{\mathcal{J}}^2 + \sqrt{|\mathcal{J}|}\,\delta\, R_{\mathcal{J}} + \sqrt{|\mathcal{J}|}\,\tau, \qquad C_1^{\mathcal{J}} = \bar{C} + \sqrt{|\mathcal{J}|}\, L_H.$$

In particular, after a full sweep with $\tau = 0$ and $\mathcal{J} = \{1, \dots, B\}$, $\|\nabla f(x_{k+1})\| \le C_1 R_k^2 + \sqrt{B}\,\delta\, R_k$ with $C_1 = \bar{C} + \sqrt{B}\, L_H$.

Proof. Block-diagonal extraction is a spectral norm contraction, since $\operatorname{blkdiag} A = \mathbb{E}[DAD]$ for a random diagonal D with entries ± 1 constant within blocks, so $\|\operatorname{blkdiag} A\| \le \|A\|$. Fix $b \in \mathcal{J}$ and let d_b be the displacement after the visit of b that is supported off block b (the later Gauss–Seidel steps of the sweep). Then $\|d_b\| \le R_{\mathcal{J}}$. Write $x^{(b)}$ for the point at which Lemma B.6 applies (or the visit point if b is skipped). Then $g_b(x^+) - g_b(x^{(b)}) = \int_0^1 \mathbf{U}_b^\top [\nabla^2 f(y_t) - \operatorname{blkdiag} \nabla^2 f(y_t)]\, d_b\, dt$ with $y_t = x^{(b)} + t d_b$, because the block-diagonal part annihilates d_b in row block b. The integrand differs from its value at $x^{(b)}$ by at most $2L_H t\, \|d_b\|$ in norm (Assumption 2.1 and the contraction), and $x^{(b)}$ lies in the sublevel set by Proposition A.10(i), so $\left\|g_b(x^+) - g_b(x^{(b)})\right\| \le \delta R_{\mathcal{J}} + L_H R_{\mathcal{J}}^2$. With Lemma B.6, $\|g_b(x^+)\| \le c_b r_b^2 + \delta R_{\mathcal{J}} + L_H R_{\mathcal{J}}^2$ on an updated block, and the same with $c_b r_b^2$ replaced by τ on a skipped block. Summing the squares over $\mathcal{J}$ and using $\sum_{b\in\mathcal{J}} r_b^4 \le R_{\mathcal{J}}^4$ gives the claim. □

Theorem B.8 (Phase A rate)**.** *Let Assumption 2.1 hold, let every g_b, $b \in \mathcal{S}$, be evaluated at $x^{(b-1)}$, and take the lazy exact step of Doikov et al. (2023) as a hypothesis. An exact cubic step with $M_b = 6 p_b L_H$ from a Hessian at most p_b sweeps old satisfies Proposition A.10(i) with $\eta_1 = 1$ and the gradient bound of Lemma B.6 with some $c_b \le \bar{C}$. For $p_b = 1$ the second part is Lemma B.6 itself. For $p_b > 1$ the stale Hessian adds the term $L_H\, r_b \left\|x^{(b-1)} - x_{\mathrm{snap}}\right\|$, with x_{snap} the point of the last Hessian rebuild, to the new block gradient. Doikov et al. (2023) absorb that term in their potential, and it is part of the hypothesis here. After Phase A of Algorithm 1, every $b \in \mathcal{S}$ has taken such a step or was skipped with $g_b = 0$. Write $x^{(A)}$ for the point after Phase A, $M_A = \min_{b\in\mathcal{S}} 6 p_b L_H$, $C_1^{\mathcal{S}}$ as in Lemma B.7 with $\mathcal{J} = \mathcal{S}$, and $\varepsilon_\star^{\mathcal{S}} = 2|\mathcal{S}|\delta^2 / C_1^{\mathcal{S}}$. Then $f(x_k) - f(x^{(A)}) \ge M_A R_{\mathcal{S}}^3 / (12\sqrt{|\mathcal{S}|})$, and for every $\varepsilon \ge \varepsilon_\star^{\mathcal{S}}$ some Phase A index $k < K_A(\varepsilon)$,*

$$K_A(\varepsilon) = \left\lceil \frac{12\sqrt{|\mathcal{S}|}\,(2C_1^{\mathcal{S}})^{3/2}\,(f(x_0) - f^\star)}{M_A\, \varepsilon^{3/2}} \right\rceil = \mathcal{O}\left(\sqrt{|\mathcal{S}|}\,\varepsilon^{-3/2}\right),$$

produces $\left\|g_{\mathcal{S}}(x^{(A)})\right\| \le \varepsilon$.

Proof. By the lazy decrease hypothesis, $f(x_k) - f(x^{(A)}) \geq (M_A/12) \sum_{b\in\mathcal{S}} r_b^3$. The power mean over $|\mathcal{S}|$ blocks gives the $R_{\mathcal{S}}^3$ bound. Summing over $k < K$ and using $f \geq f^\star$ (Phase B decreases only tighten the telescope) yields $\min_{k<K} R_{\mathcal{S}}^3 \leq 12\sqrt{|\mathcal{S}|}(f(x_0) - f^\star)/(M_A K)$, so at $K = K_A(\varepsilon)$ some Phase A has $R_{\mathcal{S}} \leq (\varepsilon/2C_1^{\mathcal{S}})^{1/2}$. There Lemma B.7 with $\tau = 0$ gives $\|g_{\mathcal{S}}(x^{(A)})\| \leq \varepsilon/2 + \sqrt{|\mathcal{S}|}\,\delta\,(\varepsilon/2C_1^{\mathcal{S}})^{1/2} \leq \varepsilon$, the last step being $\varepsilon \geq \varepsilon_\star^{\mathcal{S}}$. □

Theorem B.9 (Phase B rate)**.** *Let Assumption 2.1 hold, let every accepted CubicKrylov step satisfy Lemma B.6 with (19), and let $M_{\min} \leq M_b \leq M_{\max}$. After Phase B of Algorithm 1 with an unbounded rejection cap, every $b \in \mathcal{L}$ has been accepted or skipped with $g_b = 0$. A rejection restores the block and the inner loop retries until Proposition A.10(ii) forces acceptance. Write $C_1^{\mathcal{L}}$ as in Lemma B.7 with $\mathcal{J} = \mathcal{L}$ and $\varepsilon_\star^{\mathcal{L}} = 2|\mathcal{L}|\delta^2/C_1^{\mathcal{L}}$. Then $f(x^{(A)}) - f(x_{k+1}) \geq \eta_1 M_{\min} R_{\mathcal{L}}^3/(12\sqrt{|\mathcal{L}|})$, and for every $\varepsilon \geq \varepsilon_\star^{\mathcal{L}}$ some Phase B index $k < K_B(\varepsilon)$,*

$$K_B(\varepsilon) = \left\lceil \frac{12\sqrt{|\mathcal{L}|}\,(2C_1^{\mathcal{L}})^{3/2}\,(f(x_0) - f^\star)}{\eta_1 M_{\min}\,\varepsilon^{3/2}} \right\rceil = \mathcal{O}\big(\sqrt{|\mathcal{L}|}\,\varepsilon^{-3/2}\big),$$

produces $\|g_{\mathcal{L}}(x_{k+1})\| \leq \varepsilon$.

Proof. Rejected trials revert, so only the accepted step of each $b \in \mathcal{L}$ contributes to the decrease. Proposition A.10(i) and the power mean over $|\mathcal{L}|$ blocks give the $R_{\mathcal{L}}^3$ bound. The same telescope as in Theorem B.8, now on the Phase B summands, produces a sweep with $R_{\mathcal{L}} \leq (\varepsilon/2C_1^{\mathcal{L}})^{1/2}$, and Lemma B.7 with $\tau = 0$ closes as there. □

Theorem B.10 (Blockwise complexity with a coupling floor)**.** *Let the hypotheses of Theorems B.8 and B.9 hold (Assumption 2.1, every g_b evaluated at $x^{(b-1)}$, the lazy exact step hypothesis in Phase A, decrease and gradient bound, Lemma B.6 for every accepted Phase B step, exact or CubicKrylov with (19), $M_{\min} \leq M_b \leq M_{\max}$ throughout, and $M_A \geq \eta_1 M_{\min}$). After both phases of a sweep of Algorithm 1, every block is accepted or skipped with $g_b = 0$, so Lemma B.7 applies to ∇f with $\tau = 0$. With C_1 as in that lemma and $\varepsilon_\star = 2B\delta^2/C_1$, for every $\varepsilon \geq \varepsilon_\star$ some sweep $k < K(\varepsilon)$,*

$$K(\varepsilon) = \left\lceil \frac{12\sqrt{B}\,(2C_1)^{3/2}\,(f(x_0) - f^\star)}{\eta_1 M_{\min}\,\varepsilon^{3/2}} \right\rceil = \mathcal{O}\big(\sqrt{B}\,\varepsilon^{-3/2}\big),$$

produces x_{k+1} with $\|\nabla f(x_{k+1})\| \leq \varepsilon$. If in addition every accepted step is exact, or is a Krylov step with the negative curvature check of Cartis et al. (2011b), then also $\lambda_n(\nabla^2 f(x_{k+1})) \geq -(\frac{M_{\max}}{2} + L_H)(\varepsilon/2C_1)^{1/2} - \delta$. For a Krylov step without that check the second-order condition is certified on $\mathcal{K}_L$ only (Remark B.11(iii)). At $B = 1$, $\delta = 0$ the theorem recovers the $\varepsilon^{-3/2}$ exponent of Theorem 3.1 and Corollary A.9, with a looser constant (C_1 still includes the within-sweep remainder term L_H). Theorems B.8 and B.9 do not by themselves give $\|\nabla f\| \leq \varepsilon$ at $k = \max(K_A, K_B)$. Phase B displaces $g_{\mathcal{S}}$, and the two phases may meet their tolerances at different sweeps.

Proof. Phase A contributes at least $(M_A/12)\sum_{b\in\mathcal{S}} r_b^3$ and Phase B at least $(\eta_1 M_{\min}/12)\sum_{b\in\mathcal{L}} r_b^3$. By the hypothesis $M_A \geq \eta_1 M_{\min}$, Proposition A.10(i) and the power mean $\sum_b r_b^3 \geq B^{-1/2}(\sum_b r_b^2)^{3/2}$ give $f(x_k) - f(x_{k+1}) \geq \frac{\eta_1 M_{\min}}{12\sqrt{B}} R_k^3$. Summing over $k < K$ and using $f \geq f^\star$, $\min_{k<K} R_k^3 \leq 12\sqrt{B}(f(x_0) - f^\star)/(\eta_1 M_{\min} K)$, so at $K = K(\varepsilon)$ some sweep has $R_k \leq (\varepsilon/2C_1)^{1/2}$. There Lemma B.7 with $\tau = 0$ gives $\|\nabla f(x_{k+1})\| \leq \frac{\varepsilon}{2} + \sqrt{B}\,\delta\,(\varepsilon/2C_1)^{1/2} \leq \varepsilon$, the last step being $\varepsilon \geq \varepsilon_\star = 2B\delta^2/C_1$. For the curvature clause, the second-order condition of each accepted step gives $\lambda_n(\mathbf{H}_b(x^{(b-1)})) \geq -\frac{M_b}{2} r_b$ on the full block (exact step, or Krylov step with the negative curvature check). Assumption 2.1 extends it to $x^{(B)}$ at a cost $L_H R_k$, and Weyl's inequality with the δ bound gives $\lambda_n(\nabla^2 f(x^{(B)})) \geq \min_b \lambda_n(\mathbf{H}_b(x^{(B)})) - \delta$. □

Algorithm 1 realizes the fresh gradient hypothesis in Phase B and, with $\mathcal{S} = \emptyset$ (the FINER configuration, every tensor through Phase B), for the whole sweep. With $\mathcal{S} \neq \emptyset$ its Phase A takes the small block gradients from the sweep start closure and is monotone by its descent guard, so the sweep stays monotone, while the Phase A decrease bound of Theorem B.8 is proved for the fresh gradient variant.

Remark B.11 (Scope of the theorem). (i) Below $\varepsilon_\star$ the linear coupling term dominates the sweep end gradient and the argument yields no further decrease. The block models cannot see the off-block curvature that keeps ∇f away from zero. The floor is measurable before training, $\delta \le \left\|\nabla^2 f\right\|_F \sqrt{1-\beta}$ with $\beta = \left\|\mathrm{blkdiag}\,\nabla^2 f\right\|_F^2 / \left\|\nabla^2 f\right\|_F^2$ the tensor-block mass (Appendix S reports the FINER values). This β is not the coordinate-diagonal ratio ρ of Section 6.2. (ii) The rigorous C_1 includes the term $\sqrt{B}L_H$ from the second-order remainder along the sweep. (iii) For the Krylov subsolver the second-order condition is certified on $\mathcal{K}_L$ only, through the smallest Ritz value of $\mathbf{T}_L$ against the shift. Cartis et al. (2011b) recover the full space clause for Lanczos-based ARC with an occasional negative curvature check, which applies per block here. (iv) The argument is the inexact-Hessian analysis of Xu et al. (2019) with the block-diagonal model as the inexact Hessian and δ its error, plus the sequential Gauss–Seidel sweep, which is the source of the $\sqrt{B}$ factors. The published block cubic rates, $\mathcal{O}(\varepsilon^{-2})$ under greedy block selection (Cristofari, 2026) and $\mathcal{O}(\varepsilon^{-3})$ for general block choice (Amaral et al., 2022), hold without a floor but with the weaker exponent. (v) The φ_1 step is not a model minimizer. Its decrease is the Cauchy type bound of Section 5.2, and Appendix S reports the gradient-regularized horizon that places its positive-mode step in the $\mathcal{O}(\varepsilon^{-3/2})$ framework of regularized Newton methods (Mishchenko, 2023; Doikov and Nesterov, 2023; Gratton et al., 2023).

Remark B.12 (Derivation of the step rule (5)). Write $m(x) = f(x_0) + \langle g,\, x - x_0\rangle + \frac{1}{2}\langle x - x_0,\, G\,(x - x_0)\rangle$ for the local quadratic model, so its gradient flow is $\dot{x} = -\nabla m(x) = -g - G\,(x - x_0)$ with $x(0) = x_0$. Substituting $u(\tau) = x(\tau) - x_0$ gives the linear constant-coefficient system $\dot{u} = -g - G\,u$, $u(0) = 0$. Multiplying by the integrating factor $e^{G\tau}$ and observing $\frac{d}{d\tau}\big(e^{G\tau}u\big) = e^{G\tau}(\dot{u} + Gu) = -e^{G\tau}g$, integration over $[0, h]$ yields

$$e^{Gh}\,u(h) \;=\; -\Big(\int_0^h e^{G\tau}\,d\tau\Big)\,g \quad\Longrightarrow\quad u(h) \;=\; -\int_0^h e^{-G(h-\tau)}\,d\tau\;g \;=\; -\int_0^h e^{-G\sigma}\,d\sigma\;g,$$

after the change of variable $\sigma = h - \tau$ (all factors are functions of the single symmetric matrix G and commute). This is exactly $s = -H(G,h)\,g$ with $H(G,h) = \int_0^h e^{-G\tau}\,d\tau$. When G is invertible the integral evaluates in closed form to $H(G,h) = G^{-1}\big(\mathbf{I} - e^{-hG}\big)$, and comparison with $\varphi_1(z) = (e^z - 1)/z$ gives $H = h\,\varphi_1(-hG)$. For singular G the same identity holds through the entire-function series $\varphi_1(z) = \sum_{k\ge 0} z^k/(k+1)!$, so the step requires no invertibility assumption. Projecting the same computation onto an eigenpair (λ_i, v_i) of G reduces it to the scalar ODE $\dot{u}_i = -\langle v_i,\, g\rangle - \lambda_i u_i$ and produces the per-mode multiplier $\eta_h(\lambda)$ of Proposition 5.1.

Finally, the literal interpretation of infinitesimal learning rate. Fix the horizon h, split it into n explicit gradient descent steps of size $\eta = h/n$ on the frozen quadratic, $u_{k+1} = u_k - \eta\,(g + Gu_k)$, $u_0 = 0$, and let n grow so that the step size shrinks toward zero at fixed total budget h. The iteration unrolls to the geometric sum

$$u_n \;=\; -\eta\sum_{j=0}^{n-1}(\mathbf{I} - \eta G)^j\,g \;=\; -G^{-1}\Big(\mathbf{I} - \big(\mathbf{I} - \tfrac{h}{n}G\big)^n\Big)g \;\xrightarrow{\;n\to\infty\;}\; -G^{-1}\big(\mathbf{I} - e^{-hG}\big)\,g,$$

for invertible G (the general case follows by continuity of both sides in G), since $(\mathbf{I} - \frac{h}{n}G)^n \to e^{-hG}$. The φ_1 step is the limit of n gradient descent steps of total budget h on the quadratic, integrated in closed form rather than iterated.

C Chebyshev Second Kind Preconditioning in the DSO Construction

Chebyshev polynomials of the second kind are defined by $U_n(\cos\zeta) = \sin((n+1)\zeta)/\sin\zeta$ with the recurrence $U_0 = 1$, $U_1(x) = 2x$, $U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x)$. The Dynamic Spectral Optimizer (DSO) of Podorozhny (2026) implements this Chebyshev second kind preconditioning. The construction is restated here in full because the OP26 presentation has no accompanying paper, and every DSO result in this paper is a run of this restatement. DSO normalizes the Hessian, $G = \mathbf{H}/\rho$ with $\rho = \|\mathbf{H}\|_2$, maps eigenvalues $\lambda \in [0, 1]$ of G to angles via $\cos\zeta = 1 - 2\lambda$, and builds

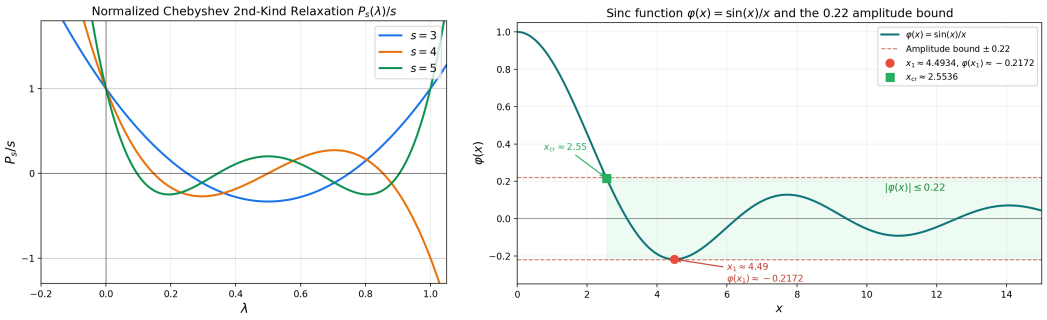

Figure 2: The DSO relaxation function. Left, $R_s(\lambda) = P_s(\lambda)/s$ for $s = 3, 4, 5$ on the normalized spectrum. Each curve has $R_s(0) = 1$, oscillates with small amplitude on the interior of $[0, 1]$, returns to $(-1)^{s-1}$ at $\lambda = 1$, and grows without bound for $\lambda < 0$. Right, the sinc function $\varphi(x) = \sin(x)/x$ that governs R_s near both ends of the spectrum through $x = 2s\sqrt{\lambda}$. Its first minimum is $\varphi(x_1) \approx -0.217$ at $x_1 \approx 4.49$, and $|\varphi(x)| \le 0.22$ for all $x \ge x_{\mathrm{cr}} \approx 2.55$ (shaded). Reaching $\lambda_{\min} = 1/\kappa$ with $2L\sqrt{\lambda_{\min}} \ge x_{\mathrm{cr}}$ gives the degree rule $L \ge (x_{\mathrm{cr}}/2)\sqrt{\kappa} \approx 1.28\sqrt{\kappa}$ of Remark C.1.

the *relaxation function*

$$R_L(\lambda) = \frac{P_L(\lambda)}{L}, \qquad P_L(\lambda) = \frac{\sin(L\zeta)}{\sin\zeta}, \tag{20}$$

a degree-$(L-1)$ polynomial with $R_L(0) = 1$. The induced preconditioner

$$H(\lambda) = \frac{1 - R_L(\lambda)}{\lambda} \tag{21}$$

is a polynomial of degree $L - 2$ that approximates $1/\lambda$ on most of the normalized spectrum. Figure 2 shows why. Near $\lambda = 0$ the angle is $\zeta \approx 2\sqrt{\lambda}$, so $R_L(\lambda) \approx \varphi(2L\sqrt{\lambda})$ with the sinc function $\varphi(x) = \sin(x)/x$, and $|R_L(1 - \lambda)| = |R_L(\lambda)|$ mirrors this behavior at the upper end. The sinc function satisfies $|\varphi(x)| \le 0.22$ for every $x \ge x_{\mathrm{cr}} \approx 2.55$, its first minimum being $\varphi(4.49) \approx -0.217$. On the band $x_{\mathrm{cr}}^2/(4L^2) \le \lambda \le 1 - x_{\mathrm{cr}}^2/(4L^2)$ the polynomial therefore stays small, with $\max|R_L| = 0.33$, 0.27, 0.25 for $L = 3, 4, 5$ and 0.22 from $L \approx 15$ on. One application of the degree-L polynomial (one sweep of $L-2$ HVPs) contracts the residual along every eigendirection in that band to at most this fraction of its value, and $H(\lambda)$ is within the same fraction of $1/\lambda$ there. This uniform contraction across the band, rather than at the spectral extremes only, is the mechanism that equalizes contraction rates across frequencies and counteracts spectral bias when the recurrence is used as a standalone optimizer. The two ends of the spectrum behave differently. As $\lambda \to 0$ (flat directions, the high frequency modes of an INR) $R_L \to 1$ and $H(\lambda) \to H_L(0) = 2(L^2 - 1)/3$, so the gradient signal along a flat direction is amplified by a factor that grows with L^2 and stays finite, whereas Newton's method would divide by λ itself. At $\lambda = 1$ (the highest curvature direction) $R_L(1) = (-1)^{L-1}$, so $H(1)$ is 0 for odd L and 2 for even L, a bounded overshoot along the top eigenvector. For $\lambda < 0$ (negative curvature) $\cos\zeta > 1$, the angle becomes imaginary, and R_L grows like a hyperbolic sine, which is the automatic saddle escape of the standalone DSO and the reason its steps need a clamp. The step $d_L = -H(G)\tilde{g}$ is computed by the three term recurrence

$$d_{s+1} = \frac{2s}{s+1}(\mathbf{I} - 2G)\, d_s - \frac{s-1}{s+1}\, d_{s-1} - \frac{4s}{s+1}\,\tilde{g}, \qquad s \ge 2, \qquad d_1 = 0, \quad d_2 = -2\tilde{g}, \tag{22}$$

so that $d_s = -H_s(G)\,\tilde{g}$ for every s, where $H_s(\lambda) := (1 - R_s(\lambda))/\lambda$ is Eq. (21) at degree s (in particular $H_1 \equiv 0$ and $H_2 \equiv 2$, and starting the same coefficients one index earlier, from $(0, -2\tilde{g})$, would leave the U_n/n relaxation family and break the $R_L(0) = 1$ normalization). The recurrence requires one HVP per degree and $\mathcal{O}(1)$ auxiliary vectors.

Two structural costs of this construction motivate the present work:

1. **Degree blow-up.** A uniform per-direction contraction over the normalized Hessian eigenvalue spectrum requires degree $L \approx 1.28\sqrt{\kappa}$, where κ is the Hessian condition number, because the contraction band of Figure 2 must reach $\lambda_{\min} = 1/\kappa$ (Remark C.1). DSO implementation bounds L heuristically when κ is extreme (e.g. due to inaccuracy of eigenvalues estimation at scale).

2. **Indefiniteness.** For $\lambda < 0$ the recurrence enters hyperbolic growth. This provides automatic saddle escape but also requires clamping safeguards whose failure modes are documented in the DSO ablations.

Remark C.1 (The Chebyshev second kind degree bound at a fixed contraction). The constant 1.28 is $x_{\mathrm{cr}}/2$ of Figure 2, half the smallest argument beyond which the sinc envelope of R_L stays inside ± 0.22, rounded up. It can also be derived from the degree bound for Chebyshev iteration on a positive definite system (Saad, 2003), instantiated at a fixed per-sweep contraction. For a spectrum in $[\lambda_{\min}, \lambda_{\max}]$ the optimal degree-L residual polynomial satisfies $\|e_L\| \le 2\big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\big)^L \|e_0\|$, and since $\ln \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \ge \frac{2}{\sqrt{\kappa}}$, the contraction target $\|e_L\| \le \varepsilon \, \|e_0\|$ is met by

$$L \;\ge\; \tfrac{1}{2} \ln\!\big(\tfrac{2}{\varepsilon}\big)\sqrt{\kappa} \;=\; c(\varepsilon)\,\sqrt{\kappa}.$$

Setting $c = 1.28$ gives $\varepsilon = 2e^{-2c} \approx 0.15$, so the Chebyshev second kind degree bound $L \approx 1.28\sqrt{\kappa}$ is exactly this bound with the worst case residual contracted to $\approx 15\%$ per polynomial application. This is consistent with the ≈ 0.22 sinc envelope of Eq. (20), which measures the same per-degree contraction in the second kind normalization. By Proposition A.3, the cubic shift replaces the κ in this bound by the bounded κ_{eff} of Eq. (14). So the same bound yields a degree that remains constant independent of κ while the iterate is far from stationarity and the gradient driven shift dominates the negative curvature (Eq. (13)), where the large gradient keeps the shift large and κ_{eff} bounded.

D ARC-block Design Decisions and Computational Cost Evaluation

Algorithm 1 embeds the step into the block optimizer used for many experiments described in this paper (e.g. the ViSIR Chebyshev ablations). Three design decisions, each in response to an observed failure, are emphasized below.

(a) Two phase update with guarded small blocks. Small blocks (biases and norms, $n_b \le 512$) take lazy exact cubic steps (Doikov et al., 2023) from the sweep start gradient. Each step is kept only if the loss after it is finite and not above the loss before it. A rejected small block is restored and its exact Hessian is rebuilt at the next visit. Without this guard a single non-finite or overshooting small block step (a stale lazy Hessian against a fresh gradient spike at a batch size transition) poisons the parameters and every later step. The per-block accept-or-reject of Phase B is measured against the loss *after Phase A*. Rejecting the whole step on a loss increase lets one stale small block component veto good Krylov updates and drives M to the bound.

(b) Per-block acceptance and per-block M_b (Remark B.5). A single joint test over large tensors rejects even when every individual block step is good (for the experiments in this paper, this failure mostly was observed on 91.4M ViSIR architecture with 45 tensors). Per-block testing costs one forward pass per a large block and restores monotone progress (Proposition A.10).

(c) A nearly zero floor for M_b. The step radius scales as $r \approx \sqrt{2\,\|g_b\|\,/M_b}$ (Proposition A.3). A floor $M_{\min} = 6L_H$ over-regularizes blocks whose true local Hessian Lipschitz constant is far below the global estimate L_H. Steps are *accepted but microscopic*, a second, "silent" stall mode (no prominent warning signs, the optimizer making extremely slow progress). We set $M_{\mathrm{init}} = 6L_H$ but let the adaptation, upward by a factor γ on a rejected trial and downward by the same factor on a very successful one, take M_b all the way down to a nearly zero floor, retained only to keep the step radius finite, so each block finds its own curvature scale. Two codes realize this. The Block-CN-Lazy code of the ViSIR and ViSIR-Nano runs uses a descent guard in place of the ratio test, multiplies M_b by 4 on a rejected step and by $\frac{1}{2}$ on every accepted one, and has $M_{\min} = 10^{-6}$. The stabilized code of Algorithm 1 and the FINER experiments uses the ratio test with $\eta_1 = 0.1$, $\eta_2 = 0.9$, multiplies $M_b = 2\sigma_b$ by 2 on a rejected trial and by $\frac{1}{2}$ on a very successful one, and has $M_{\min} = 2\sigma_{\min} = 2 \cdot 10^{-8}$, $M_{\max} = 2\sigma_{\max} = 2 \cdot 10^{12}$. The asymmetric factors of the first code keep the rejected step overhead bounded (Remark B.3(ii)).

Fresh gradients. Phase B gradients are recomputed after Phase A and after each preceding large block (Algorithm 1, Phase B). They come free from the HVP graph construction. Phase A keeps the sweep start gradient for its small blocks, and only its Hessians are lazy. On the ViSIR-Nano benchmark (small scale, around 15k parameters), reusing the sweep start closure gradient in Phase B

as well (a single gradient evaluated before Phase A, stale for every subsequent block) lowers the final peak signal-to-noise ratio (PSNR) by ≈8.5 dB (31.7 vs. 23.1 dB in Table 2).

D.1 Computational cost evaluation

Table 1: Per-step, per-block cost of the step variants, block size n_b, Lanczos depth L (bounded by the Chebyshev degree analysis), laziness p_b. The gradient equivalent count follows the ViSIR ablation experiment. One gradient, one HVP, and one function evaluation each cost one, and the CubicKrylov HVPs are L for the basis and one for the model decrease.

Step	Build	Solve	Cost
Cheby-ON (DSO step)	full $\mathbf{H}_b$ every step (n_b gevals)	degree-L recurrence, $\mathcal{O}(Ln_b^2)$	$n_b + 1$
Cheby-OFF (Newton)	full $\mathbf{H}_b$ every step	`linalg.solve`, $\mathcal{O}(n_b^3)$	$n_b + 1$
Block-CN-Lazy (eigh)	full $\mathbf{H}_b$ every p_b steps	cached eigh + secular, $\mathcal{O}(n_b^2)$	$n_b/p_b + 1$
CubicKrylov	**none**	L+1 **HVPs + tridiagonal secular**	$\mathbf{L + 3}$

For $L \approx 10$, the Krylov row is the only entry whose oracle cost is independent of n_b. Its wall clock and basis memory still scale linearly with n_b through the HVPs and the stored vectors. It is this independence that makes it feasible for large block sizes. Cubic regularized second-order steps become feasible on tensors where only HVPs are affordable, e.g., the SIREN decoder of the 91.4M ViSIR.

Cost per sweep. One sweep visits B blocks. Each block trial spends one gradient, at most L Hessian–vector products for the Lanczos basis, one for the model decrease, and one function evaluation, that is L+3 gradient equivalents, each Hessian–vector product costing one backward pass over the whole model regardless of block size. A sweep therefore costs $\mathcal{O}(BL)$ gradient equivalents and time linear in the parameter count n at fixed B and L. Rejected trials are included at full cost. Memory is L+1 block sized vectors for the Lanczos subsolver (the basis and one work vector) and three in place buffers for the Chebyshev and φ_1 recurrences, independent of the depth. Measured cost follows this form. On FINER (B=10, n=198,915, Section 6.3) the L=25 ARC-φ_1 run on ESM image 0 costs 276.6 gradient equivalents per sweep (2,366,167 over 8,556 sweeps) against the predicted $10 \times (25 + 3) = 280$. At L=163 an instrumented run of the converged configuration costs 1,710 per sweep over its first 42 sweeps (71,840 gradient equivalents, 11% of trials rejected, 1,630 per sweep once the first sweep's rejections are excluded) against the predicted $10 \times (163 + 3) = 1{,}660$ at full depth. The adaptive stop of Section 3.2 ends some builds below L. Per-sweep wall clock rises 7.2× between L=25 (0.84 s) and L=163 (6.08 s) against a depth ratio of 6.5× and a gradient equivalent ratio of 6.2×. Because B grows with architecture (10 tensors at 199k parameters, 45 at 91.4M), B is a controlled quantity rather than a constant, and the comparisons of Section 6 report gradient equivalents beside wall clock time.

E Trust-Controlled Horizons, Horizon Ranking, and the ARC-φ_1 Algorithm

E.1 Trust-controlled horizons and inherited guarantees

The composite scheme sets $h_b = c_h/\sigma_b$ and evaluates the acceptance ratio against the subspace quadratic model $m_b(s) = \langle g_b,\, s\rangle + \frac{1}{2}\langle s,\, \mathbf{H}_b s\rangle$ (one extra Hessian–vector product), with the stabilized monotone guard of Section 4 unchanged. Three consequences:

1. *Well posed adaptation.* Rejection multiplies σ_b by γ_2, hence contracts h_b geometrically. By Proposition 5.1(v) a sufficiently small horizon brings the cubic remainder below the model decrease, and the stabilized tolerance converts this into acceptance after finitely many rejections under the bounded Hessian and Lipschitz continuity assumptions of Section 2. The per-block rejection bound and σ-bounds of the block-σ ablation (Appendix M) apply verbatim.
2. *Newton limit under sustained acceptance.* Sustained acceptance drives $\sigma_b \to \sigma_{\min}$, hence $h_b \to c_h/\sigma_{\min}$, so high curvature modes receive Newton steps while flat modes receive steps of length $\le (c_h/\sigma_{\min})\ \|g\|$, and acceptance is decided by the guard, not by a shift.
3. *No claimed rate.* We do not transfer the $\mathcal{O}(\varepsilon^{-3/2})$ complexity of the cubic model (Theorem 3.1) to the φ_1 rule. That rate is a property of the cubic model's overestimation argument. What is

Algorithm 2 ARC-φ_1, one Gauss–Seidel sweep

build loss and gradient graph; $f_0 \leftarrow f(x)$
for each block $b = 1, \dots, B$ with $\|g_b\| > \varepsilon_g$ **do**
  Lanczos: $(\mathbf{Q}_L, \mathbf{T}_L) \leftarrow \mathrm{Lanczos}(\mathbf{H}_b, g_b, L)$; $\mathbf{T}_L = \mathbf{V}\,\mathrm{diag}(\nu)\,\mathbf{V}^\top$
  $h_b \leftarrow c_h/\sigma_b$
  candidates $\mathcal{H} \leftarrow \{h_b\}$ (or $\{h_b/2, h_b, 2h_b\}$ if `h_rank`$= 3$)
  $s_b \leftarrow$ step (8) for the model-best $h \in \mathcal{H}$, clamped per Lemma A.11
  $\rho_b \leftarrow \big(f_0 - f(x + s_b) + t\big)\big/\big(-m_b(s_b) + t\big)$; monotone guard (6)
  if $\rho_b \geq \eta_1$ **then** accept; $\sigma_b \leftarrow \max(\gamma_1\sigma_b, \sigma_{\min})$ if $\rho_b \geq \eta_2$; rebuild graph
  else restore block; $\sigma_b \leftarrow \gamma_2\sigma_b$; retry, or skip after n_{rej} rejections

retained is the acceptance driven global behavior (monotone full loss trajectory up to the stabilized tolerance) and the saddle escape mechanism, now explicit in the step rule.

E.2 Free per-trial horizon ranking

Proposition 5.1(iv) is the φ_1 method's central mechanism, doubling the horizon until the objective stops improving. A grid search over horizons is affordable here for a stronger reason. Once the Ritz decomposition of (8) exists, any candidate h costs one diagonal reweighting and one L-dimensional model evaluation, with no Hessian–vector products and no function evaluations. The `h_rank` variant ranks $\{h_b/2,\ h_b,\ 2h_b\}$ by the subspace quadratic model and submits the winner to the (unchanged) acceptance test. On a small MLP benchmark the ranked variant selects the shorter horizon 68% of the time and the longer one 29%, confirming that the single-candidate scheme is frequently off its own model's optimum.

The extended hyperparameter sweep at the scale of the FINER benchmark (Section 6.3) is organized in rungs of increasing step budget. At each rung all candidate configurations are run at that budget, and only the best-ranked ones advance to the next rung, up to a final full budget confirmation run that decides the winner. The ranked variant never advanced that far. At every rung, the configurations that extended the Lanczos depth outranked the horizon-ranked variant.

E.3 ARC-block with φ_1 subsolver algorithm

Algorithm 2 follows the same per-block, Gauss–Seidel structure as Phase B of Algorithm 1 (Krylov build, search step, fresh-gradient accept-or-reject), replacing the CubicKrylov step and its M_b-doubling rule with the φ_1 step (8) and the σ_b-parameterized tiered acceptance rule of the stabilized block-σ ablation (Appendix M). Phase A (small blocks, lazy exact cubic) is unaffected and unused for the φ_1 comparisons of Section 6, which hold the block partition and Gauss–Seidel order fixed and vary only the large block step rule. Defaults in the current implementation are $L = 25$ at scale of evaluated INR architectures, c_h and σ_0 from the per-architecture sweep, $M_{\mathrm{amp}} = 10^6$, stabilized tolerances and σ-bounds as in the block-σ ablation of Appendix M. A trial costs $L + 3$ gradient equivalents, namely one gradient, L Hessian–vector products for the basis, one for the model decrease, and one function evaluation. Gradient equivalent counting, graph rebuild policy on accepted steps, and the small block vs large block routing threshold are identical to the CubicKrylov step method, so wall clock and geval comparisons at equal L isolate the step rule performance.

F Machine-Checked Algebraic Cores

The restated full space cubic lemmas (Lemma A.1, Theorem 3.1, Corollary A.9) follow the published arguments of Nesterov and Polyak (2006); Doikov et al. (2024) and were not passed to the Lean kernel. The introduced blockwise claims use a short list of real algebraic steps, namely scalar KKT relations, finite sums, modal identities, a logarithmic bound on rejected trials, a subset power-mean, leftover inversion at a coupling floor, the φ_1 multiplier law, and a cubic remainder comparison. We wrote those steps, and only those steps, in Lean 4 (de Moura and Ullrich, 2021) with mathlib (mathlib Community, 2020).

The correctness test is the Lean kernel as used in the NanoProof arrangement. Each core is a theorem in the `ArcBlock` library. An independent build (`lake build ArcBlock`) is the only test, since

the kernel accepts a theorem only when its proof is complete. The library contains no `sorry`, `admit`, or `native_decide`. The kernel decides the Lean statement. A translation that omits a displayed constant or reverses an inequality is an error in the Lean statement, and a claim with such an error is left out of the library. The build on Lean 4.27.0 with mathlib v4.27.0 accepted all 70 theorems listed below.

Assumptions that the paper uses but does not prove by algebra remain hypotheses, namely Assumption 2.1 on f (encoded as a landscape hypothesis, not derived), the published cubic remainder (2) of Nesterov and Polyak (2006), Lanczos producing $\sum_i c_i^2 = \|g_b\|^2$, the Cartis–Gould–Toint test (19), the lazy exact step of Doikov et al. (2023) (its decrease and its new gradient bound for a stale Hessian), Weyl's inequality, and an explicit M_b trajectory over the run. Appendix S reports, for each trial, whether (19) held. That measurement is not a Lean inequality. We do not claim that the kernel establishes the assembled paper theorems under those hypotheses. The cores below are the algebraic steps that the proofs of Theorems B.8–B.10, the CubicKrylov leftover of Lemma B.6, and the φ_1 decrease of Proposition 5.1 and Section 5.2 actually use.

F.1 Blockwise rate

Let us recall the argument of Theorems B.8, B.9, and B.10. An accepted block trial satisfies the first- and second-order KKT relations of the cubic model on its subspace (exact, or Krylov of any degree). Those two scalar identities give $m(s) - f \le -(M/12)r^3$ (`model_decrease_on_subspace`), which is Lemma A.7 restricted to a coordinate subspace. The ratio test $\rho_b \ge \eta_1$ multiplies the model bound and yields the displayed one-block decrease of Proposition A.10(i) (`accepted_cubic_decrease`). Summing nonnegative summands over a phase, we obtain $(\eta/12)\sum_b M_b r_b^3$ and $f_{\mathrm{new}} \le f_{\mathrm{old}}$ (`sweep_cubic_decrease`, `sweep_cubic_monotone`).

The passage from $\sum r_b^3$ to the joint radius $R = (\sum r_b^2)^{1/2}$ is the power-mean inequality $R^3 \le \sqrt{n}\sum r^3$ on a subset of n blocks (`power_mean_cubes`). Combined with the sweep sum, this is $(\eta M/(12\sqrt{n}))R^3$ (`phase_cubic_from_power_mean`). Phase A takes the exact step at $\eta = 1$ and $M = M_A$ (`phaseA_decrease`). Phase B takes an accepted CubicKrylov trial at $\eta = \eta_1$ and $M = M_{\min}$ (`phaseB_decrease`). Since $M_A \ge \eta_1 M_{\min}$, the two phases add (`joint_phase_decrease`), and we obtain the displayed decrease of Theorem B.10, $f(x_k) - f(x_{k+1}) \ge \eta_1 M_{\min} R_k^3/(12\sqrt{B})$. Telescoping $f \ge f^\star$ inverts to the displayed sweep count $K(\varepsilon)$ without the ceiling (`sweep_radius_from_budget`).

It remains to pass from a small joint radius to $\|\nabla f\| \le \varepsilon$. The leftover algebra of Lemma B.7 is the comparison $\|g_{\mathcal{J}}\| \le C_1 R^2 + \sqrt{n}\,\delta R + \sqrt{n}\,\tau$ (`sweep_end_grad_tau`, where the paper's C_1 is `c1` in the library). The $\tau = 0$ case is `sweep_end_grad_tau0`. At $R \le \sqrt{\varepsilon/(2C_1)}$ and $\varepsilon_\star = 2n\delta^2/C_1$, this is $\|g\| \le \varepsilon$ (`tau0_grad_le_eps`). That invert is the $\mathcal{O}(\sqrt{B}\,\varepsilon^{-3/2})$ statement of Theorems B.8–B.10 down to the coupling floor $\varepsilon_\star$. The curvature clause of Theorem B.10 is the combination of the cubic second-order condition, Assumption 2.1 over radius R, and Weyl's inequality (`curvature_from_weyl`), evaluated at the same $R \le \sqrt{\varepsilon/(2c_1)}$ (`curvature_clause`).

The coupling integral that produces the leftover coefficients C_1 and δ, the fresh gradient at every block, the lazy exact decrease of Phase A (Doikov et al., 2023), and a finite retry in Phase B (Proposition A.10(ii)) stay hypotheses. Thus the kernel checks the decrease, the power-mean, the invert, and the curvature algebra. It does not re-prove the published cubic / Krylov / Weyl facts or the integral along the segment on f.

F.2 CubicKrylov leftover

Lemma B.6 is the only place where the CubicKrylov step affects the blockwise rate, through a leftover different from that of the exact step. Write $g = g_b(x^{(b-1)})$, $\mathbf{H} = \mathbf{H}_b(x^{(b-1)})$, and $s = s_b$. The cubic remainder of Assumption 2.1 and the shift $\lambda = Mr/2$ give $\|g_b(x^{(b)})\| \le ((L_{H,b} + M_b)/2)r_b^2 + \|\nabla m_b(s)\|$. The exact step has $\nabla m_b(s) = 0$, and we obtain the first display of the lemma (`blockgrad_exact`). For an accepted Krylov trial that meets (19), the same identity with $\min(1, r_b) \le r_b$ against the first summand and $\min(1, r_b) \le 1$ against the second produces the extra factor $\kappa_\theta/(1 - \kappa_\theta)(\kappa_H + M_b/2)$ (`blockgrad_krylov`). Criterion (19) itself is the step form of Cartis et al. (2011b). The library takes it as a hypothesis, exactly as the paper does.

Proposition A.10 applies to a CubicKrylov trial of any degree $L \geq 1$, because the KKT argument of `model_decrease_on_subspace` uses only the subspace minimizer, not a full space secular solve. The M-adaptation count of Proposition A.10(ii) is the comparison $M \geq L_{H,b}$ plus the cubic remainder (`cubic_model_upper_accepts`), the two one-step M updates (`accept_does_not_increase`, `reject_below_mmax`), and the logarithmic invert (`reject_count_bound`, `reject_count_explicit`). Algorithm 1 uses one factor $\gamma = 2$, which is the special case $\gamma' = \gamma$. Phase B of Theorem B.9 uses this count only to guarantee a finite retry. The rate itself is the accepted-step decrease of Section F.1.

F.3 φ_1 SUBSOLVER RATES

Proposition 5.1 is a statement about the multiplier $\eta_h(\lambda) = (1 - e^{-h\lambda})/\lambda$ ($\eta_h(0) = h$) and the matrix $H(G,h) = V \operatorname{diag}(\eta_h(\lambda_i))V^\top$. The Lean library records the five clauses as follows. (i) For $\lambda > 0$, $|\eta_h(\lambda) - 1/\lambda| = e^{-h\lambda}/\lambda$ (`erlaw_newton`). (ii) The value at zero is h (`erlaw_grad_at_zero`), and the two-term expansion about $\lambda = 0$ is exact (`erlaw_grad_expand`). (iii) For $\lambda < 0$, $\eta_h(\lambda) = (e^{h|\lambda|} - 1)/|\lambda|$ (`erlaw_saddle`). (iv) The doubling identity $\eta_{2h} = \eta_h(2 - \lambda\eta_h)$ holds on scalars (`erlaw_doubling`) and lifts to $H(G,2h) = H(G,h)\,(2\mathbf{I} - GH(G,h))$ under an orthogonal eigendecomposition (`erlaw_doubling_mat`, `prop_erlaw_iv`). (v) On the frozen quadratic $m(s) = \langle g, s\rangle + \frac{1}{2}\langle s, Gs\rangle$, the step $s = -H(G,h)g$ satisfies $m(s) < 0$ for every $h > 0$ with $g \neq 0$ (`erlaw_quad_descent`, `prop_erlaw_v`). Clauses (i)–(iv) are packaged as `prop_erlaw`.

Diagonalizing G and summing the per-mode identities produces the displayed decrease law (9). One mode is $-m(s) = h\,\varphi_1(-2h\lambda)\,g^2$ with $\varphi_1(z) = (e^z - 1)/z$ and $\varphi_1(0) = 1$ (`phi1_decrease_modal`). The factor is strictly positive for every λ (`phi1_decrease_pos`). Parseval $\langle g, f(G)g\rangle = \sum_i f(\lambda_i)c_i^2$ with $c = V^\top g$ (`parseval_spectral`) lifts the identity to $-m(-Hg) = h\,\langle g, \varphi_1(-2hG)\,g\rangle$ (`phi1_decrease_mat`, `eq_phi1decrease`), strictly positive if $g \neq 0$ (`eq_phi1decrease_pos`). This is an identity on the frozen quadratic.

The Cauchy type lower bound of Section 5.2 is the modal comparison $(1 - e^{-2h\theta})/(2\theta) \geq \frac{1}{4}\min(h, 1/\theta)$ (`modal_newton_limit`), the factor $1/(4\ell)$ when $\theta \leq \ell$ and $h \geq 1/\ell$ for a smoothness constant ℓ (`modal_le_smooth`, where the library writes it `L`), and Parseval (`modal_weighted_sum`). We obtain $\frac{1}{4}\|g\|^2 \min(h, 1/\|G\|)$ and $\|g\|^2/(4\ell)$ (`modal_sum_min_h`, `modal_sum_smooth`). An accepted trial multiplies by η_1 (`cauchy_accepted_factor`). Inverting a uniform floor m on $\min(h, 1/\|G\|)$ against a functional residual $f - f^\star$ gives the displayed first-order count $K \leq 4(f - f^\star)/(\eta_1 m\varepsilon^2)$ (`cauchy_eps2_count`), which is the $\mathcal{O}(\varepsilon^{-2})$ claim of Section 5.2 once Assumption 2.1 transfers the model decrease to f. Negative modes only increase the left-hand side, so the Lean statement uses the tight case of positive eigenvalues. Lanczos and $\sum_i c_i^2 = \|g_b\|^2$ remain hypotheses.

The basin rate of Appendix T (Lemma T.3, Theorem T.4) needs a relative bound from block gradients to ∇f along a Gauss–Seidel sweep. The one-block triangle $(x + y)^2 \leq 2x^2 + 2y^2$ (`sweep_triangle_one`), summed over an orthogonal prefix, produces the factor $2\ell^2 B$ (`sweep_triangle_to_hrel`). With $\|s_b\| \leq \|g_b\|/\mu$ we obtain the displayed $h_{\mathrm{rel}} = 2(1 + B\ell^2/\mu^2)$ (`sweep_hrel_from_triangle`, `sweep_rel_from_steps`). Per-block $\eta_1/(4\ell)$ sums (`sweep_phi1_telescope`) and inverts to $c_{\mathrm{sw}} = \eta_1/(8\ell(1 + B\ell^2/\mu^2))$ (`sweep_invert`). The potential comparison $V' \leq V - qV^{3/2}$ (the factor is written γ in the library) gives $1/\sqrt{V'} - 1/\sqrt{V} \geq q/2$ and the telescope $V_k \leq 4/(q(k - K_0) + 2/\sqrt{V_{K_0}})^2$ (`potential_step`, `potential_bound`). The $1/G_0$ downgrade, the smoothness envelope $C_\star = 2\sqrt{2\ell}/q$, and the Q-linear contraction $V' \leq (1 - 2\mu c)V$ are `phi1_cubic_downgrade`, `phi1_grad_envelope`, and `phi1_qlinear`. The basin assumption itself (Assumption T.1), which produces $\|s_b\| \leq \|g_b\|/\mu$ on f, is not in the library.

The transfer of Proposition 5.1(v) from m to f uses the working remainder (2), $f^+ \leq f + m + L_H r^3/6$ (`cubicBound`), as a hypothesis (`cubicFromLip`), since it is the published implication of Assumption 2.1 (Nesterov and Polyak, 2006), not derived here. Assumption 2.1 itself is the landscape hypothesis $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L_H \|x - y\|$ (`assLip`, `LipLandscape.hess_lip`). The norms are parameters, so the library does not replace the operator norm by the Frobenius norm. On the PSD side, $\eta_h(\lambda) \leq h$ (`etaH_le_h`) bounds the step energy by $h^2\|g\|^2$ (`erlaw_step_energy`). For

$h \le (1/10)/\|G\|$, the factor φ_1 is at least $\varphi_1(-1/5)$ (`phi1_floor_horizon`). The cubic remainder is then strictly smaller than that model decrease precisely when $L_H h^2 \sqrt{\sum c_i^2}/6 < \varphi_1(-1/5)$ (`erlaw_remainder_lt`). The comparison `erlaw_transfer` / `prop_erlaw_transfer` therefore keeps L_H explicit. The horizon $0.1/\|G\|$ does not cancel L_H. Lemma A.11 is the scalar bound $|\tilde{\eta}_h(\lambda)\lambda| \le M_{\text{amp}} - 1$ under $M_{\text{amp}} \ge 2$ (`clamp_amp`). The fp32 finiteness remark is not algebra.

F.4 Catalog of checked theorems

Model decrease and per-block cubic arguments. Lemma A.7 on a subspace, and the displayed parts of Proposition A.10.

Lean theorem	Paper claim
`model_decrease_on_subspace`	Lemma A.7: $m(s) \le -(M/12)r^3$ from subspace KKT
`accepted_cubic_decrease`	Prop. A.10(i): accepted ratio and the model bound give $\eta_1(M_b/12)r_b^3$
`sweep_cubic_decrease`	Prop. A.10(i): one-block bounds sum to $(\eta_1/12)\sum_b M_b r_b^3$
`sweep_cubic_monotone`	Prop. A.10(i): nonnegative summands, sweep monotone
`cubic_model_upper_accepts`	Prop. A.10(ii): $M \ge L_{H,b}$ and the cubic remainder give $\rho_b \ge 1 \ge \eta_1$
`accept_does_not_increase`	Prop. A.10(ii): $\max(M/\gamma, M_{\min}) \le M$
`reject_below_mmax`	Prop. A.10(ii): $\gamma M < \gamma \max(L_{H,b}, M^{\text{init}})$
`reject_count_bound`	Prop. A.10(ii): $U \log\gamma - A\log\gamma' \le \log(M_{\max}/M_{\min})$
`reject_count_explicit`	Prop. A.10(ii): $U \le (\log\gamma'/\log\gamma)A + \log_\gamma(M_{\max}/M_{\min})$

The U bound keeps a separate γ'. Algorithm 1 uses one factor $\gamma = 2$, which is the special case $\gamma' = \gamma$ stated in Proposition A.10(ii). The Lipschitz fact that produces the cubic remainder, and the M_b trajectory over the run, are not in the library.

φ_1 modal decrease and the Cauchy invert. The Cauchy lower bound of Section 5.2 and the $\mathcal{O}(\varepsilon^{-2})$ count.

Lean theorem	Paper claim
`modal_newton_limit`	one mode: $(1 - e^{-2h\theta})/(2\theta) \ge \frac{1}{4}\min(h, 1/\theta)$
`modal_le_smooth`	$\theta \le \ell$ and $h \ge 1/\ell$ give the factor $1/(4\ell)$
`modal_weighted_sum`	$\sum_i c_i^2 a_i \ge (\inf a_i)\sum_i c_i^2$
`modal_sum_min_h`	first bound: $\ge \frac{1}{4}\|g\|^2 \min(h, 1/\|G\|)$
`modal_sum_smooth`	second bound: $\ge \|g\|^2/(4\ell)$
`cauchy_accepted_factor`	accepted trial: $\eta_1/4$ times $\min(h, 1/\|G\|)$
`cauchy_eps2_count`	invert: $K \le 4(f - f^\star)/(\eta_1 m\varepsilon^2)$

Lanczos and the identity $\sum_i c_i^2 = \|g_b\|^2$ stay outside the library. They appear in the checked theorems as a hypothesis. Negative modes only increase the left-hand side, so the Lean statement uses the tight case of positive eigenvalues.

Sweep invert, triangle, and potential envelope. These relations close the basin rate of Appendix T (Lemma T.3, Theorem T.4) once the relative bound $\|\nabla f\|^2 \le 2(1 + B\ell^2/\mu^2)\sum_b \|g_b\|^2$ and the potential step $V' \le V - qV^{3/2}$ are assumed.

Lean theorem	Checked conclusion
`sweep_triangle_one`	$(x+y)^2 \le 2x^2 + 2y^2$ on one prefix term
`sweep_triangle_to_hrel`	orthogonal prefix energy gives the factor $2\ell^2 B$
`sweep_hrel_from_triangle`	triangle and $\|s_b\| \le \|g_b\| / \mu$ give $2(1 + B\ell^2/\mu^2)$
`sweep_phi1_telescope`	per-block $\eta_1/(4\ell)$ sums to the same factor on $\sum_b \|g_b\|^2$
`sweep_rel_from_steps`	$\|s_b\| \le \|g_b\| / \mu$ produces the factor $2(1 + B\ell^2/\mu^2)$
`sweep_invert`	those two bounds invert to $c_{\mathrm{sw}} = \eta_1/(8\ell(1+B\ell^2/\mu^2))$
`potential_step`	$V' \le V - qV^{3/2}$ gives $1/\sqrt{V'} - 1/\sqrt{V} \ge q/2$
`potential_bound`	the telescope $V_k \le 4/(q(k-K_0) + 2/\sqrt{V_{K_0}})^2$
`phi1_cubic_downgrade`	$\Delta V \ge c\|g\|^2$, $\|g\| \le G_0$, and $\|g\|^2 \ge 2\mu V$ give $\Delta V \ge (c/G_0)(2\mu)^{3/2} V^{3/2}$
`phi1_grad_envelope`	smoothness and the potential bound give $\|g\| \le C_\star/(n+k_\star)$ with $C_\star = 2\sqrt{2\ell}/q$
`phi1_qlinear`	without the $1/G_0$ step, $V' \le (1-2\mu c)V$

Phase A / Phase B / joint rate, leftover, and curvature. The subset power-mean, the sweep end leftover, the $\tau = 0$ invert, the Phase A / Phase B / joint decrease identities, the CubicKrylov leftover, the clamp, and the curvature clause of Theorem B.10.

Lean theorem	Checked conclusion
`power_mean_cubes`	$R^3 \le \sqrt{n} \sum r^3$ on a subset of n blocks
`sweep_end_grad_tau`	$\|g_{\mathcal{J}}\| \le C_1 R^2 + \sqrt{n}\,\delta R + \sqrt{n}\,\tau$
`sweep_end_grad_tau0`	the same with $\tau = 0$
`tau0_grad_le_eps`	$R \le \sqrt{\varepsilon/(2C_1)}$ and $\varepsilon_\star = 2n\delta^2/C_1$ give $\|g\| \le \varepsilon$
`phase_cubic_from_power_mean`	$(\eta M/12)\sum r^3$ and the power-mean give $(\eta M/(12\sqrt{n}))R^3$
`phaseA_decrease`	Theorem B.8: $\eta = 1$, $M = M_A$
`phaseB_decrease`	Theorem B.9: $\eta = \eta_1$, $M = M_{\min}$
`joint_phase_decrease`	Theorem B.10: $M_A \ge \eta_1 M_{\min}$ sums the two phases
`sweep_radius_from_budget`	displayed $K(\varepsilon)$ without the ceiling
`blockgrad_exact`	Lemma B.6: $((L_{H,b} + M_b)/2) r_b^2$
`blockgrad_krylov`	Lemma B.6: extra $\kappa_\theta/(1-\kappa_\theta)(\kappa_H + M_b/2)$
`clamp_amp`	Lemma A.11: $\|\tilde{\eta}_h(\lambda)\lambda\| \le M_{\mathrm{amp}} - 1$
`curvature_from_weyl`	Theorem B.10: SO, Lipschitz, and Weyl at radius R
`curvature_clause`	Theorem B.10: $R \le \sqrt{\varepsilon/(2C_1)}$ into the displayed curvature bound

Proposition 5.1 and equation (9). Scalar clauses, the matrix $H(G,h)$, and the named packages.

Lean theorem	Paper claim
`erlaw_newton`	Prop. 5.1(i): $\|\eta_h - 1/\lambda\| = e^{-h\lambda}/\lambda$ for $\lambda > 0$
`erlaw_grad_at_zero`	Prop. 5.1(ii): $\eta_h(0) = h$
`erlaw_grad_expand`	Prop. 5.1(ii): exact two-term remainder
`erlaw_saddle`	Prop. 5.1(iii): $(e^{h\|\lambda\|} - 1)/\|\lambda\|$
`erlaw_doubling`	Prop. 5.1(iv): $\eta_{2h} = \eta_h(2 - \lambda\eta_h)$
`erlaw_doubling_mat`	Prop. 5.1(iv): $H(2h) = H(h)(2\mathbf{I} - GH)$
`phi1_decrease_modal`	(9): one mode $-m(s) = h\,\varphi_1(-2h\lambda)\,g^2$
`phi1_decrease_pos`	(9): the factor is positive for every λ
`phi1_decrease_sum`	(9): modal sum of the identity
`parseval_spectral`	$\langle g,\, f(G)g\rangle = \sum_i f(\lambda_i)c_i^2$
`phi1_decrease_mat`	(9): $-m(-Hg) = h\,\langle g,\, \varphi_1(-2hG)\,g\rangle$
`phi1_decrease_mat_pos`	(9): the factor is positive if $g \neq 0$
`erlaw_quad_descent`	Prop. 5.1(v): $m(s) < 0$ for any $h > 0$
`prop_erlaw`	Prop. 5.1(i)–(iv), packaged
`prop_erlaw_iv`	Prop. 5.1(iv), matrix form
`prop_erlaw_v`	Prop. 5.1(v) on the quadratic
`eq_phi1decrease`	(9), packaged
`eq_phi1decrease_pos`	(9) sign, packaged

Landscape hypothesis and transfer to f. Assumption 2.1 as a hypothesis on f, the working remainder (2), and the $h \le 0.1/\|G\|$ comparison that keeps L_H.

Lean theorem	Paper claim
`assLip_apply`	Ass. 2.1: $\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| \le L_H \|x - y\|$
`landscape_cubic_bound`	(2) at a point, given `cubicFromLip`
`cubic_bound_strict`	remainder domination implies $f^+ < f$
`cubic_bound_point`	the same on a point (f, g, G)
`etaH_le_h`	Prop. 5.1(v): $\eta_h(\lambda) \le h$ if $\lambda \ge 0$
`phi1_floor_le`	$\varphi_1(-t) \ge \varphi_1(-1/5)$ on $[0, 1/5]$
`phi1_floor_horizon`	$h \le (1/10)/\|G\|$ gives that floor
`erlaw_step_energy`	$\sum_i (\eta_h(\lambda_i)c_i)^2 \le h^2 \sum_i c_i^2$
`erlaw_remainder_lt`	$L_H h^2 \sqrt{\sum c_i^2}/6 < \varphi_1(-1/5)$
`erlaw_transfer`	cubic bound and rem$<\Delta$ imply $f^+ < f$
`prop_erlaw_transfer`	Prop. 5.1(v) to f; 0.1 does not drop L_H

F.5 Hypotheses left outside the library

The triangle at the start of a sweep that would produce the relative bound on f, the basin of Assumption T.1 that produces $\|s_b\| \le \|g_b\|/\mu$, Lanczos producing $\sum_i c_i^2 = \|g_b\|^2$, a fresh gradient at every block, the lazy step's gradient bound for a stale Hessian, and an explicit M trajectory remain outside the library. In particular, the kernel does not establish Theorems B.8–B.10, the $\mathcal{O}(\varepsilon^{-2})$ claim of Section 5.2, or Theorem T.4 as statements about f alone, because those arguments use Assumption 2.1 on f and the published cubic / Krylov / Weyl facts that the library treats as hypotheses rather than re-proves. The named packages `prop_erlaw` and `eq_phi1decrease` are the algebraic cores of Proposition 5.1 and (9). The transfer `prop_erlaw_transfer` still requires (2).

G Subproblem Solver Verification

We compare the degree-L Krylov solver of Section 3.2 against the exact eigendecomposition based cubic solve on random symmetric matrices. *Positive definite*, $n = 200$. Machine precision is reached at $L = 5$, an empirical confirmation of Proposition A.3, since the cubic shift renders the spectrum benign enough for a degree-5 polynomial. *Indefinite*, $n \in \{200, 1000\}$. Relative model suboptimality 7.8×10^{-2} at $L = 5$, 4.3×10^{-5} at $L = 25$, $\sim 10^{-15}$ at $L = 50$, as higher-degree Krylov subspace iterates yield progressively better approximations as the hard-case is approached (Remark B.2).

H ViSIR-Nano Ablation (15,235 Parameters)

The purpose of the nano ablation is to make a direct comparison with the full Hessian cubic Newton methods possible at all. The original cubic Newton method (Nesterov and Polyak, 2006), its lazy Hessian and adaptive variants (Doikov et al., 2023), the first-order finite-difference implementation (Doikov and Grapiglia, 2023), and the stochastic subspace version SSCN (Zhao et al., 2025) all form their step by solving a regularized linear system in the (full or snapshot) Hessian. Even with their cost-reduction features, an m-step snapshot reuse or a finite-difference oracle, each Hessian build costs n gradient evaluations, the factorization behind the solve costs $\mathcal{O}(n^3)$ arithmetic, and the matrix itself requires $\mathcal{O}(n^2)$ memory (per-step complexity classes summarized in Appendix M). This makes them too expensive for networks of practical size. Already at $15\text{k} \times 15\text{k}$ the dense Hessian requires $\sim$1.8 GB in float64, a single build with its eigendecomposition takes minutes, and any larger scale is prohibitive, let alone the 91.4M parameter model of Section 6. ViSIR-Nano, a downsized ViSIR with 15,235 parameters, is chosen at this feasibility boundary, so that every cubic Newton variant above can run on the identical benchmark and be compared directly with the matrix-free CubicKrylov step.

The experimental setup is single image $3\times$ SR (ESM image #0, 100 steps, $L_H = 10$). All rows were measured on the same benchmark (shared init, seed 42, one idle RTX 6000). In the mechanism block, routing is identical across rows. Small blocks $n_b \le 512$ take lazy-eigh cubic steps with the per-block

accept-or-reject guard of Section 4 active in phase A throughout, so those rows differ only in the phase-B mechanism. The reference block runs the full Hessian methods evaluated elsewhere in the paper, granted twice the oracle budget of the final CubicKrylov configuration.

Table 2: Nano A/B comparison of large block step variants and full Hessian references (single image experiment, image #0, seed 42). Mechanism rows run 100 steps. The reference rows below the second rule are full Hessian methods granted twice the CubicKrylov oracle budget, 35.3k gevals. Rows exceeding it do so because a single dense $15\text{k} \times 15\text{k}$ Hessian build, 15.2k gevals, is atomic. The three mechanism rows correspond to the design decisions (a)–(c) and the fresh-gradient rule of Section 4.

Large block step	best PSNR (dB)	gevals	wall (s)	notes
Damped Newton (DSO-ablation fallback)	20.73	26.8k	11	reference
CubicKrylov, fixed $M{=}6L_H$, stale grad	25.91	16.1k	14	non-monotone, no phase-B safeguard
+ adaptive M, *joint* accept-or-reject	25.12	28.7k	16	319 rejections; M drifts to 596
+ two-phase accept-or-reject (B only)	23.13	31.9k	16	stale gradient, 875 rejections
+ fresh gradient in phase B	**31.66**	17.7k	15	104 rejections; every M_b at floor
Reg. Newton lazy, $m{=}n$ (Doikov et al., 2023)	31.69	30.2k	457	1 Hessian build, 15k steps
CN-Lazy, $m{=}n$ (Doikov et al., 2023)	31.39	30.2k	313	1 Hessian build, 15k steps
SSCN $\tau{=}64$ (Zhao et al., 2025)	23.18	35.4k	130	544 subspace steps
FD-CNM, $m{=}50$ (Doikov and Grapiglia, 2023)	16.72	45.8k	149	3 FD Hessian builds, 52 steps
DSO-BlockHess (per-layer full Hessian)	8.16	45.7k	9	budget consumed by rebuilds
Adaptive cubic Newton, $m{=}1$ (Doikov et al., 2023)	7.67	45.7k	178	3 builds, 6 steps
Cubic Newton (original, fresh Hessian) (Nesterov and Polyak, 2006)	3.42	45.7k	181	3 builds, 3 steps

The final configuration gains $+10.9$ dB over the damped-Newton fallback at $0.66\times$ the gradient equivalent cost on identical routing. The three mechanism rows isolate the contribution of each design decision. Without fresh phase-B gradients every variant plateaus 6–8.5 dB below the final configuration, whether the step is unsafeguarded (fixed M), vetoed jointly (319 rejections and M driven upward to 596), or vetoed per block against a stale model (875 rejections).

The reference block reports the full Hessian second-order methods evaluated elsewhere in the paper on the one benchmark where the dense $15\text{k} \times 15\text{k}$ Hessian is still feasible. Only the two lazy $m{=}n$ variants are competitive. Gradient regularized Newton (31.69 dB) and lazy cubic Newton (31.39 dB) bracket the CubicKrylov result (31.66 dB), but each needs $1.7\times$ its oracle budget and 21–30$\times$ its wall clock, and both are eliminated outright at 91.4M parameters where the dense build no longer fits. Methods that rebuild the Hessian frequently are excluded by the build cost alone. The original cubic Newton with a fresh Hessian every step (Nesterov and Polyak, 2006) exhausts the doubled budget after three steps (3.42 dB), and its adaptive-M lazy variant with $m{=}1$ after six. SSCN sidesteps dense builds via $\tau{=}64$ subspaces yet plateaus 8.5 dB below CubicKrylov, and DSO-BlockHess, which refreshes per-layer Hessians every 10 steps, spends the budget on rebuilds within 21 steps.

To rule out a single image artifact, the nano configurations were rerun on the first 20 ESM images with three random seeds each (Appendix V). The single image ranking is stable. CubicKrylov reaches 32.8 ± 1.4 dB, ahead of gradient regularized Newton on 17 of 20 images and of lazy cubic Newton on 19 of 20, at 59% of their oracle budget and $\sim 14\times$ less wall clock (Table 28, Figure 5).

I Extended Experimental Details

I.1 Setup, architectures, and metrics

Architectures. The full-scale single image model is ViSIR (91,383,042 parameters). A 60×60 RGB image is partitioned into $6 \times 6 = 36$ non-overlapping 10×10 patches by a strided convolution into an embedding of width $d = 256$, to which a learnable positional embedding is added. Two transformer encoder layers follow, each combining 8-head self-attention with a *SIREN* feed-forward block ($256 \to 512 \to 512 \to 256$, sinusoidal activations, $\omega_0 = 10$, outermost-linear) wrapped by two LayerNorms and dropout 0.1. The 36×256 encoder output is flattened and mapped by a linear layer to a two-dimensional latent, which a SIREN decoder ($2 \to 512 \to 512 \to 3 \cdot 240^2$, $\omega_0 = 10$, outermost-linear) expands directly into the $240 \times 240 \times 3$ high-resolution image. The final SIREN layer ($512 \times 172{,}800$) accounts for the bulk of the parameters. The resulting block structure, many tiny bias and LayerNorm blocks alongside a few very large weight tensors, is precisely the structure that the per-block algorithm of Section 4 targets. The *ViSIR-Nano* model of Appendix H is the same vision transformer and SIREN template scaled down (narrower embedding and SIREN decoder) to 15,235 parameters for single image $3\times$ super-resolution, small enough that the full $15{,}235 \times 15{,}235$ Hessian can be formed for the exact-cubic-Newton reference row of Table 2.

Metrics. We report four metrics. *PSNR* is the peak signal-to-noise ratio in dB, higher is better, computed as $\mathrm{PSNR} = 10\log_{10}\big(\mathrm{MAX}^2/\mathrm{MSE}\big)$ for peak signal value MAX. The training *MSE* is the mean squared error, the loss itself. *LPIPS* is the learned perceptual image patch similarity of Zhang et al. (2018), a perceptual distance between prediction and ground truth, lower is better. The *H/L ratio* is the ratio of high to low frequency band MSE under a radial FFT decomposition, and values near 1 indicate uniform per-band convergence, that is, little spectral bias. Per-band MSE is reported over the DC, Low, Mid-Low, Mid-High, and High radial frequency bands. Computational cost is measured in *HVPs* (Hessian–vector products), *gradient equivalents* ("gevals", one geval = the cost of a single gradient or backward pass), and wall clock seconds.

Iso-time comparison. Because second-order steps cost more per iteration than first-order ones, a fixed-step comparison favours the more expensive methods. We therefore complement the fixed 100-step results with an *iso-time* experiment. Letting T be the wall clock time of the slowest method's 100-step run, every optimizer is granted the same budget T (equivalently $\lfloor T/\bar{t}_i \rfloor$ steps for a per-step cost $\bar{t}_i$), so that cheaper first-order methods receive proportionally more steps. We report both experiments together with the measured per-method wall clock times. The convergence matched comparison on further INR architectures (SIREN, FINER, and WIRE) and on signed distance tasks appears in Appendix O.

I.2 ViSIR variants and scale-dependent failure modes

Chebyshev-ON and Chebyshev-OFF optimizer variants. The two reference rows (Table 3), an implementation of the DSO step following Appendix C and of its damped Newton counterpart, isolate the *step rule* on identical curvature. Both build the same exact per-block Hessian $\mathbf{H}_b$ on small tensors ($n_b \le 512$) and use the same Hutchinson-diagonal damped-Newton step on the large tensors. They differ only in how the small block step is formed. **Chebyshev-ON** (the DSO step) applies the second kind relaxation polynomial R_L to the normalized block Hessian via the three term recurrence Eq. (22), a degree-L matrix polynomial in $\mathbf{H}_b$ applied to the gradient, using only matrix–vector products and *no* linear solve. **Chebyshev-OFF** replaces that polynomial with a damped Newton step $-(\mathbf{H}_b + \delta\mathbf{I})^{-1} g_b$ obtained from a direct dense solve. The ON−OFF difference therefore measures the effect of the Chebyshev polynomial alone, with curvature, damping, line search, and large block handling held fixed.

Two scale-dependent failure modes surfaced when moving from Nano (∼8 large tensors) to the full model (45 large tensors), both predicted by the theory of Section 3.4:

1. **Joint acceptance stall** (Remark B.5). With a single accept-or-reject over all large blocks, the summed step jointly overshoots every trust region. The phase was vetoed every step and the loss froze at 0.40 for 19 steps. The correction is to accept or reject each block separately, with its own M_b (Proposition A.10).
2. **M-floor over regularization.** With the floor $M_{\min} = 6L_H = 60$, blocks whose local Hessian-Lipschitz constant is far below L_H took accepted but microscopic steps ($r \propto M^{-1/2}$, Proposition A.3). The loss crept at $\sim 10^{-3}$ per step. The correction is the nearly-zero floor $M_{\min} = 10^{-6}$ combined with the adaptation of M_b in both directions (design decision (c)).

With both corrections, training loss fell from 0.40 to 2.4×10^{-3} by step 16 and $\sim 10^{-5}$ by step 20.

I.3 Spectral bias mitigation on the full ViSIR

Spectral bias mitigation. The per-frequency band diagnostics of the ablation experiment show all five spatial bands (DC, Low, Mid-Low, Mid-High, High) reaching their convergence thresholds within *one step* of each other (DC at step 14, the remaining four at step 15, spread = 1), with final non-DC band MSEs within 2.3% of one another (0.1742–0.1782, Fourier domain units of the experimental setup). This is the per-band uniformity that the Chebyshev second kind equalization analysis predicts (Section 1), now achieved with the polynomial degree fixed at $L = 10$ rather than $L \propto \sqrt{\kappa}$, an empirical confirmation of the self-adaptive degree bound of Proposition A.3.

The stabilized configuration. The 100-step ablation above runs without the two per-block stabilizations of Section 4. With the per-block M_b floor and the noise-tolerant monotone guard active, the

Table 3: Full ViSIR single image ablation (91,383,042 parameters, ESM image #1, 100 steps). †Wall clock is the 100-step time on a single idle RTX 6000 (Blackwell). The Chebyshev-ON and Chebyshev-OFF rows are an implementation of the DSO step following Appendix C and of its damped Newton counterpart under the same experimental setup. LPIPS for Adam, Muon+AdamW, and SOAP and the final loss, LPIPS, and wall clock entries of the damped Newton large block row were measured in a second run under the identical setup (image #1, seed 42, 100 steps, idle GPU). The PSNRs of the two runs agree (Adam and SOAP exactly, Muon+AdamW 6.43 against 5.73 dB and damped Newton large blocks 21.98 against 22.25 dB, within run-to-run variation).

Method	PSNR (dB)	final loss	LPIPS	wall (s)†
CubicKrylov, degree L=10 in Krylov subspace on large blocks	**51.65**	$\mathbf{6.7 \times 10^{-6}}$	**0.000**	**148**
Cheby-ON n=5 (Chebyshev-2)	41.75	2.0×10^{-4}	0.001	80
Cheby-OFF n=5 (damped Newton)	22.81	8.3×10^{-3}	0.322	67
Block-CN-Lazy, damped Newton on large blocks	22.25	8.9×10^{-3}	0.426	76
SOAP	20.33	1.3×10^{-2}	0.374	5.7
Adam	15.65	0.142	1.321	1.3
Muon+AdamW	5.73	2.7×10^{-1}	0.867	2.3

blockwise optimizer reaches **64.4 dB** best PSNR over 150 block sweeps with the CubicKrylov subsolver (stored-basis Lanczos) (∼11 GB solver memory on the 88.5M parameter block) and **64.2 dB** with the Chebyshev three term recurrence at 7.0–8.7 GB total peak (Appendix N.1). A global single subspace ARC on the same experimental design stalls at 4.0 dB.

I.4 Fingerprint metric definitions and a parallel diagnosis

Fingerprint. At log-spaced checkpoints of an Adam run we measure five quantities matrix-free, by Lanczos on the exact Hessian–vector oracle. The first is the raw extreme-eigenvalue conditioning $\kappa_{\text{raw}} = \lambda_{\max}/|\lambda_{\min}|$. The second is the *Adam-preconditioned* conditioning κ_{Adam}, the same quantity for $D^{-1/2} H D^{-1/2}$ with the run's own second-moment state D, which is the conditioning Adam operates on, reported together with the reduction factor $\kappa_{\text{raw}}/\kappa_{\text{Adam}}$. The third is the diagonal mass $\rho = \|\operatorname{diag} H\|^2/\|H\|_F^2$, estimated without bias from Hutchinson probes, which measures how much curvature *any* diagonal can represent. The fourth is the negative spectral mass, the share of the spectral density below $-10^{-3}\lambda_{\max}$ by stochastic Lanczos quadrature (Ghorbani et al., 2019), a saddle indicator (Dauphin et al., 2014). The fifth is the gradient energy split over curvature classes from a Lanczos run seeded at $v_0 = g/\|g\|$ (Gur-Ari et al., 2018). With Ritz pairs (ν_i, w_i), $w_i = (e_1^\top y_i)^2$,

$$\text{flat_frac} = \sum_{i:\,|\nu_i| \le 10^{-3}\lambda_{\max}} w_i = \frac{\|P_{\text{flat}}\, g\|^2}{\|g\|^2}$$

is the share of the gradient's energy in near-zero curvature directions, and the high curvature and negative classes are $\nu_i > 0.1\,\lambda_{\max}$ and $\nu_i < -10^{-3}\lambda_{\max}$. The near-zero band is defined relative to the largest eigenvalue ($|\nu_i| \le 10^{-3}\lambda_{\max}$) because that is where the bulk of an overparametrized Hessian spectrum concentrates, a bulk of eigenvalues at zero plus a handful of data dependent outliers (Sagun et al., 2018). Gradient energy in this flat band produces little change in the loss, so only the components along the outlier and negative curvature directions can drive optimization progress. When flat_frac remains large throughout training, the gradient's energy stays trapped in near zero curvature directions, and that persistence is the mechanism behind the stall.

A parallel landscape diagnosis. The fingerprint above is measured along Adam's own trajectory with the method's raw second moment. A separate study uses this metric, among others, to provide an explanation for Adam (and its variants) training stalls on landscapes such as FINER's. Three of its findings bear directly on the dispatch rule above and on the φ_1 results of Section 6.3. First, the raw diagonal-mass ratio ρ is biased by Adam's own preconditioning and by finite-sample curvature noise. That study derives a debiased estimator that removes this bias and shows the correction matters most exactly on the coupled and saddle dominated cases of Table 4. Second, a 2×2 coupling model separates the Hessian's action into an axis-aligned block (H_1, where a diagonal preconditioner is asymptotically sufficient) and a genuinely coupled block (H_2, where there is reduction in effective Hessian condition for Adam), giving a quantitative measurement of the axis-aligned vs. cross-coupled ill-conditioning degree. Third, two further first-order and quasi-Newton references confirm that the stall is not an artifact of Adam's specific preconditioner. On ESM image 0, L-BFGS (three memory

Table 4: Landscape fingerprints along the Adam trajectory (late trajectory values, initialization in parentheses). † marks an indefinite spectrum, where ($\lambda_{\max} \approx -\lambda_{\min}$) makes the raw ratio uninformative. κ_{Adam} is the residual conditioning. Outcomes compare plateau-converged tuned first-order methods (Adam, SOAP) with the blockwise cubic methods. INR rows use 240×240 image regression and Thai-statue SDF fitting with 3×256 coordinate MLPs. The FINER 2D row is ESM image 0. Over all 20 ESM images the step-10^5 values span $\kappa_{\mathrm{raw}} = 1.5 \times 10^4$–$5.1 \times 10^5$, $\kappa_{\mathrm{Adam}} = 4.6$–$2.6 \times 10^3$ (median 1.2×10^2), $\rho = 0.95$–0.99, flat fraction ≤ 0.001, and the initial negative mass spans 0.57–0.89 (median 0.81).

Task	κ_{raw}	κ_{Adam}	red.	ρ	neg. mass	flat frac	outcome
35k spectral	4.9e2	3.1e1	16×	.04	.92	.000	Adam at precision floor
SIREN 2D	1.6e3	9.9e1	16×	.09	.09	.000	parity
WIRE 2D	1.2e5	8.4e2	143×	.65	.00	.000	parity
FINER 2D	5.4e4	6.1e1	887×	.98	.19 (.81 init)	.000	ARC +59.5 dB
SDF-SIREN	8.9e3	2.4e1	372×	.06	.67	.000	parity; CubicKrylov best test
SDF-FINER	1.0e3	6.9	146×	.58	.68	.001	Adam chamfer 4.6–5.5× worse
multi-saddle	2.0e6	1.5e6	1.3×	.02	.46	.005	second-order better
Chebyshev-Rosenbrock	†	5.5e5	—	.04	.49	.001	second-order better

sizes) plateaus at a constant-like 7.3 dB by iteration 500 and never changes again, collapsing at the initialization saddle with no bounded escape mechanism for negative curvature. SOAP, tuned by inheriting Adam's own best learning rate, is the strongest short-horizon first-order optimzier but it stalls near Adam's own ceiling at its best logged checkpoint (77.65 dB), after more total wall clock than Adam needs to reach that ceiling itself. Both observations are consistent with the fingerprint's diagnosis that the limiting factor at this initialization is saddle structure, not curvature conditioning that a better-preconditioned first-order method could mitigate.

I.5 FINER SWEEP DETAILS AND PER-BAND ANALYSIS

Sweep. A six-configuration successive-halving sweep ($c_h \in \{1, 3, 10\} \times \sigma_0 \in \{0.3, 1.0\}$, $L = 25$, 500-step confirmation on two images) ranks c_h=1.0, σ_0=0.3 first at $\mathbf{64.79}$ dB mean best PSNR, within 0.4 dB of converged, tuned Adam, at only 500 steps. The winner lies on the sweep boundary on both axes, triggering the extended sweep (geometric edge extension, a Lanczos-degree axis, and the free horizon ranking of Appendix E.2). The extension identifies the Lanczos depth as the dominant untuned axis. The edge rule raises L from the grid's 25 through geometric extensions to a confirmed winner at $L = 163$ ($c_h = 1.0$, $\sigma_0 = 0.3$) with $\mathbf{90.65}$ dB mean best PSNR at 500 steps, +25.9 dB over the $L = 25$ configuration of the grid above, with the response flattening beyond $L \approx 102$ (89.4/90.3/90.7 at $L = 102/129/163$) while per-cell wall clock grows from 393 s to 3.3 ks. The in-subspace horizon ranking did not reach the confirmation rung. The depth axis dominated. A cubic step at the same Lanczos depth separates the deep-basis effect from the step rule. At L=163 and the same step budget, the cubic step rule reaches 72.1 dB where the φ_1 rule reaches 90.65 dB, so the margin comes from the step rule rather than from subspace depth.

Bounded grid. At the tuned 500-step configuration, the recorded experimental design (20 images × 3 seeds, 500 steps) gives $\mathbf{63.53 \pm 3.83}$ dB mean best PSNR over 60 cells, the strongest 500-step experimental setup recorded on this benchmark, +1.75 dB over ARC-block with the cubic step (61.78 ± 4.22), +9.2 dB over the tuned Adam experimental setup (54.34 ± 1.48), +16.3 dB over CubicKrylov (47.21 ± 9.04), and +27.6 dB over DSO-v2r (35.95 ± 2.18), with per-cell wall clock ∼400 s.

Spectral-bias mitigation at 500 steps. Gradient-type training drains the error component along each Hessian (equivalently, tangent kernel) eigenmode at a rate proportional to its eigenvalue (Jacot et al., 2018), the mechanism behind band-ordered convergence in coordinate networks. A band uniform residual is therefore the observable signature of a step rule that equalizes the per-mode rates. Over the full grid (60 cells, identical images and seeds per row), the per-band residual power decomposition reads:

ARC-φ_1's high-band residual power is ∼3× below the tuned Adam experimental setup's (2.18×10^{-2} vs 6.48×10^{-2}) while its low-band residual is lower by the same factor. Every band improves together, so the mitigation is not a redistribution of error across frequencies. It also improves on the cubic step reference by ∼1.6× per band, consistent with its +1.75 dB mean PSNR margin on the same cells. The remaining residual is band uniform to within 10% across Low–High ($H/L \approx 1.10$), so

Table 5: FINER, 500 steps, mean residual FFT power per band over the 60 grid cells. H/L = median high/low band ratio. Values near 1 indicate a band uniform residual.

optimizer	Low	Mid-Low	Mid-High	High	H/L
ARC-φ_1	$\mathbf{2.41\times10^{-2}}$	$\mathbf{2.21\times10^{-2}}$	$\mathbf{2.19\times10^{-2}}$	$\mathbf{2.18\times10^{-2}}$	1.10
ARC-block (cubic)	3.94×10^{-2}	3.57×10^{-2}	3.52×10^{-2}	3.49×10^{-2}	1.07
Adam (source paper experimental setup)	6.61×10^{-2}	6.46×10^{-2}	6.46×10^{-2}	6.48×10^{-2}	0.81
Adam (untuned)	2.1×10^{1}	1.8×10^{1}	1.8×10^{1}	1.7×10^{1}	0.37

no frequency preference remains at this accuracy level. This is the result at 500 steps. Run to full convergence (below), the margin between ARC-φ_1 and Adam widens by more than 50 dB while the reconstructions remain visually indistinguishable at native resolution, which is the point the convergence figures make directly.

Optimizer	421 s full-budget run	4,207 s full-budget run
Adam (tuned)	67.7 dB	78.2 dB
SOAP (Adam's lr)	$\approx$ 72.9 dB	$\approx$ 77.3 dB
L-BFGS (m=10, 20, 50)	7.3 dB	7.3 dB
ARC-φ_1 ($L_{\max}$=163)	58.3 dB	**95.6** dB
AdaCubic (authors' settings)	41.3 dB	46.1 dB

Table 6: Iso-time comparison at Adam's two full-budget runs, FINER ESM image 0. SOAP values are linearly interpolated between its logged checkpoints. AdaCubic values are the best PSNR reached within each wall clock allowance on its own 2 h run. All others are direct readings.

AdaCubic on the same images. AdaCubic (Tsingalis et al., 2026), the cubic regularized optimizer with a Hutchinson diagonal curvature model of Section 1, was run from its authors' implementation with their published settings (η_1=0.05, η_2=0.75, one Hutchinson probe per step) under a 2 h time limit on ESM images 0 and 1, same seed and hardware as above. On image 0 it reaches 28.5 dB at 122 s, 40.8 at 417 s, 45.4 at 800 s, 45.9 at 3,600 s, and its best value of 46.4 dB at 6,328 s, after which the trajectory oscillates without improving. On image 1 the best is 46.7 dB. It spends 8.5×10^5 gradient equivalents in the 2 h (425,673 steps at one gradient and one Hessian–vector product each, and its 425,673 forward-only ratio-test evaluations would add a further 4.3×10^5 under the gradient equivalent count of Appendix U.2), a fraction of the 2.4×10^6 that ARC-φ_1 at L=25 spends in the same 2 h to reach 125.5 dB (image 0) and 117.2 dB (image 1), and it is 26–32 dB below tuned Adam at equal wall clock. Two measurements locate the difference. Its residual stays low frequency dominated, $H/L = 0.11$ on both images against 1.06–1.38 for ARC-φ_1 and 0.28–0.93 for Adam, so the diagonal cubic step does not remove the spectral bias, and its acceptance rule kept 38.8% of the steps on which the loss rose (the "negative ratio" branch of its trust update), reverting only 8.9%, which is the oscillation seen after 6,328 s. The diagonal surrogate leaves out the cross parameter curvature that the per-block step retains, and on this landscape that curvature is where the accuracy is.

20 ESM images. The full convergence comparison of Section 6.3 is one image. The same three configurations were run on ESM images 0–19 at seed 0. ARC-φ_1 ($L_{\max} = 163$, $c_h = 1$, $\sigma_0 = 0.3$) ran for 2,000 sweeps under an 18,000 s cap, tuned Adam (learning rate 10^{-4}) for 10^6 steps, and AdaCubic with its authors' settings for 2 h, one run per cell on the same GPU model (Table 7). At 2,000 sweeps ARC-φ_1 reaches a median best PSNR of 131.7 dB (mean 120.3 $\pm$ 18.9) against 71.8 dB (72.8 $\pm$ 4.5) for Adam and 46.7 dB (46.7 $\pm$ 0.3) for AdaCubic, and it is ahead on every image. The paired gain over Adam is 47.4 $\pm$ 20.6 dB (Wilcoxon $p = 1.9 \times 10^{-6}$, the smallest value 20 paired samples admit) and over AdaCubic 73.6 $\pm$ 18.9 dB. The spread is on the ARC-φ_1 side, and it is an artifact of the cutoff. Twelve images end between 131.1 and 135.6 dB. On the other eight the best sweep is the last one or within two of it and the trajectory is still gaining 1.5–4.2 dB per 200 sweeps, namely image 12 (128.9 dB), images 2, 17, 4, and 10 (101.9–116.1 dB), and images 16, 8, and 9 (78.6–87.7 dB). Image 9, the slowest, is 1.0 dB above Adam's best at that point.

Continuation to a plateau. The eleven images whose best value was still moving at sweep 2,000, the eight above and images 0, 11, and 19, whose last 200 sweeps gained 1.3–3.6 dB, were rerun from initialization with the same configuration under a plateau rule that stops once the best PSNR has gained less than 0.05 dB over the preceding 400 sweeps, after at least 4 h (Table 8). Eight plateaued under a 12 h cap, after 4.1–9.1 h, between 128.8 and 133.5 dB, inside or within 2.3 dB of the 131.1–135.6 dB band the other twelve images had reached at 2,000 sweeps. The three that started lowest, images 8, 9, and 16, were still gaining 1.7–2.0 dB per 400 sweeps when that cap stopped

them at 117.2, 107.6, and 123.5 dB. Rerun under a 48 h cap, they plateaued on their own after 20.0, 20.8, and 27.5 h, at 128.5, 119.7, and 130.3 dB, that is, image 16 inside the band, image 8 0.3 dB below its lower edge, and image 9, the slowest at every stage, 9 dB below it. Images 0, 11, and 19, already above 131 dB at the cutoff, gained 0.7–1.3 dB and plateaued within an hour of the 4 h floor. With the plateau values in place of the 2,000-sweep ones, the per-image best of ARC-φ_1 has median 132.4 dB (mean 131.5 ± 3.3, minimum 119.7), sixteen of the 20 images are at or above 130 dB and 19 at or above 128 dB, and the paired gain over Adam is 58.7 ± 5.8 dB, from 42.1 dB (image 9) to 66.4 dB (image 10). On image 0 the continuation stops at 133.5 dB, the value of the Section 6.3 run, with its best at sweep 2,425 and no gain over the following 400, so that figure is a plateau and not a snapshot of a climb. The continuations cost 154 GPU hours, 86 for the eleven runs under the 12 h cap and 68 for the three reruns under the 48 h cap.

Adam's behavior is the same on every image. Its best value is reached between step 616k and 999k (2,444–5,064 s) and the final value is 3–19 dB below it. AdaCubic's best values lie in a 1 dB band, 46.2–47.2 dB, reached between step 232k and 478k, after which the trajectory oscillates and ends 3–25 dB lower. Neither has anything to gain from more wall clock, which is why only the φ_1 arm was continued. The coupling probes of Appendix S, taken at the last checkpoint of each 2,000-sweep run (images 1–19, since the image-0 cell predates the probes), give $\beta = 0.61$–0.95, $\delta_{\mathrm{spec}} = 4.1 \times 10^2$–$1.6 \times 10^3$, and $\varepsilon_\star = 2B\delta_{\mathrm{spec}}^2/C_1 = 7.7 \times 10^3$–$3.8 \times 10^5$ from Theorem B.10, while the recorded $\|\nabla f\|_{\min}$ lies between 7.7×10^{-8} and 1.6×10^{-5}. As on image 0, the floor limits the counting argument of Theorem B.10. The landscape fingerprint of Section 6.2, measured along tuned Adam on all 20 images, gives an initial negative spectral mass of 0.57–0.89 (median 0.81, the image-0 value) and, at step 10^5, $\kappa_{\mathrm{Adam}} = 4.6$–$2.6 \times 10^3$ (median 1.2×10^2), $\rho \geq 0.95$, and a flat gradient fraction of at most 0.001, so every image is in the saddle dominated case *(iii)* of Table 4. The four images with the highest initial negative mass (0.87–0.89, images 8, 4, 16, and 10) are four of the eight slow ones, and the rank correlation between initial negative mass and PSNR at 2,000 sweeps is -0.42. At 20 images this is an association. The image-0 comparison of Figure 1 stays the representative figure, and the tables give the spread around it.

Table 7: 20 ESM images, seed 0. Best PSNR (dB) with the sweep or step and the wall time (s) at which it was reached. "final" is the value at the end of the run. ARC-φ_1 ran with $L_{\max} = 163$ for 2,000 sweeps, Adam for 10^6 steps, and AdaCubic for 2 h. Medians are over the 20 images. $*$ marks image 19, whose Adam run shared its GPU with an unrelated job for its whole duration (194 against $\sim$256 steps/s on the other images). Its step count and PSNR are unaffected, but its wall time is not comparable.

	ARC-φ_1			Adam				AdaCubic		
image	best	sweep	s	best	step	s	final	best	step	final
0	132.64	1994	10582	78.16	615983	2444	63.18	46.36	374124	38.11
1	135.61	1980	10327	77.19	975354	3890	67.61	46.69	417803	42.64
2	116.08	2000	11002	70.73	985679	3926	63.87	46.93	377292	35.68
3	132.58	1821	10321	71.47	998731	3986	64.61	46.65	466877	39.27
4	108.73	2000	10934	67.47	943498	3762	57.26	46.34	404278	20.94
5	132.37	1803	9850	66.42	972526	3874	57.05	46.31	235447	40.24
6	131.92	1937	10455	70.44	974421	3885	61.87	47.06	438311	40.63
7	132.87	1981	10529	73.42	668047	2659	57.41	47.23	385325	34.67
8	84.34	2000	11365	78.40	874444	3477	68.16	46.62	280061	35.21
9	78.65	2000	10746	77.60	876819	3506	60.47	46.99	466616	31.85
10	101.86	2000	10436	63.44	922487	3595	60.63	46.20	317237	42.57
11	131.11	2000	10809	68.64	977808	3823	57.85	46.59	232185	43.54
12	128.92	1998	10600	75.94	992083	3892	65.10	46.90	466820	42.24
13	135.20	1941	10097	72.11	983105	3844	62.69	46.58	305321	35.37
14	132.72	1940	10482	71.52	964129	3774	56.64	47.06	412915	37.08
15	132.89	1860	9821	75.53	993842	3880	62.33	46.38	333601	32.33
16	87.70	2000	11672	77.06	762730	2976	61.85	47.12	478491	42.61
17	105.23	2000	10431	79.73	935191	3659	60.78	46.83	353494	29.57
18	131.54	1984	10416	70.33	967572	3791	62.36	46.86	420730	30.18
19	132.50	1981	11076	71.27	984822	5064*	57.29	46.46	364598	36.25
median	131.73	1989	10505	71.82	970049	3807	61.86	46.67	381308	36.67

Table 8: Continuation of the eleven ARC-φ_1 runs whose best value was still moving at 2,000 sweeps, same configuration from initialization, stopped by the plateau rule (less than 0.05 dB gained over 400 sweeps, after at least 4 h). "at 2k" is the 2,000-sweep value of Table 7. "sweep" and "s" locate the continued best. "h" is the run's wall time. "cap" is the wall clock limit under which the run plateaued. "gain" is the paired margin over Adam's best on the same image. Images 8, 9, and 16 first ran under the 12 h cap, which stopped them, still rising, at 117.2, 107.6, and 123.5 dB. Their rows give the reruns under the 48 h cap.

image	at 2k	continued	sweep	s	h	cap	Adam	gain
0	132.64	133.49	2425	12703	4.10	12 h	78.16	55.3
2	116.08	131.80	4001	26042	7.93	12 h	70.73	61.1
4	108.73	128.83	4048	23556	6.72	12 h	67.47	61.4
8	84.34	128.50	12007	71662	19.97	48 h	78.40	50.1
9	78.65	119.69	13398	74650	20.76	48 h	77.60	42.1
10	101.86	129.81	5050	27433	8.21	12 h	63.44	66.4
11	131.11	132.40	2516	13482	4.34	12 h	68.64	63.8
12	128.92	133.10	3143	16598	5.19	12 h	75.94	57.2
16	87.70	130.29	10733	96100	27.51	48 h	77.06	53.2
17	105.23	131.59	5788	30519	9.07	12 h	79.73	51.9
19	132.50	133.20	2370	13282	4.28	12 h	71.27	61.9

Table 9: Adam on the 35k SIREN-like benchmark hyperparameter sweep

lr	0.05	0.03	0.01	3×10^{-3}	10^{-3}	3×10^{-4}	10^{-4}
best total MSE	2.1×10^{-3}	0.587	0.591	5.3×10^{-6}	5.3×10^{-7}	3.0×10^{-13}	6.3×10^{-14}
steps	5000 ep.	60k	60k	60k	60k	60k	76k
wall (s)	—	163	168	170	169	164	212

J TUNED ADAM ON THE 35K SIREN-LIKE BENCHMARK

This appendix reports the Adam tuning study on the 35k parameter SIREN-like benchmark that supports case *(i)* of the fingerprint taxonomy of Section 6.2.

Tuned Adam on the 35k SIREN-like benchmark. The 35k parameter SIREN-like regression benchmark composes six octave-spaced frequency bands $\omega = 1, \ldots, 32$ with harmonic amplitude decay $a_k = 1/k$ over a polynomial double well and a curved valley (maximum curvature ratio 1229). The per-band MSE metric requires every band to converge relative to its own scale. A coarse learning rate grid (best point 0.05 at 5,000 epochs) gives Adam 2.1×10^{-3}, the loss that makes second-order methods look much better. Extending the sweep toward the small learning rates and running to a plateau criterion significantly improves Adam's achieved loss (Table 9). At $\mathrm{lr} = 10^{-4}$ Adam reaches 6.3×10^{-14} total MSE in 76k steps (212 s), the fp32 precision floor, and converges all six bands. Refinements of $\beta_2 \in \{0.99, 0.999, 0.9999\}$ and $\epsilon \in \{10^{-8}, 10^{-10}\}$ at that learning rate do not improve further. The improvement is not obvious at first since the loss change dynamics with decreasing learning rate is not monotonic. The loss goes up at lr = $\sim$0.03. Adam stalls at $\sim$0.59 for $\mathrm{lr} \in [0.01,\ 0.03]$. Yet, it reaches machine precision at $\mathrm{lr} \leq 3 \times 10^{-4}$, so a coarse grid misses the region entirely. On this landscape a thoroughly tuned Adam is sufficient, and the trajectory fingerprint below explains some of the reasons for this behavior.

K SMALL SCALE HARD LANDSCAPE BENCHMARKS

Before the FINER-scale comparison, four small benchmarks exercise the three curvature cases of Proposition 5.1 in isolation, namely MLP fits of the coupled Rosenbrock–Ackley and Chebyshev–Rosenbrock surfaces (ill-conditioned, coupled valleys with $\rho \approx 0$ in the fingerprint taxonomy above), a multi-saddle surface (negative mode escape, since the persistence of small negative eigenvalues late in training is a documented property of overparametrized landscapes (Sagun et al., 2018), which is what the φ_1 escape term of Section 5.2 addresses deterministically), and the 35k parameter multi-frequency SIREN benchmark already used for the spectral-bias correction of Section 6.2 (terminal accuracy rather than convergence speed). CPU runs, matched sweep budgets, identical experimental designs per row, comparing the φ_1 step of this section against the CubicKrylov step of Section 2.2 at

equal Lanczos depth, so every delta below isolates the step rule, independent of the granularity-and-stabilization advantage already established for the cubic step over first-order comparisons.

Table 10: Small scale benchmarks, terminal loss after the stated sweeps (CPU, matched budgets, gradient equivalents in parentheses). ARC-φ_1 uses σ_0=1, c_h=3, L=15. The reference is ARC-block (CubicKrylov step) with σ_0=1, $L_{\max}$=15.

benchmark (sweeps)	initial loss	ARC-φ_1	ARC-block (cubic)
Rosenbrock–Ackley (300)	1.98×10^{4}	$\mathbf{3.4\times10^{-2}}$ (69.4k)	1.37 (21.4k)
Chebyshev–Rosenbrock (300)	1.30×10^{4}	$\mathbf{7.8\times10^{-3}}$ (69.8k)	0.73 (21.4k)
multi-saddle (300)	7.51×10^{3}	$\mathbf{2.2\times10^{-3}}$ (67.0k)	2.5×10^{-2} (29.4k)
mf-SIREN 35k (40)	4.61	$\mathbf{4.1\times10^{-10}}$ (18.6k)	1.3×10^{-4} (9.6k)

Three observations. First, the φ_1 step reaches terminal losses 1–5 orders below the cubic step on every benchmark. The largest margin is on the 35k SIREN, where 40 sweeps take the loss to 4×10^{-10}, five decades past the cubic step. Second, the advantage costs oracle budget through the acceptance loop. The default configuration rejects more trials than the cubic step (1,176 vs 161 rejections on Rosenbrock–Ackley), spending $\sim3\times$ the gradient equivalents in the same sweep count. The sweep of Section 6.3 indicates the elevated rejection rate is a tuning artifact of c_h. Third, escape statistics show the escape mechanism in action. On the multi-saddle benchmark the negative mode branch (Section 5.2) performs 353 block trials, and on a rotated (coupled) saddle, the subspace step escapes through a direction that no diagonal method can exploit.

L Comparison with Other ARC-Based Approaches

This section contrasts the design choices of this work, per-tensor blocks with an independent cubic constant and acceptance test per block and a Chebyshev-bounded Krylov subsolver driven only by Hessian–vector products, with recent ARC-based methods. Table 11 collates the families closest to the claim of Appendix M by the curvature used in the step, the largest demonstrated scale, and the exactness of the cubic model step on the true curvature. Table 12 compares the corresponding inner subproblem solvers by computation and working memory.

Appendix Q reviews these families in detail, namely cubic model minimization in Krylov subspaces, blockwise and partition-based cubic methods, quasi-Newton cubic models, Chebyshev recurrence stability, parameter-free and inexact-Hessian ARC theory, stochastic and lazy cubic Newton, subspace and multilevel methods, Krylov-type inner solvers, second-order methods at deep learning scale, trust region hybrids, and cubic regularization beyond unconstrained training.

M Feasibility of Second-Order Steps at the 91.4M-Parameter Scale

In this section we assess the feasibility of the evaluated optimization methods at large scale. For a benchmark with millions of parameters, we used the 91.4M parameter ViSIR.

The cubic step is applied to every tensor, including the largest. The 91.4M parameter ViSIR model of Section 6 contains 45 parameter tensors. Under the block partition of Section 3.4 with maximal exact-block size 512, the 23 tensors with at most 512 entries (LayerNorm parameters, small biases, and similar vector-shaped parameters, 7,938 parameters in total) are optimized with the lazy exact-Hessian cubic step of Algorithm 1. The remaining 22 tensors (the patch embedding, positional embedding, attention projections, SIREN linear weights, and the decoder) are large blocks and receive the matrix-free CubicKrylov step. No tensor is excluded, and no tensor falls back to a first-order update.

The largest block is the outermost decoder weight, of shape $172{,}800 \times 512$, that is, 88,473,600 parameters or 96.8% of the model held in a single tensor. Its cubic step is computed from $L = 10$ Hessian–vector products, and the only additional storage is the Lanczos basis with its work vector, eleven vectors of the tensor's own size (≈3.9 GB in float32). Dense Hessian of this block alone would occupy $\sim3 \times 10^7$ GB, and a factorization-based cubic solve on it would require on the order of $n_b^3 \approx 7 \times 10^{23}$ arithmetic operations. Since the cost of $L+3$ gradient equivalents per step (Table 1) does not depend on the block size, this tensor has no higher *oracle* cost than a block of a few hundred

Table 11: Curvature model, demonstrated scale, and cubic model exactness of the second-order families reviewed in this section. Exactness is meant in the sense of Lemma A.4, that the step minimizes the cubic model over the true (block) Hessian, accessed through Hessian–vector products, within the constructed subspace.

Method	Curvature in the step	Demonstrated scale	Exact cubic model step on true curvature
AdaCubic (Tsingalis et al., 2026)	Hutchinson diagonal	deep learning training; FINER/ESM here, 46.4 dB at 2 h (Appendix I.5)	no: diagonal surrogate
Block-Kronecker Fisher (Martins Gomes, 2025)	per-layer Kronecker factors	deep learning training	no: factored surrogate
Full Gauss–Newton study (Abreu et al., 2026)	full or layerwise Gauss–Newton	150M parameter transformers	outside the cubic Newton family; supports per-tensor blocking
Quasi-Newton ARC (Foristal et al., 2022; Ranganath et al., 2025)	limited-memory low-rank	deep network training	exact solve, but on a surrogate Hessian
True-Hessian subspace ARC (Bellavia et al., 2025; Al Daas and Gould, 2025; Dussault et al., 2023; Cartis et al., 2025; Tansley and Cartis, 2025)	exact Hessian (subspace or shifted solves)	standard test sets, moderate dimension	not demonstrated at network scale
Stochastic, sketched, and multilevel cubic Newton (Pasechnyuk-Vilensky et al., 2025; Scheinberg and Xie, 2023; Chen et al., 2026; Higuchi et al., 2025; Tsipinakis et al., 2026)	sampled, sketched, or compressed	small to moderate	subspace or compression replaces the exact model
ARC-BLOCK (this work)	exact block Hessian via HVPs	91.4M parameters; single 88.5M tensor	exact on every block (Lemma A.4), under either rule that sets the subspace dimension

entries per step. The wall clock cost of each HVP and the Lanczos-basis memory still scale linearly with the tensor size.

The feasibility of the evaluated optimizers on ViSIR-Nano model. The dividing line is whether a method requires a dense representation or factorization of the (block) Hessian, or only Hessian–vector products, Hessian diagonals, or per-dimension factored statistics. The dense-solve cubic Newton family (the original method (Nesterov and Polyak, 2006), its lazy and adaptive variants (Doikov et al., 2023), the finite-difference implementation (Doikov and Grapiglia, 2023), and SSCN with exact subspace Hessians (Zhao et al., 2025)) solves a regularized system in a dense Hessian and is therefore confined to models near the 15k-parameter feasibility boundary studied in Appendix H (their per-step complexity classes are collated in Table 11). Among cubic regularized methods, only the matrix-free Krylov variants, adaptive cubic regularization with Lanczos subproblem solves (Cartis et al., 2011a; Gould et al., 1999; Carmon and Duchi, 2018; Dussault et al., 2023) and the CubicKrylov step of this work, remain feasible at 10^8 parameters, since their per-step cost is a fixed number of HVPs, independent of the tensor size. Within that family, the multi-shift ARC$_q$K solver of Dussault et al. (2023) makes the shift search itself matrix-free by solving a grid of shifted systems with a single Lanczos-CG process in the full space. The recurrence of this work instead reuses per-block Krylov bases across σ_b updates. The other second-order optimizers of the multi-image nano benchmark (Table 28) that remain feasible by never representing the Hessian beyond cheap surrogates. The Chebyshev ablations and AdaHessian use Hutchinson diagonal estimates from a few HVPs, SOAP maintains per-dimension factored moment matrices, and Muon (Jordan et al., 2024) orthogonalizes per-tensor updates. Rank-τ spectral preconditioning (Doikov et al., 2024) is likewise matrix-free in principle, at a cost of τ HVPs per preconditioner rebuild. The distinction matters for the following comparisons. At 91.4M ViSIR scale the CubicKrylov optimizer and the per-block ARC ablation introduced later in this section are, to our knowledge, **the only members of the cubic Newton family**

Table 12: Inner subproblem solvers of the methods in Table 11, by computation per subproblem solve and working memory beyond the iterate and gradient. In the notation, n is the problem (or block, n_b) dimension, L the number of Krylov steps, ℓ the sketch or subspace dimension, m the quasi-Newton memory. Computation excludes the gradient evaluation itself.

Inner solver	Methods	Computation per solve	Working memory
generalized Lanczos trust region (GLTR) tridiagonalization + secular solve	Cartis et al. (2011a); Gould et al. (1999); Carmon and Duchi (2018)	L HVPs + $\mathcal{O}(L)$ secular iteration	basis of L n-vectors, or a second Lanczos pass with $\mathcal{O}(1)$ vectors
Multi-shift Lanczos conjugate gradient (CG) over a shift grid	Dussault et al. (2023)	one Krylov process, L HVPs shared by all shifts	$\mathcal{O}(1)$ n-vectors per shift; no stored basis
Reused polynomial/rational Krylov subspace	Bellavia et al. (2025)	basis built once, reused across outer iterations; rational variant needs shifted direct solves	stored basis; factors for the rational variant
Extended-Krylov solution-family basis	Al Daas and Gould (2025)	one matrix factorization + basis extension; reduced solve serves every regularization weight	factorization + basis
Sketched reduced cubic model	Cartis et al. (2025); Tansley and Cartis (2025); Higuchi et al. (2025)	ℓ projection products + dense reduced solve $\mathcal{O}(\ell^3)$	$\ell \times n$ sketch or basis
CG / minimal-residual (MINRES) iteration on a shifted quadratic	Zhou et al. (2025); Zeng et al. (2026a;b)	one HVP per iteration, short recurrence; no cubic model	$\mathcal{O}(1)$ n-vectors
Closed form low-rank quasi-Newton cubic	Forristal et al. (2022); Ranganath et al. (2025)	$\mathcal{O}(m^2 n)$ dense algebra; no HVPs	$2m$ n-vectors of curvature pairs
Separable scalar cubic on a Hutchinson diagonal	Tsingalis et al. (2026)	a few HVPs for the diagonal estimate + $\mathcal{O}(n)$ per-coordinate closed form	$\mathcal{O}(n)$ diagonal
Kronecker-factor inversion	Martins Gomes (2025)	small per-layer factor inverses	two small factor matrices per layer
CubicKrylov step (this work, Section 3.2)	per block	L HVPs + tridiagonal secular solve on an $L \times L$ matrix + 1 HVP for the model decrease	Lanczos basis of L n_b-vectors plus one work vector (full reorthogonalization)
φ_1 step in the same subspace (this work, Section 5)	per block	$L{+}1$ HVPs + eigendecomposition of the $L \times L$ tridiagonal	same Lanczos basis
Shifted-system CG (this work, ARC-BLOCK-PAMM, Appendix R.1; step rule after Doikov, 2026)	per block	at most k_{cg} HVPs, $\mathcal{O}(n_b)$ per iteration, + one HVP for the model decrease; no cubic model	four n_b-vectors
Chebyshev three term recurrence (this work, DSO ablation, small blocks)	per block	explicit $\mathbf{H}_b$ build + degree-L recurrence, $\mathcal{O}(Ln_b^2)$	$\mathbf{H}_b$ (n_b^2) + $\mathcal{O}(1)$ vectors

in this evaluation whose steps remain exact in the cubic model sense on every block at this scale, rather than an approximation through a diagonal or factored surrogate.

Comparison to ARC The use of Krylov subspace by itself is not sufficient to make an optimizer feasible at 91.4M parameter scale.

We checked this by implementing ARC (Cartis et al., 2011a;b) in its global form, Algorithm 2.1 with a Lanczos subproblem solve over the whole 91.4M-dimensional parameter vector (degree bound $d \leq 15$, termination criterion TC.s, noise-regularized acceptance ratio), and ran it on the single image reconstruction experimental design of Section 6 (91.4M ViSIR). TC.s is the residual norm

test of Cartis et al. (2011a), under which the Lanczos iterations stop once the norm of the cubic model gradient at the current inner iterate falls below its threshold, so this reference sets its Krylov subspace dimension at run time from the residual, in contrast to the precomputed degree of Section 3.2. Published applications of ARC-type methods to neural networks, to our knowledge, use sub-sampled or stochastic Hessian estimates on models in the 10^4–10^5 parameter range (Kohler and Lucchi, 2017; Tripuraneni et al., 2018; Xu et al., 2019). We are not aware of existing experimentations on variants of ARC with exact Hessian–vector products, the Lanczos subproblem solver, and the unmodified acceptance test that were performed at the 10^7–10^8 parameter scale (as in this section).

The original ARC per-step cost is indeed modest, ≈ 0.3 s and ≈ 11 gradient equivalents per accepted step, yet the method stalls at 4.0 dB PSNR (ViSIR), while CubicKrylov reaches 51.7 dB. After $10{,}000$ steps ($2{,}858$ s, 1.1×10^5 gradient equivalents, (its oracle budget is greater than the budget consumed by CubicKrylov in its entire run). The reason for the stall of ARC optimizer can be deduced from the regularization trace. With a *single global* weight σ, every search step must satisfy the acceptance test along all 91.4M directions simultaneously, so σ is driven as high as 5×10^{11}, 46% of all trials are rejected, and the accepted steps are shrunk to noise level for the entire network. At the scale of 15k parameters the same implementation is in first place in this evaluation (Appendix X). The per-block decomposition is more than a cost optimization. Replacing one global σ by per-block constants M_b, each adapted to its own tensor's curvature scale, is what allows large steps in well-conditioned blocks while the ill-conditioned ones are regularized individually.

Isolating the mechanism with a block-σ ARC ablation. To determine if the failure is solely due to the σ being global, we ran an ablation, the identical ARC implementation with one regularization weight σ_b *per parameter tensor*, applied in Gauss–Seidel sweeps (per block, a Lanczos cubic solve on the block-restricted Hessian–vector operator, a trial update of that block only, and its own accept-or-reject ratio test with a fresh function evaluation). Nothing else changes, with the same η/γ constants, the same TC.s residual norm termination of the Lanczos iterations, same degree limit. The difference in performance between a small scale and large scale is striking. Where the global σ ARC variant stalls at 4.0 dB, the block variant reaches **61.9 dB** after 13 sweeps, in 19 s and 8.2×10^3 gradient equivalents, on the identical experimental design (by comparison, the Cubic Krylov reference on this image reaches 51.7 dB at 3.7×10^4 gradient equivalents). The computed weights confirm this result. After 150 sweeps the σ_b span more than twelve orders of magnitude across the 45 tensors (from $\sim10^{-5}$ to $\sim10^7$, median 64), a spread for which a single global value is utterly infeasible. Two points to be made about this result. First, the plain block variant is not monotone late in the run. Its noise offset in the ratio test is absolute, so once the loss falls to $\sim10^{-5}$ the offset exceeds the loss itself, loss-increasing block trials pass the test, and the trajectory oscillates (final sweep 43.7 dB). A stabilized variant that makes the offset proportional to the current loss and requires an actual decrease for acceptance raises the best to 64.4 dB and the final sweep to 56.3 dB. The remaining fluctuation is due to the architecture of the benchmark used (ViSIR), which trains with active dropout, so the sampled objective is stochastic for every optimizer in the benchmark.

Second, block-coordinate cubic regularization has no $\mathcal{O}(\varepsilon^{-3/2})$ guarantee of the Cartis et al. theory. The closest published block-cubic method, the randomized block cubic Newton of Doikov and Richtárik (2018), samples random blocks and its convergence theory covers convex composite objectives, not this deterministic non-convex case, and its greedy per-block acceptance requires one extra function evaluation per block per sweep. The ablation nevertheless gives the following answer. Per-block adaptive regularization, the architecture that CubicKrylov instantiates with its M_b rule (adapted from gradient information alone, without the per-block function evaluations), is what makes it perform well on this large scale ViSIR architecture with a dropout, and the contrast between the results for the two ARC variants at 91.4M ViSIR are due to this as well.

A gradient-based acceptance rule. The per-block function evaluation required by the ratio test on every block trial is the one cost of the block-ARC design that CubicKrylov's M_b rule avoids, and the reduced-operator method of Doikov and Nesterov (2025) suggests a middle ground, to accept a search step when the *computed gradient* at the trial point matches the cubic model's prediction of it, a test whose ingredients (two gradients) the sweep computes anyway. Transplanted verbatim, the test fails. In single precision the true reduced operator sinks below the gradient-noise floor as the loss approaches $\sim10^{-5}$, noise passes the test on loss-increasing trials, and the nano run collapses to 10.8 dB. And, for the same reason, the plain ratio test loses monotonicity above, now amplified because the gradient passes the noise through undamped. Two guards repair it. Accept on floor when the reduced operator is within gradient-noise tolerance of zero (the model is then exact to measurable

precision, and σ_b is left untouched), and a monotone guard that rejects any trial whose computed loss increases beyond a relative noise tolerance. So guarded, the rule reaches 36.49 dB on the nano equal-budget comparison (Table 30), level with the variants that test the minimized function value, and its behavior on the full multi-image benchmark matches the other ARC variants (Appendix N, the hybrids at full scale).

N The Cost Structure of the Subsolver, Stored Basis (CubicKrylov) against Chebyshev Recurrence

The cubic subproblem of Section 3 is, at its core, a family of shifted linear solves (Section 2.2), and both solver families considered in this paper build their steps in the same Krylov subspace at one HVP per degree, namely the Lanczos process with a stored basis (Lemma A.4) and the Chebyshev second kind three term recurrence Eq. (22) with three vectors of state. It is therefore natural to ask whether replacing the stored basis by the recurrence, the substitution DSO makes for its preconditioned gradient step, would make the cubic step, or the ARC reference of Appendix M, faster. The answer is divided into two distinct costs, memory and per-HVP overhead, where the recurrence is cheaper, and the number of HVPs required to reach a given model decrease, where the stored basis is provably cheaper, by a mechanism specific to the cubic subproblem.

Benefits of using the three term recurrence. The Lanczos solver must retain $\mathbf{Q}_L$ to reconstruct the step $s_L = \mathbf{Q}_L y^\star$, that is, $L{+}1$ vectors (the basis and one work vector) of n_b float64 entries, $\approx$11 GB for the 88.5M-parameter ViSIR decoder at $L = 15$, plus full reorthogonalization at $O(Ln_b)$ inner products per Lanczos step, each a global reduction. The recurrence keeps three n_b-vectors regardless of degree, performs no inner products beyond an optional residual norm check that can stop the sweep before the degree limit, and streams plain `axpy` arithmetic, a memory-access pattern that GPUs execute at close to peak bandwidth. When the basis footprint is the limiting factor, with higher degree limits, larger single tensors, or several models resident on one device, the recurrence is the better option of the two.

Downsides of the three term recurrence. The substitution loses two structural advantages of the stored basis. First, per HVP the Krylov step is optimal, since by Corollary A.5 the Rayleigh–Ritz solve extracts the exact cubic model minimizer over $\mathcal{K}_L(\mathbf{H}, g)$, whereas any fixed-coefficient recurrence step is one particular polynomial in that same subspace. The recurrence moreover needs its spectral interval in advance, estimated by extra probe HVPs (power iterations or Hutchinson probes) whose cost precedes the solve, and its minimax optimality holds only for the assumed interval, while Lanczos adapts to the computed spectrum and right-hand side with no prior bounds. Second, and specific to the cubic model, the Krylov subspace $\mathcal{K}_L(\mathbf{H}, g)$ is independent of the spectral shift λ and of the regularization weight. With $\mathbf{T}_L$ stored, every trial value of λ in the secular iteration of Lemma A.1, and every re-solve after a rejected step changes M_b or σ, costs $O(L)$ scalar arithmetic on the tridiagonal eigendecomposition and *zero* new HVPs. A recurrence has no stored subspace to revisit, so each secular trial and each rejection retry is a fresh degree-L sweep, so the HVP count multiplies by the number of secular iterations (bracketing solvers typically need several). Since one HVP is a double-backward pass through the network and dominates wall clock time at scale, this multiplication generally outweighs the per-iteration savings.

Application scopes for the solvers. The results of Appendix M show which of the two costs dominates at scale. At 91.4M parameters the subproblem solve is not the bottleneck. With the basis stored, the HVPs are spent once per step to build the subspace, every secular trial afterwards costs only $O(L)$ scalar arithmetic, and an accepted ARC-global step completes in $\approx$0.3 s (for this benchmark on this hardware). The failure observed at this scale is due to the single global σ. Replacing the stored basis by the recurrence would therefore relieve the one constraint the basis does impose, its $\approx$11 GB memory footprint, while multiplying the HVP cost of every secular solve. The recurrence is instead preferable in the cases when the spectral bounds are cached and EMA updated across steps (Section 1), so the estimation cost is spread across steps, and a preconditioned *gradient* step that requires no secular iteration, so each update is exactly one sweep.

The evaluated hybrid designs. We implemented several hybrid designs. One of them uses the cubic step of Algorithm 1 with the Lanczos solve replaced by Chebyshev sweeps inside the secular iteration. Then, we evaluated it under the equal-budget nano experimental design of Table 30. The spectral interval is computed by Hutchinson probes, either re-estimated at every block step (*fresh bounds*, as

in DSO-BlockHess) or EMA-cached across steps with a warm-started shift (*EMA-cached bounds*, as in DSO-Adaptive). A first version (v1) precomputes the degree $L = 1.3\sqrt{\kappa}$ inside a doubling-bracket secular solver, and a second (v2) adds an early exit that stops the sweep once the residual norm falls below its threshold, turning the precomputed degree into an upper bound rather than an exact sweep length, and a slope-bounded bracketing that exploits $\varphi'(\lambda) \leq -1$ to compute the secular root after one evaluation. The results substantiate the HVP argument above with actual numbers. The best hybrid, v2 with EMA-cached bounds, reaches 35.18 dB at 14.4 HVPs per block step, since the shift passes the acceptance test in a single sweep at the steady state, the expected behavior of Chebyshev recurrence from DSO. Fresh bounds cost 24.4–34.3 HVPs per block step and 0.3–1.0 dB more, and v1's full-degree sweeps under EMA caching (without the early exit due to the residual norm) give the worst PSNR value of the four (33.55 dB). All four variants run in $O(1)$ solver memory, three recurrence vectors, no stored basis, and all four trail the stored-basis CubicKrylov (36.64 dB) by 1.5–3.1 dB at the same budget. With the basis stored, every secular trial and every rejection retry after an M_b update re-solves on the stored tridiagonalization at zero HVPs, while the recurrence performs a fresh sweep, 51k–92k rejected block trials in these runs.

The trade-off is therefore $\sim$1.5 dB at equal oracle budget for a degree independent memory footprint. At nano scale ($\sim$ 15k parameters) that saving does not give anything, since the $L{+}1$ Lanczos vectors of a block fit comfortably in memory, so the stored-basis solver, CubicKrylov, wins outright over all four hybrid variants (v1 and v2, under fresh and EMA-cached bounds alike). The hybrid becomes the viable variant only at scales where storing $L{+}1$ vectors of the block dimension exceeds the memory limit, as it does for the 91.4M-parameter model of Appendix M with the $\approx$11 GB basis footprint quoted above, while the recurrence keeps three vectors at any degree.

Refined hybrids. Three refinements of the v2 hybrid were further experimented with. The first (*v3*) adds a linear regularization floor to the secular solve, the doubly regularized subproblem of the reduced-operator method of Doikov and Nesterov (2025) has a linear term $\alpha\|s\|$ alongside the cubic one, and importing it as a tolerance-derived lower bound $\alpha = \sqrt{M\,\tau\,\|g_b\|}$ on the spectral shift removes the near-singular shifts at which the recurrence needs its highest degrees. The second (*v2r*) replaces the fixed refresh schedule of the EMA-cached spectral bounds by a gradient-triggered one. The gradient-normalized smoothness of Semenov et al. (2025) measures how far a cached curvature model remains valid in units of the current gradient norm, so the Hutchinson probes are re-spent exactly when $\|g_b\|$ has dropped by a fixed factor since the last estimate (with a hard bound on the visit count). The third (*grS*) is the most aggressive. The gradient regularized step of the same paper replaces the secular iteration outright by the closed form shift $\lambda = \|g_b\|/\gamma_b$, reducing every block step to exactly one recurrence sweep. Under the equal-budget nano experimental design the ordering is v2r (35.46 dB, 13.2 HVPs per block step), v3 (35.34 dB, 18.2), v2 EMA-cached (35.18 dB, 14.4), grS (34.93 dB, 12.8). The gradient-triggered refresh is the one refinement that beats the fixed schedule on both quality and cost, while the closed form shift obtains its lower HVP count at the cost of 0.5 dB.

The hybrids at full scale. The three refined hybrids were then run unchanged on the 91.4M parameter ViSIR, their intended scale, first on the single image experimental design of Appendix M (two-hour limit), then on the ESM-20 benchmark (20 images $\times$ 3 seeds, 150 steps each, the rule of Appendix I.5 at fixed step count). On the single image, v2r reaches 64.2 dB best PSNR, v3 62.9 dB, and grS 59.8 dB, at a peak allocation of 7.0–8.7 GB *for the entire training run*, the $(L{+}1)$-vector Lanczos basis of the ARC runs occupies $\approx$11 GB on the decoder block alone, and grS indeed runs at 1.0 HVP per block step. On ESM-20 the comparison results in the following reliability ordering. v2r averages 63.8 ± 4.0 dB and v3 61.2 ± 4.0 dB with *zero* failures across the 60 runs, both above the stored-basis CubicKrylov reference (58.5 dB) and below the per-block ARC references (67.0–67.4 dB), while grS fails outright on 6 of 60 runs (stalling at $\sim$4 dB, with 60.4 ± 6.6 dB over its successful runs).

The closed form shift, in other words, is cheap and usually sufficient, but its reliability depends entirely on a single scalar, γ_b, the per-block gradient-normalized smoothness estimate that converts the current gradient norm into the spectral shift $\lambda = \|g_b\|/\gamma_b$.

When the cached γ_b is stale or inaccurate, the shift is wrong, and grS applies the resulting step anyway, because replacing the secular iteration removed the cubic acceptance test along with it. The secular solving hybrids v2r and v3 keep that test and reject such steps instead of applying them. The full scale runs thereby confirm both halves of the nano conclusion. Where the stored basis fits in memory, the experimental setups using the CubicKrylov subsolver still lead. Where solver memory

Table 13: Subsolver comparison at equal solver memory on the 91.4M parameter ViSIR (single image, seed 0). The time-limited rows use 2 h wall clock and an fp32 HVP oracle, and the full-budget run rows are the recorded 150-sweep runs of Section 6.1. Both best and final PSNR are quoted because the bounded runs oscillate late. Run-to-run variation of this single image fit spans several dB (Appendix U).

Subsolver	$L_{\max}$	best PSNR (dB)	final PSNR (dB)	peak GB	gevals
Lanczos basis	4	74.1	43.3	9.6	2.8×10^6
Lanczos basis	10	**78.6**	67.1	9.6	2.8×10^6
Lanczos basis	21	69.4	63.4	9.6	2.8×10^6
Lanczos basis (full-budget run)	15	64.4	—	∼11	—
Chebyshev recurrence (full-budget run)	free	64.2	—	7.0–8.7	—

Table 14: bf16 HVP-oracle robustness at matched configurations. Nano runs use the 6.53M-geval budget and a 2,400 s limit, 91.4M runs a single image and a 2 h limit (basis at L=15). The fp32 references are the per-block ARC and CubicCheby-DSO v2r rows of Table 30 (nano) and the full-budget run runs of Table 13 (91.4M).

Scale	Subsolver	fp32 (dB)	bf16 (dB)	Δ	residual (med/max)
nano	Lanczos basis	36.4	30.5	−5.8	1.8×10^{-5} / 0.17
nano	recurrence	35.5	26.5	−9.0	0.75 / 0.98
91.4M	Lanczos basis	64.4	63.5	−0.9	—
91.4M	recurrence	64.2	59.7	−4.5	—

is scarce, the refined recurrence hybrids become competitive. The best of them (v2r) comes within 3.6 dB of the ARC leaders at a third of their solver memory and ∼30% less wall clock time per run.

N.1 Comparison at Equal Solver Memory

The CubicKrylov subsolver (stored-basis Lanczos) keeps $L{+}1$ float64 basis vectors of the block dimension, so on the 88.5M parameter decoder block each vector costs ∼0.708 GB and a solver memory limit B_{mem} bounds the degree to $L_{\max}(B_{\text{mem}}) = \lfloor B_{\text{mem}}/0.708\,\text{GB}\rfloor - 1$. The three term recurrence holds three block sized vectors at *any* degree. We therefore compare the two subsolvers at equal solver memory, ARC-block (stabilized) at $L_{\max} \in \{4, 10, 21\}$ (budgets ∼4/8/16 GB) against the degree-free recurrence, on the 91.4M single image experimental design (image 0, seed 0, 2 h limits), plus degree-bounded nano experiments ($L_{\max} \in \{2, 4\}$, 6.53M-geval budget) probing the terminal phase where the effective condition number approaches κ.

Three observations. First, at these degree bounds the measured peak memory is identical (9.6 GB) across $L_{\max} \in \{4, 10, 21\}$, because the HVP autograd graph on the 88.5M parameter block dominates the memory footprint, so the formula for $L_{\max}(B_{\text{mem}})$ enforces a bound only above $L \approx 15$ (this run required ∼11 GB). Second, accuracy at equal wall clock is not monotone in the degree, $L_{\max} = 10$ attains the best value, so a bounded basis does not degrade below the degree-free recurrence at the same memory size. Third, the same bounds at nano scale (6.53M-geval budget) lose less than 1 dB, 35.6 dB at $L_{\max} = 2$ and 36.2 dB at $L_{\max} = 4$ against 36.4 dB for the unbounded per-block ARC of Table 30. The recurrence's remaining advantage is therefore its constant, degree-independent memory footprint, observed at lower memory requirement 7.0–8.7 GB against ∼11 GB at similar accuracy in these experiments.

N.2 Robustness to a bf16 HVP Oracle

Mixed-precision training makes the Hessian–vector oracle itself noisy. We rebuild the per-block loss, gradient, and HVP graph under `bfloat16` autocast (fp32 master weights, with accept-or-reject function evaluations kept in fp32 to isolate subsolver numerics) and rerun both subsolvers on the nano experimental design and the 91.4M single image experimental design. Both solvers project the same noisy operator, but the CubicKrylov solver (stored basis) additionally relies on the mutual consistency of L successive HVP columns, while the recurrence uses each HVP once with fixed coefficients. The measured quantities are the best-PSNR delta against the fp32 references, the rejection rate (bf16 noise reaches the per-block ratio test through the model decrease), and the sampled stationarity residual $\|\nabla m_b(s)\|/\|g_b\|$ of the returned step.

Table 15: 2D image fitting at convergence (mean best PSNR, dB). First-order experimental setups use 20 images and second-order experimental setups images 0–4 (ARC-block on SIREN 0–1, WIRE 0–1 for all setups). Restricting the first-order means to the same images changes them by less than 0.4 dB. conv a/b means the run reached the plateau criterion on a of b images, and $+N$ that N runs ended at the time limit while still improving (their PSNR is a lower bound).

Arch	method	PSNR	H/L	steps	gevals	conv
FINER	Adam	65.2	0.22	112k	1.1e5	20/20
FINER	SOAP	75.1	0.05	200k	2.0e5	20/20
FINER	ARC-block	**124.7**	1.22	15k	3.7e6	2/5+3
FINER	CubicKrylov	86.8	1.32	35k	4.3e6	5/5
SIREN	Adam	70.7	0.93	150k	1.5e5	20/20
SIREN	SOAP	71.4	0.05	218k	2.2e5	20/20
SIREN	ARC-block	69.9	2431*	35k	1.1e7	2/2
SIREN	CubicKrylov	58.7	502*	34k	4.2e6	5/5
WIRE	Adam	30.5	0.04	103k	1.0e5	2/2
WIRE	SOAP	31.6	0.07	100k	1.0e5	2/2
WIRE	ARC-block	29.4	0.01	2.2k	4.7e5	2/2

Table 16: SDF fitting, Thai statue, IGR supervision, 100k training samples (one run per row, seed 0). conv = plateau criterion reached ("limit" means that time limit reached while still improving).

Backbone	method	chamfer-L1	train dB	test dB	steps	conv
FINER	Adam	0.0842	76.1	27.3	100k	yes
FINER	SOAP	0.0153	82.1	34.5	165k	yes
FINER	ARC-block	0.0177	**131.0**	34.0	9.1k	yes
FINER	CubicKrylov	0.0185	106.8	32.5	11.9k	yes
SIREN	Adam	0.0092	92.7	47.7	178k	yes
SIREN	SOAP	**0.0089**	80.8	47.1	282k	yes
SIREN	ARC-block	0.0135	71.8	41.2	14.6k	limit
SIREN	CubicKrylov	0.0102	61.3	**49.5**	17.4k	yes

The mechanism evaluation shows that under bf16 the sampled basis orthogonality stays at fp64 roundoff ($\max|Q^\top Q - I| \approx 10^{-16}$–$10^{-13}$), while the returned steps' stationarity residuals grow (to 0.17 at the tail for the basis and to 0.75–0.98 throughout for the recurrence) and the nano rejection rate rises from below 1% to 27% for the basis solver, the damage comes from the projected model and the recurrence's fixed-coefficient solve, not from lost orthogonality. The recurrence loses more at both scales (-4.5 against -0.9 dB at 91.4M), because it applies each noisy Hessian–vector product once, with coefficients fixed in advance, whereas the adaptive basis solve can partially compensate. Mixed-precision deployment of either subsolver therefore requires an fp32 HVP implementation.

O Convergence Matched INR and SDF Study

In this section we evaluate the method against first-order methods on prominent INR architectures, SIREN (Sitzmann et al., 2020), FINER (Liu et al., 2024), and the real-valued Gabor form of WIRE (Saragadam et al., 2023), on 240×240 RGB image regression (one network per image, 3 hidden layers of width 256, ∼199k parameters), and fit signed distance functions of the Thai statue with IGR supervision (Gropp et al., 2020) on the SIREN and FINER backbones. Adam and SOAP are tuned as in the corresponding source papers and run until convergence under a plateau criterion. The second-order experimental setups are the ARC-block cubic step and CubicKrylov, run to their own plateaus. Costs are reported in gradient equivalents. The metric H/L is the residual FFT power ratio (Mid-High+High)/(DC+Low), near 1 for band uniform residuals.

*H/L is a within-run ratio. On SIREN the second-order runs drive the DC and Low residual to the machine-precision floor (∼10^{-7}–10^{-6} per coefficient, 3–6 orders below Adam) while matching Adam's high-band accuracy, so the large values arise from the denominator. Judged on absolute per-band residual power, ARC-block is at or below Adam in every band on image 0. Total MSE is dominated by the coefficient-rich High band.

The comparison distinguishes two cases. On FINER the separation is a matter of reachability. Converged Adam plateaus at 65.2 dB in 2D and at a chamfer error 4.6–5.5× above every curvature-

aware method in 3D, and no additional budget affects these results. The blockwise cubic step continues to 120–129 dB with a band uniform residual. On SIREN the methods are comparable at convergence (CubicKrylov reaches the best SDF test accuracy, 49.5 dB), and the wall clock comparison favors the first-order method, CubicKrylov required $8.3\times$ Adam's wall time.

P NTK Loss-Curve Forecast, Its Closed Form, Calibration, and Limits

In the infinite-width limit the residual decays in the kernel eigenbasis (Jacot et al., 2018; Lee et al., 2019), which gives the loss of full batch gradient descent at every step t in closed form,

$$L_{\text{pred}}(t) \;=\; \frac{1}{N}\sum_{k}(1-\eta\mu_k)^{2t}\,c_k^2,$$

over the eigenvalues μ_k of the empirical tangent kernel at initialization, the initial residual coefficients c_k in the corresponding eigenbasis, the step size η, and the sample count N. Each mode drains at a rate proportional to μ_k, which is the mechanism behind the band-ordered convergence of Section 6.3. The form has been used predictively for coordinate networks, where the kernel spectrum explains why Fourier features widen the set of learnable frequencies (Tancik et al., 2020), and to forecast training time before training (Zancato et al., 2020).

Calibration. We built the kernel once at initialization and never refreshed it, on width-16 MLP fits (881 parameters, 500 samples) to three analytic targets, Rosenbrock–Ackley, Chebyshev–Rosenbrock, and multi-saddle, the small benchmarks of Section K. Two readings follow. First, the residual ratio $s = \left\|\mathbf{H} - \frac{2}{N}J^\top J\right\|_F / \|\mathbf{H}\|_F$, with J the network Jacobian on the samples, is 1.000 on all three surfaces at initialization. The residual term constitutes the Hessian and the Gauss–Newton term is about a percent of its spectral scale ($\mu_{\max} = 4.4$ against $\lambda_{\max} = 232$–327), with 115–145 of the 881 Hessian eigenvalues negative. Along tuned Adam, s falls to 0.004–0.008 by mid-training (30k of 60k steps) and to 0.001–0.004 at the end, while the negative eigenvalue count goes to zero. The saddle rich curvature that the fingerprint of Section 6.2 measures is the residual term, and it decays. Second, the fixed-kernel stability edge is not a usable step size. Gradient descent at $0.5/\mu_{\max}$ diverges within about 50 steps, and the runs reported here use $\eta = 0.1/\mu_{\max}$. Within that window the lazy forecast tracks full batch gradient descent to within a factor of 1.36–1.91 (measured over predicted loss) at 20k steps on the three surfaces, and the factor does not grow over the window. The forecast predicts the slow descent of gradient descent on these surfaces correctly. It says nothing about Adam, whose loss at the same step count is 6–$280\times$ lower across the three.

Limits. What is not established is whether this survives the widths, step sizes, and optimizers of practice. The linearization is documented to break at finite width and large learning rate (Fort et al., 2020), and FINER's own tangent kernel analysis is qualitative (Liu et al., 2024). The published loss-curve predictors are either fitted from partial runs (Luo et al., 2025) or meta-learned across architectures (Pourali et al., 2026), so neither is a forecast from initialization. The kernel view has been extended to physics-informed training, where sub-kernel traces set each loss term's convergence rate and drive adaptive loss weighting (Wang et al., 2022), and to regularized Newton and trust region methods, whose limiting kernel yields loss decay uniform across the spectrum rather than the spectral bias of gradient descent (Riedl et al., 2026). That result is asymptotic, full space, and infinite width, whereas ARC-BLOCK acts at finite width on block restrictions with measured curvature, and the band uniform residuals of Tables 5 and 15 are the finite-width observation the asymptotic result points at. A phase based switch from a preconditioned second-order method to a first-order one has been recommended qualitatively (Jiang et al., 2026). The fingerprint of Section 6.2 is the per-block, measured form of that decision. The forecast has not been scored at the 199k or 91.4M parameter tiers of this paper, and we do not assume it transfers.

Q Extended Comparison with ARC-Based Approaches

This appendix reviews, family by family, the ARC-based methods summarized in Tables 11 and 12 of Appendix L.

Cubic model minimization in Krylov subspaces. Bellavia et al. (2025) compute the ARC search step by minimizing the cubic model in a low-dimensional subspace, polynomial or rational Krylov,

that is reused across several iterations, and retain the worst case complexity of classic ARC. Their construction operates in the full space with a single regularization parameter and is aimed at problems where sparse factorizations or direct solves are available. The recurrence of this work instead reuses per-block Krylov bases under a per-block σ_b with Hessian–vector products only. Al Daas and Gould (2025) observe that the solutions of trust region and norm-regularization subproblems, viewed as functions of the regularization weight, span a very low-dimensional subspace, and build a basis for the whole family by an extended Krylov iteration requiring a single matrix factorization, so one subspace then serves every value of the weight. The observation matches the needs of an adaptive cubic constant, but the factorization step is not available in the matrix-free case considered here. Tansley and Cartis (2025) adapt the dimension of a random subspace to the rank of the projected Hessian and retain the full-dimensional cubic regularization rate. Random projections replace the deterministic tensor blocks used here. Zhou et al. (2025) reach the same $\mathcal{O}(\varepsilon^{-3/2})$ complexity with a quadratic local rate by regularizing the Newton system with a quantity built from current and previous gradients and solving it by conjugate gradients with a negative curvature monitor. Their method is parameter-free but adapts one global regularizer rather than one per block.

Blockwise and partition-based cubic methods. Cristofari (2026) selects one block per iteration by a greedy Gauss–Southwell rule on the stationarity violation and approximately minimizes a cubic model of that block (the method is named IBCN, inexact block cubic Newton), with $\mathcal{O}(\varepsilon^{-3/2})$ iterations to drive the selected block's stationarity violation below ε and $\mathcal{O}(\varepsilon^{-2})$ for all variables, improving on the $\mathcal{O}(\varepsilon^{-3})$ of cyclic high-order block descent (Amaral et al., 2022). ARC-BLOCK instead updates all blocks in every step, each with its own cubic constant and acceptance test. Appendix R.2 measures IBCN's selection rule inside the ARC-BLOCK outer scheme. Wolinski (2025) computes exact projections of the Hessian and of third-order derivatives onto the subspaces induced by a partition of the parameters and outputs one learning rate per subset. The per-tensor granularity is shared with ARC-BLOCK, but no cubic model, regularization constant, or acceptance test is attached to a subset.

Quasi-Newton cubic models with closed form subproblems. Ranganath et al. (2025) combine limited-memory symmetric-rank-one curvature approximations, which admit indefiniteness and negative curvature, with a cubic scheme whose subproblems have closed form solutions for suitable regularization choices. The ARCLQN solver of Forristal et al. (2022) solves the cubic subproblem exactly and matrix-free by exploiting the internal structure of limited-memory quasi-Newton matrices. Both replace the exact block Hessian with a low-rank surrogate. The CubicKrylov and φ_1 subsolvers of this work apply the exact blockwise Hessian through Hessian–vector products.

Stability of Chebyshev recurrences under Hessian variation. Pasechnyuk-Vilensky and Takáč (2026) quantify the ℓ_2 gain of a prefix-exact Chebyshev recurrence under time-dependent Hessian perturbations and prove the bound sharp in the causal two-term class. This analysis addresses the same failure mode that the negative-mode clamp (Section 5.2) and the trust-controlled horizons of the φ_1 subsolver control in practice, but their bound is proven for the recurrence in isolation under an adversarial perturbation model. This paper instead couples the same recurrence to a per-block adaptive cubic constant and an outer acceptance test, and demonstrates the resulting stability empirically at the 91.4M parameter scale (Appendix M) rather than through a closed form gain bound.

Parameter-free and inexact-Hessian ARC theory. Cartis et al. (2025) approximate the highest-order tensor of a p-th-order regularization method from lower-order derivatives and refresh the approximation every m steps while retaining complexity to second-order stationarity, theoretical cover for reusing stale per-block curvature between refreshes. Marumo and Takeda (2026) obtain complexity bounds without backtracking or acceptance tests by regularizing the local model with an exponent above the model-error order, and Fang et al. (2026) give a parameter-free cubic regularized Newton method with sharp complexity under a generalized smoothness condition. Both remove the constants that the per-block ratio test of ARC-BLOCK estimates online. Shestakov et al. (2025) analyze adaptive regularized Newton methods under a range of inexact Hessian models and Bregman geometries, complementary theory for approximate block curvature, but their experiments do not extend past moderate-dimensional problems. The per-block Krylov subsolver of this paper is instead evaluated up to a single 88.5M parameter tensor (Appendix M).

Stochastic, variance-reduced, and lazy cubic Newton. Pasechnyuk-Vilensky et al. (2025) give a variance-reduced cubic Newton method for finite sums whose analysis assumes only an inexact cubic subproblem certificate, which is the interface a per-block Krylov subsolver satisfies. Scheinberg and Xie (2023) prove high-probability complexity bounds for stochastic ARC under probabilistic

zeroth-, first- and second-order oracles, the oracle conditions a noisy per-block acceptance test would need. Chen et al. (2026) quantify the computational gain of reusing Hessians across iterations while keeping near-optimal iteration counts, a cost model directly comparable to reusing a block's Krylov basis across σ_b updates. None of these analyses is combined with a per-block adaptive cubic constant or evaluated on network-scale training. This paper considers only full batch, deterministic training (Section 7), and combining the per-block acceptance test with a stochastic or variance-reduced oracle is left open.

Subspace and multilevel second-order methods. Fuji et al. (2025) prove that the randomized subspace regularized Newton method attains at best a local linear rate in general, an argument for structured per-tensor blocks over random projections. Higuchi et al. (2025) reach $\mathcal{O}(\varepsilon^{-3/2})$ iterations with a homogenized trust region subproblem, an eigenvalue problem solvable by Krylov methods and solved as a single problem rather than per block. Tsipinakis et al. (2026) prove super-linear rates for stochastic low-rank Newton methods through a multigrid correspondence, with a truncated variant aimed at saddle escape in high dimension. Curvature is compressed by level rather than partitioned by tensor.

Krylov-type inner solvers for regularized Newton systems. Zeng et al. (2026a) adaptively regularize the Newton system inside a conjugate-gradient solver with auto-conditioning, removing the nested line searches of earlier Newton-CG methods under Hölder continuous Hessians. Zeng et al. (2026b) generate descent directions with the minimal-residual Krylov method MINRES and exploit detected non-positive curvature for saddle avoidance. Both spend the same Hessian–vector-product budget as a Lanczos or Chebyshev cubic subsolver but attach it to a quadratic rather than a cubic model.

Second-order methods at deep learning scale. Abreu et al. (2026) apply full Gauss–Newton preconditioning to transformers of up to 150M parameters and find that a layerwise preconditioner ignoring cross-layer information nearly matches full Gauss–Newton, direct empirical support for partitioning curvature by tensor, obtained for a Gauss–Newton rather than a cubic model. Martins Gomes (2025) preconditions with a per-layer block-Kronecker approximation of the Fisher information matrix. The curvature within a block is factored, whereas ARC-BLOCK applies the exact block Hessian through Hessian–vector products.

Trust region and ARC hybrids. Hamad and Hinder (2024) give an adaptive trust region method with inexact subproblem solves whose complexity bound improves earlier trust region results, benchmarked head-to-head against ARC. Ha et al. (2026) add an adaptively regularized local model to adaptive-sampling trust region optimization and reach almost-sure $\mathcal{O}(\varepsilon^{-3/2})$ iteration complexity, up to logarithmic factors, under subexponential noise. Both adapt a single global radius or regularizer, with no per-block granularity.

Cubic regularization beyond unconstrained training. Cubic regularization has also been applied to problem classes not considered here, namely Riemannian adaptive regularization with an explicit count of Hessian–vector products under subproblem solver inexactness (Zhang and Jiang, 2025), equality-constrained sequential ARC descending from the multi-shift ARC$_q$K line (Pei et al., 2025), parameter-free cubic regularization for convex-concave minimax problems (Wang et al., 2025), and federated cubic Newton with differential privacy (Huo et al., 2024). These works transfer the ARC template to other problem classes. None partitions the regularization within a single training problem.

R Two Recent Relatives, Measured Inside the Outer Loop

The comparisons of Appendix Q are by design. For two recent methods we also have matched measurements, obtained by placing the foreign component inside the ARC-BLOCK outer scheme of Algorithm 1 so that the A/B isolates one change, the per-block step rule in the case of the primal accelerated Newton method of Doikov (2026), and the block selection rule in the case of the greedy block cubic Newton method of Cristofari (2026). Both were run on the FINER and SIREN networks of Appendix O (240×240 image 0, n=198,915, B=10 tensors), with the sweep of Section 6.3 (log-spaced grids, geometric extension of any axis whose winner lies on the grid edge, confirmation at the full step allowance on images 0 and 7). The two experiments produced the same architecture split, which is the single most useful observation from them and feeds the dispatch rule of Section 6.2.

R.1 THE PRIMAL ACCELERATED NEWTON STEP AS A SUBSOLVER, ARC-BLOCK-PAMM

Doikov (2026) minimizes convex f with Lipschitz Hessian by one shifted linear solve per iteration, $x_{k+1} = y_k - (\nabla^2 f(y_k) + \lambda_k \mathbf{I})^{-1} \nabla f(y_k)$, with momentum $y_{k+1} = x_{k+1} + \beta_k(x_{k+1} - x_k)$ and the predetermined schedule $\lambda_k = 3L_2R\,(k+3)/((k+1)(k+2))$, $\beta_k = k/(k+4)$, and reaches the accelerated $\mathcal{O}(1/k^3)$ rate in functional residual without a cubic subproblem, a parameter search, or dual corrections. The solve may be inexact and Hessian-free. It is a convex-class method, so on the non-convex landscapes of this paper there is no guarantee to lose, only a step rule to measure. ARC-BLOCK-PAMM keeps the outer scheme of Algorithm 1 verbatim (per-block σ_b, the stabilized ratio test, the monotone guard) and replaces the Lanczos cubic solve by one conjugate gradient solve of $(\mathbf{H}_b + \lambda_b \mathbf{I})s_b = -g_b$ in double precision, at most k_{cg} iterations, with three deviations from the source, each of them measured. (i) The shift is set by the trust parameter, $\lambda_b = \sigma_b$, in place of the $1/k$ schedule. The scheduled shift is $\sim$450$\times$ worse in terminal loss on the 881-parameter benchmark of Section K, because it cannot respond to the landscape and the accept/reject feedback can. Dimensionally σ_b multiplies $\|s\|^3/3$, so the cubic step's effective shift is $\sigma_b\|s\|$ and $\lambda_b = \sigma_b$ is a units mismatch. The gradient-normalized form $\lambda_b = \sqrt{\sigma_b\|g_b\|}$ (Semenov et al., 2025) was also run and changes SIREN by $+0.6$ dB and FINER by -0.5 dB at 40 sweeps, so the mismatch is not what decides the comparison below. (ii) Momentum is off. Folded into the trial displacement so that the ratio test covers it, it still loses (38$\times$ worse loss on the same benchmark, rejections $43 \to 767$), because the Gauss–Seidel sweep updates every other block between a block's consecutive visits and the stored displacement is a stale direction. (iii) A conjugate gradient breakdown (nonpositive curvature, i.e. $\mathbf{H}_b + \lambda_b \mathbf{I}$ indefinite) is a rejection signal that raises σ_b and escalates λ_b within the trial.

Standalone, at the same number of steps. Before the swap, the source method itself was run against ARC-φ_1 on the three small benchmarks of Section K, 500 steps each, with L_2R tuned on a log grid and extended past any edge, namely pan-dense (exact factorized solve, reference only, since a dense Hessian costs n HVPs per step) and pan-cg (the inexact solve of its Section 5). Table 17 gives the best loss. On the two non-convex surfaces the shifted solve breaks down on most steps (336 and 52 of 500) and the method stays within a factor 2–8 of the initial loss while ARC-φ_1 improves it by six orders. On the 35k SIREN, where every solve completes (0 breakdowns, 79 CG iterations per step on average), pan-cg passes ARC-φ_1 late. ARC-φ_1 leads by four to five orders at step 72 (3.8×10^{-10} against 1.4×10^{-5}), still leads at step 372, and is overtaken between steps 372 and 492. The breakdown count in the first sweeps predicts which case applies, at no extra cost. Oracle counts differ by arm (ARC-φ_1 1.0–1.9×10^5 gevals against 2.4–40×10^3 for pan-cg), so these rows compare the arms at the same number of steps, and the comparison at equal oracle calls is the one below.

Table 17: Primal accelerated Newton (PAMM) against ARC-φ_1 on the small benchmarks, 500 steps, best loss, L_2R tuned per row. pan-dense uses an exact factorized solve and is a reference, and pan-cg is the Hessian-free variant. ARC-φ_1 uses σ_0=1, $L_{\max}$=15, c_h=3. Adam tuned on a log grid.

benchmark (n, f_0)	ARC-φ_1	pan-cg	pan-dense	Adam	CG breakdowns	mean CG its
Rosenbrock–Ackley (881, 1.9×10^4)	$\mathbf{2.3\times10^{-2}}$	1.0×10^4	5.4×10^3	1.2×10^4	336/500	3.8
multi-saddle (881, 6.7×10^3)	$\mathbf{2.0\times10^{-3}}$	3.1×10^3	8.9×10^2	3.9×10^3	52/500	4.0
35k SIREN (35,543, 4.6)	3.3×10^{-10}	$\mathbf{8.1\times10^{-13}}$	–	1.4×10^{-4}	0/500	78.8

As a subsolver, at equal oracle calls and at the same number of steps. Table 18 reports ARC-BLOCK-PAMM against the Lanczos cubic step (ARC-BLOCK) and the φ_1 step (ARC-φ_1) under two experimental designs, every arm stopped at 6×10^4 gradient equivalents on image 0, and the 500-step confirmation of the tuning sweep on images 0 and 7 with both tuned axes extended until the optimum was interior (k_{cg} 100/200/300 gives 50.61/49.49/47.08 dB on SIREN, and σ_0 3/10/30 gives 55.81/56.57/56.37 on SIREN and 69.75/72.05/58.76 on FINER). Both designs agree on direction. On SIREN the PAMM step wins, $+1.35$ dB over the Lanczos step and $+2.13$ over φ_1 at equal gevals, $+2.25$ over the tuned Lanczos step at equal steps, and does so in 252 s against 1,724 s for the Lanczos arm at identical gevals. Equal gevals are not equal wall clock, since Lanczos also requires full reorthogonalization, $\mathcal{O}(L^2 n_b)$ vector work and a $k \times k$ eigendecomposition per trial, while conjugate gradients is $\mathcal{O}(n_b)$ per iteration in four vectors. On FINER the result inverts, -7.37 / -7.50 dB at equal gevals and -6.62 at equal steps, with 138–330 rejections against 15–30 on SIREN and a non-monotone response to k_{cg} (optimum at 30, breakdowns 27/35/45 at depth 10/30/100), since deeper solves find the negative curvature faster and the step itself does not improve. At the same number of steps the PAMM arm also spends 2.2$\times$ the gevals of the L=50 Lanczos arm (3.3×10^5

against 1.5×10^5), so its SIREN margin there is obtained at the cost of oracle calls. The rows at equal oracle calls remove that advantage.

Table 18: ARC-BLOCK-PAMM against the Lanczos cubic and φ_1 subsolvers in the same outer scheme, image 0 (rows at equal oracle calls) and mean of images 0 and 7 (rows at the same number of steps), best PSNR in dB. k_{cg} is the conjugate gradient iteration limit, rej. the rejected trials, and bd. the CG breakdowns. Wall clock on a shared GPU, so gevals and steps are the exact axes.

experiment	arm	SIREN	wall	rej./bd.	FINER	wall	rej./bd.
6×10^4 gevals	ARC-BLOCK-PAMM, k_{cg}=100	**50.61**	252 s	30/25	48.29	272 s	138/45
	ARC-BLOCK-PAMM, k_{cg}=30	49.02	271 s	16/16	49.69	294 s	222/35
	ARC-BLOCK, L=25	49.26	1,724 s	30/–	57.06	842 s	138/–
	ARC-φ_1, L=25	48.48	410 s	16/–	**57.19**	438 s	70/–
500 steps, tuned	ARC-BLOCK-PAMM, k_{cg}=100, σ_0=10	**56.57**	700 s		72.05	755 s	
	ARC-BLOCK, L=50 (tuned σ_0)	54.32	574 s		**78.67**	636 s	

Mechanism. A per-block diagnosis (40 sweeps, Lanczos degree 40 in double precision, stochastic Lanczos quadrature negative mass over three probes) refutes the first explanation. All ten blocks are positive definite on both architectures (negative mass 0.0, no indefinite block), so FINER is not more indefinite than SIREN, and the residual breakdowns are rounding on ill-conditioned systems. What differs is curvature scale. FINER's two scale-modulation layers (blocks 0 and 1) have $\lambda_{\max}$ of 1.8×10^4 and 4.5×10^4 against 1.4 and 2.7 for the corresponding SIREN blocks. On SIREN σ_b never leaves its 10^{-8} floor on any block, so ARC-BLOCK-PAMM runs as near-exact Newton on a positive definite problem, which is the whole SIREN win. On FINER the ratio test escalates σ_b on those two blocks by fourteen orders and the PAMM step on block 0 comes out $3{,}052\times$ shorter than the cubic step at the same σ_b. That damping is real but is not what costs FINER its 7 dB. Routing blocks $\{0, 1\}$ to the φ_1 step recovers only $+0.86$ dB, a comparison of the same size routing two low curvature blocks $\{3, 5\}$ recovers more ($+1.71$), and routing all ten reproduces ARC-φ_1 exactly (45.63 dB, 43 rejections, 11,564 gevals in both, a correctness check on the routing). The deficit is spread over nearly every block. The φ_1 spectral multiplier beats a constant-shift truncated conjugate gradient step roughly uniformly on FINER's larger curvature blocks, and the untested account is that 30 iterations are uniformly too few on the larger curvature operator. For the dispatch rule this is a second, cheap predictor beside the fingerprint. A subsolver whose shifted solve completes without breakdown and whose σ_b remains at the floor is on a landscape where the cheapest curvature step suffices.

R.2 GREEDY BLOCK SELECTION (IBCN) INSIDE THE OUTER SCHEME

The inexact block cubic Newton method of Cristofari (2026) picks, at each iteration, the coordinate with the largest gradient entry, pads the block to a fixed dimension with uniformly random other coordinates, rebuilds it every iteration, forms the dense block Hessian explicitly, and minimizes a cubic model of that block under one global regularizer that is doubled on rejection and never decreased. Its published block dimensions are 1 to 100, and it affords them because least squares and logistic regression expose the block Hessian in closed form from $A^\top A$. A network does not. Our Hessian–vector product costs one backward pass through the whole model regardless of block size, so the number of blocks multiplies total cost. On FINER, blocks of IBCN's size would need $\sim$1,989 of them to touch every coordinate once, at $\sim$28 gevals each, about 5.6×10^4 gevals per pass where a cyclic sweep spends 280, a factor near 200. The port therefore keeps the ARC-φ_1 subsolver and the σ_b adaptation (raised on rejection, lowered on a very successful trial) unchanged and changes only the selection, behind one code path with three modes. *cyclic* is the sweep of Algorithm 1. *greedy-tensor* is Gauss–Southwell over the existing tensor partition, scoring from the gradient already in hand and updating only the worst offender, by IBCN's own $\max_i |g_i|$, by $\|g_b\|_2$, or by the size-neutral $\|g_b\| / \sqrt{\dim b}$. One update costs $L+3$ gevals, a tenth of a sweep, so greedy and cyclic are comparable at matched gevals. *greedy-pad* is IBCN's anchor-plus-random-padding construction with the block dimension raised until the Hessian–vector product is worthwhile. Its index set crosses tensor boundaries, so it is the only ARC-BLOCK variant that sees cross-tensor coupling (restricting v to the block keeps $(\mathbf{H}v)$ on the block equal to $\mathbf{H}[\mathrm{idx}, \mathrm{idx}]\, v[\mathrm{idx}]$, verified against a dense Hessian slice to 4.4×10^{-16}), with one global σ because transient blocks give a per-block σ nothing to attach to.

Table 19 gives the confirmation. On FINER cyclic wins outright, by 23–26 dB, and is $3.4\times$ faster in wall clock at matched gevals (99 s against 333 s for greedy-pad 20,000 at the 120-step screen).

Table 19: Block selection inside the ARC-BLOCK outer scheme with the ARC-φ_1 subsolver (L=25, c_h=1, σ_0=0.3), 500-step confirmation, mean best PSNR (dB) over images 0 and 7. Screen values (120 steps, image 0) in parentheses where no confirmation was run. greedy-pad block dimension in parameters, where n=198,915 is the whole vector, i.e. no blocking.

	cyclic	greedy-pad, N	greedy-pad, 60,000	greedy-pad, 20,000	greedy-tensor, $\|g_b\|_2$
FINER	**64.79**	42.12	(38.84)	39.03	(41.83)
SIREN	50.14	**52.30**	51.88	(47.95)	50.92

The mechanism is starvation, and it is quantitative. FINER's three 65,536-parameter hidden weight tensors hold 98.8% of the parameters, and the updates landing on them per 1,200 block updates order the outcomes monotonically, cyclic 360 (54.72 dB at the screen), $\|g_b\|_2$ 91 (41.83), $\max_i |g_i|$ 17 (27.66), size-neutral 16 (26.82). IBCN's own $\max_i |g_i|$ rule gave 1,014 of 1,200 updates to a single 256-parameter tensor, because FINER's scale-modulation layers have gradient and curvature two to three orders larger than the rest (the same property that decides Appendix R.1). Coverage has to be counted in parameters. The size-neutral score touched all ten tensors with the least concentrated histogram and scored worst. greedy-pad fails for a different reason. Its single global σ runs away when blocks mix curvature scales (at block dimension 2,000, σ reaches 2.5×10^3 with 730 escape trials and 13.60 dB, and at 20,000 the same quantities are 1.2×10^{-3}, 244, 44.15). Cyclic copes with the same heterogeneity through its per-tensor σ_b, which runs to the floor on eight tensors while holding 9.8×10^3 and 3.1×10^2 on the two modulation layers. Greedy never gives a tensor enough visits for that adaptation to form. On SIREN the result inverts, as in Appendix R.1, where greedy-pad beats cyclic by 2.16 dB and greedy-tensor by 0.78. Two caveats keep this from being a greedy-selection result. The block dimension trend is monotone up to the grid edge and the edge is the whole parameter vector, so the axis cannot be extended, and at that cell the greedy anchor is irrelevant since the block is everything. The winning configuration is full-parameter ARC-φ_1, and the finding is that on SIREN, whose curvature is near uniform across tensors and whose σ_b remains at the floor on every block, no blocking beats blocking at matched gevals. And equal gevals are again not equal wall clock (289 s against 85 s at the 120-step screen). All confirmation cells are two images and one seed.

S Certified Krylov Steps and the Gradient-Regularized φ_1 Horizon

Theorem B.10 covers a CubicKrylov step on the trials where the step-form termination criterion (19) holds (Cartis et al., 2011b, TC.s). Its left side is free for a Lanczos step, $\|\nabla m_b(s_L)\| = \beta_{L+1}|e_L^\top y^\star|$, as is the subspace second-order certificate, the smallest Ritz value of $\mathbf{T}_L$ against the shift $\frac{M_b}{2}\|s_L\|$. The implementation records both on every block trial, together with the realized decrease against the bound $\eta_1 \frac{M_b}{12}\|s_b\|^3$ of Proposition A.10, and offers two stopping rules for the adaptive Lanczos depth, the *gradient form* $\|\nabla m_b(s_k)\| \le \min(1, \|g_b\|^{1/2})\|g_b\|$, which all recorded runs use, and the *certified* step form (19) with $\kappa_\theta = 0.1$. Cartis et al. (2011a) report the same pair, where the step form has the improved complexity, the gradient form gives the better numerical performance. Table 20 measures both on FINER image 0 ($B = 10$, seven tensors of at most 768 parameters and three 256×256 weights) over 500 sweeps at $\sigma_0 = 1$, $\eta_1 = 0.1$.

Three readings. *(i) Which blocks certify.* Under the step form the seven small tensors certify on all of their accepted trials at $L_{\max} = 163$ (six of the seven at 50), because Lanczos exhausts their Krylov space (breakdown at degree 3–35) and the residual identity then vanishes. The three 256×256 weights reach the cap on 96–98% of their trials at every $L_{\max}$ including 163, with median relative residual 4×10^{-6} but a 90th percentile of 0.6. On those blocks the step is a truncated Krylov step whose cubic decrease (Proposition A.10) holds at any degree, while the gradient bound of Lemma B.6 is not certified. The theorem therefore applies to the exact subsolver and to the small blocks as run. For the large weights it states what a certified step would give. *(ii) The certified rule costs accuracy per oracle call.* Pushing every block to the step-form residual raises the rejection rate from 3–13% to 39–48%. The more exact subspace step at the adapted M_b is the step the ratio test rejects more often, and the truncated gradient-form step acts as a second regularizer. At matched 2.87×10^5 gevals the step form at $L_{\max} = 163$ reaches 72.1 dB against 82.65. This is the numerical side of the Cartis et al. (2011a) remark, measured at 199k parameters. *(iii) Rates.* The running minimum gradient

Table 20: CubicKrylov stopping rules on ESM image 0, 500 sweeps. "Certified" is the fraction of accepted trials meeting (19) at $\kappa_\theta = 0.1$. "at cap" is the fraction of accepted trials that reached $L_{\max}$ without meeting the active stopping rule. "bound met" is the fraction whose realized decrease reached $\eta_1 \frac{M_b}{12} \|s_b\|^3$ outright (the rest were accepted inside the guard tolerance t, where the bound is below fp32 loss resolution). The slope is the least-squares exponent of the running minimum $\|\nabla f\|$ against sweep count over the last three quarters of the run. Theorem B.10 guarantees $-2/3$.

Stop rule	$L_{\max}$	PSNR (dB)	gevals	rejected	certified	at cap	bound met	slope
gradient form	25	60.77	1.03×10^5	3.4%	10%	47%	84%	-0.92
step form	25	60.27	1.42×10^5	38.7%	34%	66%	94%	-1.02
gradient form	50	73.16	1.53×10^5	12.5%	10%	29%	84%	-1.27
step form	50	71.40	2.08×10^5	46.9%	58%	42%	94%	-1.36
gradient form	163	**82.65**	2.87×10^5	8.9%	10%	19%	84%	-1.68
step form	163	75.89	3.76×10^5	48.1%	68%	32%	92%	-1.41

decays with exponents between -0.9 and -1.7 in sweeps, all steeper than the $-2/3$ of the theorem and steeper at larger $L_{\max}$, so the worst case bound is not tight here. The per-block decrease bound was met outright on 84–95% of accepted trials, the remainder being acceptances inside the tolerance t of Algorithm 1. Negative Ritz values appeared on 0.3–1.6% of accepted trials, and the subspace second-order certificate $\theta_{\min}(\mathbf{T}_L) + \frac{M_b}{2}\|s_L\| \geq 0$ held on all of them.

Coupling floor on FINER. Remark B.11 estimates $\delta \leq \left\|\nabla^2 f\right\|_F \sqrt{1-\beta}$ from Hutchinson HVPs on the ten parameter tensors. At initialization of the recorded image-0 run ($n = 198915$, $B = 10$), $\beta = 0.165$, $\left\|\nabla^2 f\right\|_F = 2.31\times10^3$, $\delta_F = 2.11\times10^3$, and a power iteration on $P_b \nabla^2 f\, P_{\neq b}$ gives $\delta_{\text{spec}} = 135$, with a Hessian-Lipschitz secant $L_H = 6.7$. After 50 and 250 blockwise sweeps (Table 21) the mass has shifted onto the blocks ($\beta = 0.84$ then 0.78) while δ_{spec} rises to $\sim 6\times10^2$. That operator-norm bound is loose against the leftover the sweep actually leaves, since $\hat\delta = \|\nabla f\| / (\sqrt{B}\, R)$ settles at 0.05–0.16 on the E3 logs, four orders below δ_{spec}. The E3 running minimum slope never leaves $-2/3$ down to $\|\nabla f\| = 1.4\times10^{-5}$ at $L_{\max} = 163$ (1.4×10^{-4} at $L_{\max} = 25$), so the recorded cubic-looking case lies below the conservative $\varepsilon_\star$ and no plateau was observed. A two- and ten-block coupled quartic with known $\left\|\nabla^2 f - \operatorname{blkdiag} \nabla^2 f\right\| = \delta$ recovers δ_{spec} to three digits. ARC-block reaches stationarity there, so $\varepsilon_\star$ limits the $\varepsilon^{-3/2}$ counting argument alone.

Table 21: Tensor-block coupling on ESM image 0 along the $L_{\max} = 25$ gradient-form run. $\beta = \left\|\operatorname{blkdiag} \nabla^2 f\right\|_F^2 / \left\|\nabla^2 f\right\|_F^2$ and $\delta_F = \left\|\nabla^2 f\right\|_F \sqrt{1-\beta}$ from 16 Hutchinson probes. $\delta_{\text{spec}} = \max_b \sigma_{\max}(P_b \nabla^2 f\, P_{\neq b})$. L_H is the max HVP secant. $\hat\delta = \|\nabla f\| / (\sqrt{B}\, R)$ is computed from the same sweep's $(R, \|\nabla f\|)$.

Iterate	β	δ_F	δ_{spec}	L_H	$\|\nabla f\|$	$\hat\delta$
init	0.165	2.11×10^3	135	6.7	5.62	—
sweep 50	0.842	1.13×10^3	617	20	1.16×10^{-3}	0.076
sweep 250	0.785	1.45×10^3	634	37	2.63×10^{-4}	0.130

Gradient-regularized horizon for φ_1. On positive modes the φ_1 multiplier $(1-e^{-h\lambda})/\lambda$ lies within a factor 1.3 of $1/(\lambda + 1/h)$, a Levenberg–Marquardt step with shift $1/h$. Setting $1/h_b = \sqrt{\sigma_b \|g_b\|}$, with σ_b the adaptive Hessian-Lipschitz proxy ($M_b = 2\sigma_b$), places the positive-mode step in the $\sqrt{L\|g\|}$-regularized Newton framework, whose first-order complexity is $\mathcal{O}(\varepsilon^{-3/2})$ (Mishchenko, 2023; Doikov and Nesterov, 2023; Gratton et al., 2023). Negative and flat modes keep the clamped rule of Section 5. We ran this horizon, $h_b = c_h/\sqrt{\sigma_b \|g_b\|}$, against the recorded trust horizon $h_b = c_h/\sigma_b$ on FINER with $L = 25$, $\sigma_0 = 0.3$, under the sweep of Section 6.3, a 120-sweep screen over $c_h \in \{0.3, 1, 3, 10\}$ on image 0 (interior winner $c_h = 3$), geometric midpoints 1.7 and 5.5, and 500-sweep confirmation on images 0 and 7 (Table 22). A capped variant $h_b = \min(c_h/\sigma_b, 1/\sqrt{\sigma_b \|g_b\|})$ was 2.3 dB below the trust horizon at the screen and was not taken further.

The regularized horizon gains 1.4–1.9 dB per sweep and loses 2.7–3.8 dB per gradient equivalent, since it spends 1.8× the oracle calls, almost half of them on rejected trials, because $1/\sqrt{\sigma_b \|g_b\|}$ grows as the gradient shrinks and turns every mode Newton-like faster than the ratio test accepts. Its running minimum gradient exponent against gevals (-0.7 to -1.5) is not steeper than the trust

Table 22: Trust against gradient-regularized horizon for ARC-φ_1, FINER, 500 sweeps, $L = 25$, $\sigma_0 = 0.3$, mean of images 0 and 7. The last column reads each run's best PSNR at the trust run's geval total (1.33×10^5).

Horizon	c_h	PSNR (dB)	gevals	rejected	PSNR at 1.33×10^5 gevals
trust c_h/σ_b	1.0	64.78	1.32×10^5	2%	**64.78**
regularized $c_h/\sqrt{\sigma_b \|g_b\|}$	1.7	66.61	2.37×10^5	48%	61.13
regularized	3.0	**66.69**	2.34×10^5	47%	62.12
regularized	5.5	66.19	2.46×10^5	50%	61.02

horizon's (-0.9 to -1.0), and the winner flips between images ($c_h = 1.7$ on image 0, 3.0 on image 7) and between screen and confirmation, which the sweep rules read as a flat response. A cubic-type rate for the φ_1 step would therefore still be proved by the analysis of regularized Newton methods, and the trust horizon remains the method. The Cauchy type $\mathcal{O}(\varepsilon^{-2})$ bound of Section 5.2 is what the paper claims for it. Appendix T records the same trust rule on ESM image 0 through 2,000 instrumented sweeps.

Why φ_1 leads CubicKrylov on FINER. Section 6.3 records 90.65 dB for trust φ_1 against 72.1 dB for the cubic step at the same $L_{\max} = 163$ and 500 sweeps. Tables 20 and 22 supply the reject rates that comparison omitted, and Table 23 places them on one line. At $L = 25$, trust φ_1 rejects 2% of trials and the cubic gradient form 3.4%. The cubic step form, the rule that has the $\mathcal{O}(\varepsilon^{-3/2})$ bound, rejects 39%. At $L_{\max} = 163$ the cubic gradient form rejects 8.9% (82.65 dB) and the step form 48% (75.89 dB, and 72.1 dB at the gradient form's geval total). Recorded trust φ_1 runs at the same degree on image 0 reject 1–2%.

The exponents do not contradict that ranking. The cubic $\mathcal{O}(\varepsilon^{-3/2})$ bound is for an accepted model minimizer that meets (19). On FINER the three 256×256 weights never meet that test (reading (i) above), so the cubic arm is a truncated Krylov step with adapted $M_b = 2\sigma_b$, not the object of Theorem B.10. The φ_1 bound of Section 5.2 is the Cauchy estimate $\Omega(\|g\|^2 \min(h, 1/\|\mathbf{H}_b\|))$ on a step whose exact quadratic decrease (9) is strictly positive for every sign pattern. That estimate is loose.

The cubic multiplier $1/(\lambda + \sigma\|s\|)$ ($M = 2\sigma$) damps flat modes, where an INR residual is concentrated and no extra regularizer is needed. The φ_1 step is Newton-like on high curvature Ritz values, gradient descent with horizon h on flat ones, and a clamped exponential on $\lambda < 0$. FINER starts with 81% negative spectral mass. After rejects, large M_b makes cubic steps smaller still, while $h_b = c_h/\sigma_b$ still steps on those modes.

Table 22 is the check that this is not a hidden $\varepsilon^{-3/2}$ mechanism. The gradient regularized horizon is the $\sqrt{L_H\|g\|}$-regularized Newton step, whose first-order complexity is $\mathcal{O}(\varepsilon^{-3/2})$, and on FINER it gains 1.4–1.9 dB per sweep and loses 2.7–3.8 dB per geval at 47–50% rejects. The empirically best φ_1 rule is the trust horizon, i.e. the Cauchy class. Worst case stationarity exponents hide constants and are not a PSNR forecast at a fixed sweep budget.

Table 23: FINER reject rates and PSNR for trust φ_1 against CubicKrylov. $L = 25$ φ_1 is the mean of images 0 and 7 from Table 22. Cubic rows and $L = 163$ PSNR are image 0 from Table 20. The 72.1 dB figure is the step form at the gradient form's 2.87×10^5 gevals. Trust φ_1 at $L_{\max} = 163$ is the 500-sweep confirmation of Section 6.3. Its 1–2% reject range is the recorded image-0 runs at the same (c_h, σ_0, L).

Method	$L_{\max}$	PSNR (dB)	gevals	rejected
φ_1 trust	25	64.78	1.32×10^5	2%
CubicKrylov, gradient form	25	60.77	1.03×10^5	3.4%
CubicKrylov, step form	25	60.27	1.42×10^5	38.7%
φ_1 trust	163	90.65	—	1–2%
CubicKrylov, gradient form	163	82.65	2.87×10^5	8.9%
CubicKrylov, step form	163	75.89 (72.1)	3.76×10^5	48.1%

T First-Order Rate of Trust ARC-φ_1 on FINER, Measured Slopes and a Basin Rate

Section 5.2 bounds accepted trust steps by the Cauchy estimate $\Omega(\|g\|^2 \min(h, 1/\|\mathbf{H}_b\|))$, hence $\mathcal{O}(\varepsilon^{-2})$ first-order iterations under a horizon floor (Cartis et al., 2010). Section 6.3 reports PSNR on ESM image 0 without a per-sweep gradient. This appendix logs $\|\nabla f\|$ after every sweep on that image and fits the running minimum envelope $\|\nabla f\|_{\min,k} := \min_{j\le k} \|\nabla f(x_j)\|$. Section T.1 then proves the sharper rate that the measured slopes match once the trust horizons have reached their Newton limit state.

Experiment. The architecture, image, seed, and trust parameters are those of Section 6.3, namely three hidden layers of width 256 (198,915 parameters), ESM image 0, $c_h = 1$, $\sigma_0 = 0.3$, $h_b = c_h/\sigma_b$. Two Lanczos depths, $L_{\max} = 25$ and $L_{\max} = 163$, each run for 2,000 sweeps with a 18,000 s time limit that neither run meets. After each sweep the implementation records $\|\nabla f\|$, f, $\sigma_{\min}$, h_b, accepts, and rejects. Ordinary least squares of $\log \|\nabla f\|_{\min}$ against $\log k$ on a window $[k_0, k_1]$ gives a slope α. Cauchy is $\alpha = -1/2$, cubic regularization is $\alpha = -2/3$, and $\|\nabla f\|_{\min} \asymp k^{-1}$ is $\alpha = -1$. The same fit on $\log f_{\min}$ is reported beside it.

Table 24: Trust ARC-φ_1 on ESM image 0, 2,000 sweeps. Best PSNR for $L_{\max} = 163$ is at sweep 1,994. Rejected is the fraction of block trials. Median trial horizon h_b is 10^8 on both runs (σ_b at $\sigma_{\min} = 10^{-8}$).

$L_{\max}$	PSNR (dB)	sweeps	gevals	rejected	$\|\nabla f\|_{\min}$
25	85.67	2,000	5.27×10^5	1.58%	1.11×10^{-5}
163	132.64	2,000	3.03×10^6	0.70%	1.85×10^{-7}

Table 25: OLS slope of $\log \|\nabla f\|_{\min}$ (and $\log f_{\min}$) against $\log k$ on ESM image 0. The windows are all $k \in [3, 2000]$, early $[101, 700]$, mid $[501, 2000]$, late $[1001, 2000]$. R^2 in parentheses.

$L_{\max}$	series	all	early	mid	late
25	$\|\nabla f\|_{\min}$	−1.28 (0.98)	−1.04 (1.00)	−1.62 (1.00)	−1.80 (1.00)
25	$f_{\min}$	−2.41 (0.98)	−2.06 (1.00)	−3.19 (0.99)	−3.63 (1.00)
163	$\|\nabla f\|_{\min}$	−2.42 (0.97)	−2.05 (0.98)	−3.12 (0.97)	−2.27 (0.90)
163	$f_{\min}$	−5.18 (0.93)	−4.01 (0.98)	−8.24 (0.99)	−8.92 (0.99)

$L_{\max} = 25$. Every window after sweep 100 has slope at most -1.04 ($R^2 \ge 0.996$). Sliding blocks of length 80 pass through -0.89 on $k \in [103, 182]$ and -1.12 on $k \in [403, 482]$, then steepen to -1.80 on the last thousand sweeps. The ratio $\|\nabla f\|^2 / f$ is constant at 0.045 (median and last fifty sweeps), so the loss exponent is twice the gradient exponent to the reported precision. The median h_b is 10^8, so on positive Ritz values the step is already the Newton multiplier $1/\theta$. The $\mathcal{O}(\varepsilon^{-2})$ Cauchy class of Section 5.2 is therefore loose on this image. The measured envelope is $\|\nabla f\|_{\min,k} \lesssim C\, k^{-1}$ with $C \approx 5\times10^{-2}$ on $k \in [125, 500]$, and steeper after that, on one image and one seed.

$L_{\max} = 163$. The same parameters at the confirmed Lanczos depth of Section 6.3 reach 132.64 dB (the 2,824-sweep run of Section 6.3 continues to 133.5 dB without this gradient log). The gradient slope is -2.05 on $[101, 700]$ and -3.12 after sweep 500, faster than k^{-1}. Sliding windows peak near -4.5 around $k = 1{,}100$ and flatten after $k \approx 1{,}600$ as $\|\nabla f\|$ remains at 1.9×10^{-7}. Loss keeps falling (late slope -8.92), so $\|\nabla f\|^2 / f$ rises from 0.058 (median) to 0.69 on the last fifty sweeps, so the floor is in the gradient. Mean realized Lanczos depth is 147.1. Rejects fall from 1.58% at $L_{\max} = 25$ to 0.70%.

What the paper claims. The trust rule remains in the Cauchy class of Section 5.2. Table 25 is the observed envelope on ESM image 0 under that rule. It is consistent with the k^{-1} bound of Theorem T.4 after the iterate has entered the $h_b = c_h/\sigma_{\min}$ state, and with the Q-linear contraction of the same theorem at $L_{\max} = 163$ until $\|\nabla f\|$ meets the logged floor. At the recorded $h_b = 10^8$ the overflow clamp $h\nu \le 60$ already makes every Ritz value $\nu \gtrsim 6\times10^{-7}$ an exact Newton multiplier. The gradient regularized horizon of Table 22 is a different rule and is not used here.

T.1 A basin rate for the trust horizon

The recorded state after sustained acceptance, $h_b = c_h/\sigma_{\min}$ on every block and a reject rate of 1–2%, is a Newton limit. Positive Ritz values receive the multiplier $1/\nu$, and the block step is a Krylov

Newton step on a subspace that contains g_b. Under a local Polyak–Łojasiewicz hypothesis that state gives a $1/\sqrt{V}$ potential, an explicit $\mathcal{O}(\varepsilon^{-1})$ envelope, and a Q-linear contraction of the functional residual. The algebraic steps are the eleven library theorems of the third table in Appendix F.4. Throughout, ℓ is a gradient Lipschitz constant (the library writes it `L`) and $V_k = f(x_k) - f(x^\star)$.

Assumption T.1 (Newton limit basin)**.** There exist $K_0 \in \mathbb{N}$, $\mu \in (0, \ell]$, and a convex neighbourhood U of a local minimizer $x^\star$ such that, for every sweep $k \geq K_0$:

(B1) $x_k \in U$, and on U the loss is ℓ-smooth, $\|\nabla f(x) - \nabla f(y)\| \leq \ell \|x - y\|$, and satisfies $\|\nabla f(x)\|^2 \geq 2\mu\,(f(x) - f(x^\star))$;

(B2) $\sigma_b = \sigma_{\min}$ on every block, hence $h_b = h_\star := c_h/\sigma_{\min}$ with $h_\star \geq 1/\ell$;

(B3) every accepted Krylov spectrum satisfies $\nu_{\min} \geq \mu$ (the negative mode clamp is inactive);

(B4) the first trial on every block is accepted ($\rho_b \geq \eta_1$).

ℓ-smoothness on U also gives $\|\nabla f(x)\|^2 \leq 2\ell\,(f(x) - f(x^\star))$ and $\|\mathbf{H}_b\| \leq \ell$.

(B1) holds under local strong convexity, $\nabla^2 f \succeq \mu \mathbf{I}$ on U. (B2)–(B4) are the recorded trust horizon state after sustained acceptance. On ESM image 0 the median h_b is 10^8 by sweep 500 and the reject rate is 1–2%. The Cauchy bound does not depend on the Lanczos depth L, because $g_b \in \mathcal{K}_1$, so it uses the full block gradient at every $L \geq 1$.

Lemma T.2 (Newton limit model decrease)**.** *Under (B2)–(B3), an accepted φ_1 step on block b satisfies*

$$m_b(0) - m_b(s_b) \;\geq\; \tfrac{1}{4}\,\|g_b\|^2 \min(h_\star, 1/\|\mathbf{H}_b\|) \;\geq\; \tfrac{1}{4\ell}\,\|g_b\|^2 .$$

Proof. Lanczos starts at $q_1 = g_b/\|g_b\|$ and $\mathbf{Q}_L^\top \mathbf{H}_b \mathbf{Q}_L = \mathbf{T}_L$, so the quadratic decrease is the modal sum $\sum_i c_i^2(1 - e^{-2h\nu_i})/(2\nu_i)$ with $\sum_i c_i^2 = \|g_b\|^2$ (equation (9) in Ritz coordinates). For $\nu > 0$, $(1 - e^{-2h\nu})/(2\nu) \geq \frac{1}{4}\min(h, 1/\nu)$. If $h\nu \leq 1$ then $1 - e^{-2h\nu} \geq h\nu/2$ and the left side is at least $h/4$. If $h\nu \geq 1$ then $1 - e^{-2h\nu} \geq 1 - e^{-2}$ and the left side is at least $(1 - e^{-2})/(2\nu) \geq 1/(4\nu)$ (`modal_newton_limit`). Parseval gives the first inequality (`modal_sum_min_h`). (B3) gives $\nu_i \leq \|\mathbf{H}_b\| \leq \ell$, so $\min(h_\star, 1/\nu_i) \geq 1/\ell$ once $h_\star \geq 1/\ell$, which is (B2) (`modal_le_smooth`, `modal_sum_smooth`). □

Lemma T.3 (Sweep decrease in the basin)**.** *Under Assumption T.1, up to the stabilized tolerance,*

$$f(x_k) - f(x_{k+1}) \;\geq\; c_{\mathrm{sw}}\,\|\nabla f(x_k)\|^2 , \qquad c_{\mathrm{sw}} \;=\; \frac{\eta_1}{8\ell\big(1 + B\ell^2/\mu^2\big)} .$$

Proof. (B4) and Lemma T.2 give the per-block decrease $f(x^{(b-1)}) - f(x^{(b)}) \geq \eta_1 \big\|g_b(x^{(b-1)})\big\|^2/(4\ell)$, and the telescope gives $f(x_k) - f(x_{k+1}) \geq (\eta_1/(4\ell)) \sum_b \big\|g_b(x^{(b-1)})\big\|^2$ (`sweep_phi1_telescope`). The Newton limit multiplier is at most $1/\nu \leq 1/\mu$, so $\|s_b\| \leq \mu^{-1}\big\|g_b(x^{(b-1)})\big\|$. ℓ-smoothness relates the start of sweep gradient to the intermediate ones, $\|g_b(x_k)\| \leq \big\|g_b(x^{(b-1)})\big\| + \ell\big\|x_k - x^{(b-1)}\big\|$, and the block steps s_j, $j < b$, have disjoint supports, so $\big\|x_k - x^{(b-1)}\big\|^2 = \sum_{j<b}\|s_j\|^2 \leq \sum_j \|s_j\|^2$. Squaring and summing over the B blocks,

$$\|\nabla f(x_k)\|^2 \;\leq\; 2\sum_b \Big\|g_b(x^{(b-1)})\Big\|^2 + 2\ell^2 B \sum_j \|s_j\|^2 \;\leq\; 2\Big(1 + \frac{B\ell^2}{\mu^2}\Big)\sum_b \Big\|g_b(x^{(b-1)})\Big\|^2$$

(`sweep_triangle_to_hrel`, `sweep_rel_from_steps`). Inverting and substituting the telescope gives c_{sw} (`sweep_invert`). □

Theorem T.4 (φ_1 stationarity in the basin)**.** *Under Assumptions 2.1 and T.1, the ARC-φ_1 iterates with the trust horizon and the ratio test of Algorithm 2 satisfy, for every $k \geq K_0$,*

$$\|\nabla f(x_k)\| \;\leq\; \frac{C_\star}{k - K_0 + k_\star}, \qquad C_\star = \frac{2\sqrt{2\ell}}{q}, \quad k_\star = \frac{2}{q\sqrt{V_{K_0}}}, \quad q = \frac{c_{\mathrm{sw}}(2\mu)^{3/2}}{\sqrt{2\ell V_{K_0}}}, \tag{23}$$

so at most $C_\star/\varepsilon = \mathcal{O}(\varepsilon^{-1})$ sweeps after K_0 reach $\|\nabla f\| \leq \varepsilon$. The same hypotheses give the Q-linear bound $V_{k+1} \leq (1 - 2\mu\, c_{\mathrm{sw}})V_k$, which is sharper and implies (23).

Proof. The potential argument is that of Nesterov and Polyak (2006), Section 6, for gradient dominated functions. Let $G_0 := \sqrt{2\ell V_{K_0}}$. Nested sublevel sets and ℓ-smoothness give $\|\nabla f(x_k)\| \le G_0$ for $k \ge K_0$. Lemma T.3 and (B1) then yield $V_k - V_{k+1} \ge c_{\mathrm{sw}} \|\nabla f(x_k)\|^2 \ge (c_{\mathrm{sw}}/G_0) \|\nabla f(x_k)\|^3 \ge (c_{\mathrm{sw}}/G_0)(2\mu)^{3/2} V_k^{3/2} = q V_k^{3/2}$ (`phi1_cubic_downgrade`). If $V_{k+1} \le V_k - q V_k^{3/2}$ then $1/\sqrt{V_{k+1}} - 1/\sqrt{V_k} \ge q/2$, and summing from K_0, $V_k \le 4/(q(k-K_0) + 2/\sqrt{V_{K_0}})^2$ (`potential_step`, `potential_bound`). The smoothness bound $\|\nabla f\|^2 \le 2\ell V$ turns this into (23) (`phi1_grad_envelope`). The Q-linear claim is the same decrease without the $1/G_0$ step, $V_k - V_{k+1} \ge c_{\mathrm{sw}} \cdot 2\mu V_k$ (`phi1_qlinear`). □

The $1/G_0$ step is the only slack that turns a linear rate into a polynomial one. That is the envelope a log-log fit on a short window reports as slope -1. On ESM image 0 at $L_{\max} = 25$ the measured slope of $\log \|\nabla f\|_{\min,k}$ against $\log k$ is -1.04 on $k \in [101, 700]$ and steepens to -1.80 on the last thousand sweeps (Table 25), which is the Q-linear claim becoming visible on a log-log plot. The global $\mathcal{O}(\varepsilon^{-2})$ class of Section 5.2 is not removed. It still governs the escape to the basin ($k < K_0$) and any iterate that leaves (B2)–(B4). Assumption T.1 itself is not a Lean statement (Appendix F.5).

U Iso-Time and Oracle-Budget Comparisons at 91.4M Parameters ViSIR

U.1 Iso-time results

In the iso-time experiment, every optimizer is granted the same wall clock budget $T = 148$ s (the 100-step time of the CubicKrylov optimizer on the idle GPU). Cheaper first-order and diagonal methods therefore run many more steps. The results are summarized in Table 26 and Figure 3.

First, given equal wall clock time, the CubicKrylov optimizer still performs better than the first-order and the variants with the diagonal Hutchinson approximation by a wide margin. It also leads in the experiment that lets it perform 100 steps and allows all other methods to run for the same wall clock time that those 100 steps took. Every first-order or diagonal method plateaus far below it (Adam and SOAP near 20–21 dB, Muon and AdaHessian lower) despite running thousands of steps in the same amount of time. Second, we note that this is a *single image* reconstruction task, so high PSNR reflects the quality of the one image reconstruction rather than generalization.

Table 26: Iso-time comparison at $T = 148$ s per method (91.4M ViSIR, image #1). "steps" is the number of steps in time T

Method	PSNR (dB)	steps in T
CubicKrylov	**67.28**	100
Cheby-2 n=5	52.46	187
Cheby-1 n=3	33.84	151
Cheby-2 n=3	25.69	201
Cheby-1 n=5	25.19	141
Newton n=3	24.26	243
Newton n=5	22.67	217
Adam	20.71	17,830
SOAP	20.61	3,311
Muon+AdamW	11.43	9,962
AdaHessian	5.89	7,427

U.2 Convergence vs. oracle cost (gradient equivalents)

The comparison metric common in the second-order optimization literature is the number of *oracle calls*, gradient and Hessian(-vector) evaluations, counted here in *gradient equivalents* (gevals, one HVP ≈ one backward pass) (Doikov, 2021; Doikov and Rodomanov, 2023) rather than the number of steps or wall clock time. One function evaluation likewise costs one gradient equivalent. Each block trial costs one extra Hessian–vector product for the model decrease, and rejected trials are included at full cost. The totals are summed over all blocks, Krylov iterations, and sweeps of a run. Note that this metric is hardware-independent and correctly credits methods that reuse curvature information. Table 27 and Figure 4 report PSNR against cumulative gevals.

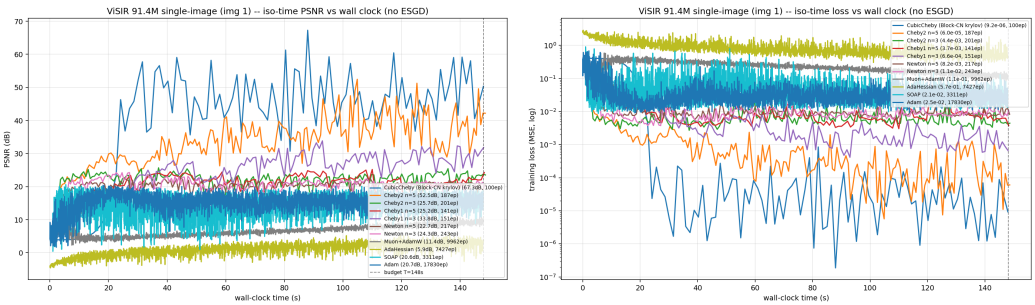

Figure 3: Iso-time training curves (PSNR and MSE loss vs. wall clock time) at the $T = 148$ s budget.

The *per-step* oracle cost is evaluated by this experiment. The matrix-free Krylov step of CubicKrylov costs ≈368 gevals/step (lazy small block Hessians plus a degree-10 Krylov solve on the large tensors), whereas the full Hessian Chebyshev-ON/OFF/Newton steps cost ≈8000 gevals/step because they rebuild the exact small block Hessians ($\sum_{n_b \le 512} n_b \approx 7900$ gevals) *every* step. This is a ∼22× reduction. Consequently, within the 100-step oracle budget of CubicKrylov (36,836 gevals) the full Hessian methods complete only ∼5 steps and reach ≤ 19 dB, while CubicKrylov reaches 67 dB. First-order methods (1 geval/step) plateau near 20 dB. In this experiment the block size independence of the Krylov step (Section 4) is crucial. It reduces the cost of a curvature-aware cubic step from ∼8000 to ∼368 gevals.

Table 27: Best PSNR reached within a matched oracle budget of 36,836 gradient equivalents (CubicKrylov's 100-step cost) on 91.4M ViSIR. ‡See Table 26.

Method	gevals/step	PSNR at ≤36,836 gevals
CubicKrylov	**368**	**67.3**‡
SOAP	1	20.6
Adam	1	20.4
Cheby-ON n=5 (Chebyshev-2)	8049	19.1
Newton n=5 (damped)	8049	13.8
Muon+AdamW	1	11.3
Cheby-1 n=5	8049	7.5
AdaHessian	2	5.8

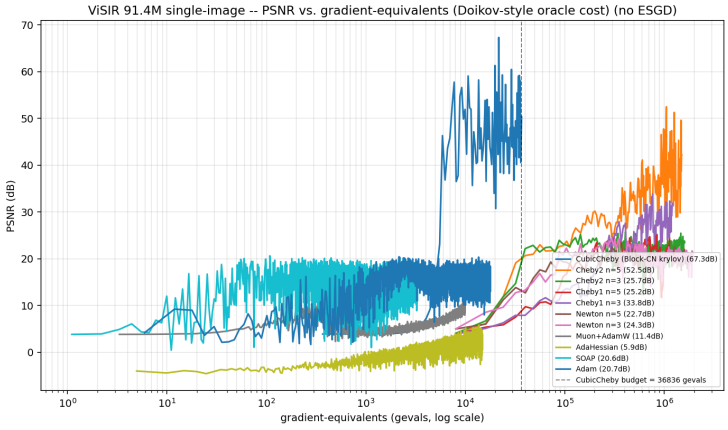

Figure 4: PSNR vs. cumulative gradient equivalents (log scale). At matched oracle budget CubicKrylov (named CubicCheby in the plot) dominates the full Hessian Chebyshev steps by ∼48 dB (they exhaust the budget in ∼5 steps), because its matrix-free Krylov step is ∼22× cheaper per step.

V Multi-Image, Multi-Seed ViSIR-Nano Benchmark (ESM-20)

To rule out a single image artifact, the nano configurations of Appendix H were rerun in the style of the ESM-20 benchmark of Appendix W, the first 20 ESM images with three random seeds each. Two

reference methods are added at this scale, first kind spectral preconditioned gradient descent with τ=10 (Doikov et al., 2024), and a second implementation of DSO-BlockHess that eigendecomposes every per-layer Hessian and preconditions with absolute eigenvalues, for ten methods and 600 runs total. The budget rule differs from the 91.4M ESM-20 experiment, which grants every optimizer the same 150 optimization steps. Here, for each (image, seed) pair, CubicKrylov runs its 100-step ViSIR-Nano experiment first and every reference method is then granted *twice* the gradient equivalents CubicKrylov spent on that pair. A fixed step count would be meaningless across this set, since a single dense Hessian build costs 15.2k gradient equivalents, and the doubled oracle budget rules out exhaustion of that budget as the explanation for the ranking.

Table 28: ESM-20-style nano benchmark (20 images $\times$ 3 seeds, ViSIR-Nano, 15,235 parameters). Best PSNR per run, aggregated over the 60 runs per method, with mean wall clock and total gradient equivalents per run. The right part is the paired per-image comparison against CubicKrylov (mean over seeds), with the mean PSNR advantage of CubicKrylov, its per-image win rate, and the Wilcoxon signed-rank p-value (N=20 images).

	aggregate over 60 runs				paired vs. CubicKrylov		
Method	PSNR (dB)	median	wall (s)	gevals	ΔPSNR	win%	p_W
CubicKrylov (nano)	$\mathbf{32.8 \pm 1.4}$	**32.9**	21	1.8×10^4	—	—	—
Reg. Newton lazy, m=n (Doikov et al., 2023)	32.4 ± 1.2	32.5	298	3.0×10^4	+0.3	85%	3.9×10^{-4}
CN-Lazy, m=n (Doikov et al., 2023)	32.1 ± 1.1	32.1	305	3.0×10^4	+0.7	95%	3.8×10^{-6}
Spectral-precond. GD τ=10 (Doikov et al., 2024)	27.7 ± 3.3	28.5	116	3.6×10^4	+5.1	100%	1.9×10^{-6}
SSCN τ=64 (Zhao et al., 2025)	24.4 ± 1.2	24.4	126	3.6×10^4	+8.4	100%	1.9×10^{-6}
FD-CNM, m=50 (Doikov and Grapiglia, 2023)	14.8 ± 4.2	14.6	124	4.6×10^4	+17.9	100%	1.9×10^{-6}
DSO-BlockHess (per-layer full Hessian)	9.0 ± 0.7	9.0	9	4.6×10^4	+23.8	100%	1.9×10^{-6}
DSO-BlockHess (abs-eig, rebuild every step)	8.2 ± 0.8	8.2	5	4.6×10^4	+24.6	100%	1.9×10^{-6}
Adaptive cubic Newton, m=1 (Doikov et al., 2023)	7.9 ± 1.3	8.1	182	4.6×10^4	+24.9	100%	1.9×10^{-6}
Cubic Newton (fresh Hessian) (Nesterov and Polyak, 2006)	3.9 ± 0.9	4.4	183	4.6×10^4	+28.9	100%	1.9×10^{-6}

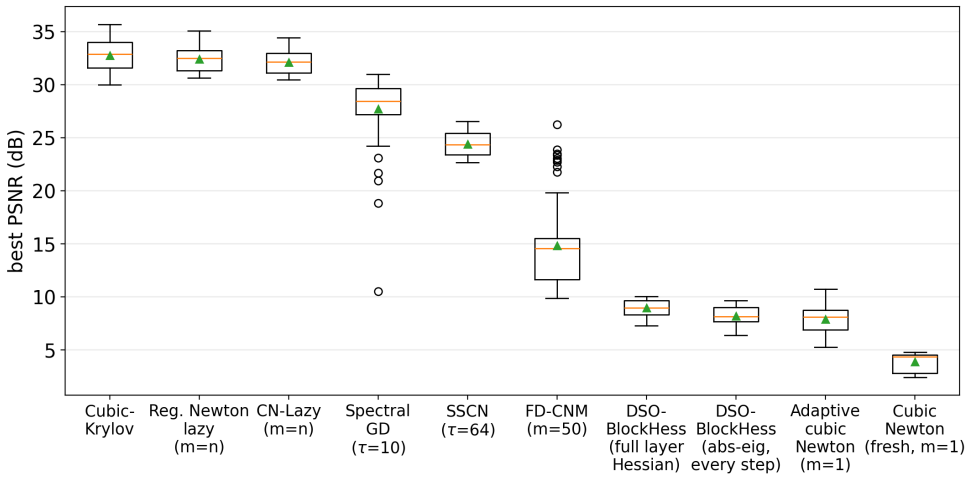

Figure 5: Best-PSNR distributions over the 60 runs per method (20 ESM images $\times$ 3 seeds, ViSIR-Nano). Every reference method is granted twice the CubicKrylov oracle budget of the same (image, seed) pair. Boxes show quartiles and triangles the means.

The single image ranking is stable across images and seeds (Table 28, Figure 5). CubicKrylov reaches 32.8 ± 1.4 dB and beats gradient regularized Newton on 17 of 20 images (+0.3 dB mean, Wilcoxon $p = 3.9 \times 10^{-4}$) and lazy cubic Newton on 19 of 20 (+0.7 dB, $p = 3.8 \times 10^{-6}$), while consuming 59% of their oracle budget and $\sim$14$\times$ less wall clock. The margins differ from those of the 91.4M ESM-20 benchmark. At nano scale, where the dense 15k $\times$ 15k Hessian is affordable, the lazy m=n variants are genuine competitors whose distributions overlap CubicKrylov's, and the advantage of the matrix-free block step is its cost. The quality of the steps is on par. First kind spectral preconditioned gradient descent (Doikov et al., 2024) falls between the lazy variants and SSCN at 27.7 ± 3.3 dB. Its rank-10 preconditioner is cheap to maintain, but ten eigendirections cannot equalize the contraction rates of a SIREN spectrum. Every other configuration loses on all 20 images at the smallest attainable p-value, for the same reasons as in the single image reference block. Frequent dense rebuilds exhaust the budget within a handful of steps, and the τ=64 subspace of SSCN cannot match full space curvature. The small DSO-BlockHess wall clock entries deserve a caveat. Per-layer rebuilds are

vectorized, hence cheap in wall clock and expensive in oracle calls (one rebuild of all layers costs n gradient equivalents), so the budget is spent within $\sim$21 steps in $\sim$9 s for the rebuild-every-10 variant and within 3 steps for the abs-eig variant that rebuilds every step. For the plain-Newton variant the cause is not an exhausted oracle budget, since an equal-time comparison on image #0 (300 s, 670 steps, $\approx 10^6$ gradient equivalents) still plateaus at 9.2 dB. The abs-eig variant, by contrast, recovers to 33.1 dB when the budget rule is lifted. Appendix X reports an equal-budget comparison that separates the rebuild cost from genuine step quality for the whole reference panel.

W Wall-Clock Statistics of the Multi-Image Nano Benchmark

The wall clock column of Table 28 reports the *mean* per-run time over the 60 runs of each method (20 images $\times$ 3 seeds). Table 29 reports the spread behind those means.

Table 29: Per-run wall clock statistics of the multi-image ViSIR-Nano benchmark (Table 28), with mean, standard deviation, extremes, and coefficient of variation over the 60 runs per method.

Method	mean (s)	std (s)	min (s)	max (s)	CV
CubicKrylov	20.5	8.7	14.5	60.4	43%
Reg. Newton lazy, $m{=}n$	298.3	35.6	274.9	424.3	12%
CN-Lazy, $m{=}n$	304.9	27.6	284.8	390.1	9%
Spectral-precond. GD $\tau{=}10$	115.7	5.9	107.4	136.8	5%
SSCN $\tau{=}64$	125.7	7.1	114.1	144.8	6%
FD-CNM, $m{=}50$	123.9	22.5	106.3	232.5	18%
DSO-BlockHess	8.7	2.6	6.0	16.7	30%
DSO-BlockHess (abs-eig)	4.8	0.1	4.8	5.5	2%
Adaptive cubic Newton, $m{=}1$	182.0	3.6	176.7	195.8	2%
Cubic Newton (fresh Hessian)	182.5	5.1	175.7	198.4	3%

Methods with fixed size dense linear solves are nearly independent of the target image, because the cost of building a dense Hessian for a $15\text{k} \times 15\text{k}$ parameter block and of its eigendecomposition depends only on the block size, not on the image being fit, so the fresh Hessian cubic Newton variants, the spectral preconditioner, and SSCN vary by only 2–6%. The lazy $m{=}n$ methods vary moderately (coefficient of variation (CV) $\sim$9–12%), because their time is dominated by one dense build plus 15,000 cheap iterations whose timing fluctuates from run to run, apparently with GPU load. CubicKrylov shows the largest *relative* spread (CV 43%, per-image means from 14.9 to 31.1 s, worst single run 60.4 s) precisely because its work does depend on the target image. Harder images cause more rejected steps, each rejection forces a fresh lazy Hessian rebuild in phase A, and phase B spends additional adaptive M_b retries. The two DSO-BlockHess variants run at a small absolute time scale (5–17 s) because of the early oracle budget exhaustion discussed in Appendix H. At these short times the abs-eig variant, the implementation that eigendecomposes every per-layer Hessian and preconditions with the absolute values of the eigenvalues (Appendix H), is nearly deterministic (CV 2%), while the plain Newton variant varies more (CV 30%).

The comparison in Table 28 corresponds to these results. The *worst* CubicKrylov run (60.4 s) is still $\sim 4.5\times$ faster than the *fastest* run of either lazy cubic Newton reference with $m{=}n$ (274.9 s). It has the $\sim 14\times$ mean wall clock advantage.

X EQUAL BUDGET COMPARISON AT NANO SCALE (15K PARAMETERS VISIR REPLICA), HESSIAN RECOMPUTATION COST VERSUS STEP QUALITY

The oracle budget rule of the nano benchmark (twice the gradient equivalents CubicKrylov spends, Appendix H) affects the results of the methods with a dense per-layer Hessian computation very much. The methods that compute Hessian frequently are stopped after a handful of steps. This raises a fairness question. Does the low ranking of the DSO-BlockHess variants (and of cubic Newton with Hessian recomputation every step) reflect poor *steps* or merely the *rebuild cost*? On other benchmarks the same second kind family is strong. The DSO-Adaptive variant that recomputes the full block Hessian only every K steps, EMA-smooths it (β=0.9), runs a Hutchinson-diagonal Chebyshev step on well-conditioned blocks, and adds an element-wise negative curvature escape performs better than every Chebyshev *first*-kind run by two to three orders of magnitude in MSE on a saddle rich 2-D regression benchmark, a ranking that was re-verified for this appendix with a fresh three-seed rerun accurately reproducing the results achieved earlier on this same benchmark. The experiment below checks whether that ordering is preserved if the methods are run until convergence.

Experimental design. All runs use the single image nano mode experimental design (ESM image #0, seed 42, 15,235 parameters). Gradient equivalents for DSO-Adaptive are computed conservatively at the harness level, without modifying the optimizer, as one per closure call (including line-search evaluations), one per Hutchinson Hessian–vector product (3 per block per step), and the block size per full block Hessian computation. Three stages: (i) DSO-Adaptive inside the benchmark budget of 2×17.7k gradient equivalents, with Hessian re-computation frequency of $K \in \{5, 10, 20\}$; (ii) DSO-Adaptive unconstrained, to locate its convergence plateau; (iii) every optimizer of the nano benchmark, plus the global-Krylov and per-block ARC implementations of Appendix M, granted that plateau budget, which is equal to 6.53×10^6 gradient equivalents ($\sim$185$\times$ the benchmark rule). Additionally, a 2,400 s wall clock limit was used to have a reasonable termination condition for this case.

Stage (i), inside the benchmark budget. DSO-Adaptive does not have sufficient budget to reach its reasonable performance level exactly like the other methods that recompute Hessian frequently. K=5 gives 11 steps (12.6 dB), K=10 21 steps (16.6 dB), K=20 41 steps (19.9 dB). Doubling the Hessian recomputation frequency doubles the usable steps and adds $\sim$3.5 dB, but even K=20 remains far below the first kind spectral preconditioner (28 dB), because the curvature based mode dispatcher of DSO-Adaptive routes $\sim$96% of the parameters through full block Hessian mode (instead of diagonal Hutchinson mode) of DSO-Adaptive and the method keeps expending its geval budget at a great rate.

Stage (ii), the plateau. Unconstrained, DSO-Adaptive (K=5) converges to 30.0 dB in $\sim$2,400 s and $\sim$3,000 steps, consuming 6.53×10^6 gradient equivalents (8,867 block rebuilds, with ten steps skipped where `eigvalsh` failed to converge on a near-singular EMA Hessian).

Table 30: Equal-budget comparison, every method granted DSO-Adaptive's plateau budget of 6.53×10^6 gradient equivalents ($\sim$185$\times$ the benchmark rule) and a 2,400 s wall clock limit, on ESM image #0, seed 42. "Limited by" is the stopping constraint.

Method	best PSNR (dB)	steps	gevals used	limited by
ARC, global Krylov (Cartis et al., 2011a;b)	**37.21**	16,391	4.6×10^5	wall clock
CubicKrylov (nano)	36.64	7,334	5.8×10^6	wall clock
ARC, per-block σ_b, reduced-operator acceptance (Doikov and Nesterov, 2025)	36.49	2,790 sweeps	1.4×10^6	wall clock
ARC, per-block σ_b (Appendix M)	36.37	1,795 sweeps	1.2×10^6	wall clock
Reg. Newton lazy, m=n (Doikov et al., 2023)	36.36	120,000	2.4×10^5	wall clock
CN-Lazy, m=n (Doikov et al., 2023)	36.18	115,000	2.4×10^5	wall clock
CubicCheby-DSO v2r, grad-triggered refresh (Appendix N)	35.46	8,113	6.5×10^6	gevals
CubicCheby-DSO v3, α-floor (Appendix N)	35.34	7,889	6.5×10^6	gevals
CubicCheby-DSO v2, EMA-cached bounds (Appendix N)	35.18	7,997	6.5×10^6	gevals
CubicCheby-DSO grS, gradient regularized shift (Appendix N)	34.93	8,018	6.5×10^6	gevals
CubicCheby-DSO v1, fresh bounds (Appendix N)	34.92	5,350	5.0×10^6	wall clock
FD-CNM, m=50 (Doikov and Grapiglia, 2023)	34.70	893	5.4×10^5	own convergence
CubicCheby-DSO v2, fresh bounds (Appendix N)	34.20	6,358	5.6×10^6	wall clock
CubicCheby-DSO v1, EMA-cached bounds (Appendix N)	33.55	5,919	5.3×10^6	wall clock
DSO-BlockHess (abs-eig, every step)	33.07	429	6.5×10^6	gevals
DSO-Adaptive, K=5 (reference)	29.95	2,496	6.5×10^6	own plateau
Adaptive cubic Newton, m=1 (Doikov et al., 2023)	29.53	80	6.1×10^5	wall clock
SSCN τ=64 (Zhao et al., 2025)	28.87	10,000	6.5×10^5	step limit
Spectral-precond. GD τ=10 (Doikov et al., 2024)	28.00	32,000	7.7×10^5	saturated
DSO-BlockHess (plain Newton, rebuild every 10)	8.20	4,281	6.5×10^6	algorithmic
Cubic Newton (fresh Hessian) (Nesterov and Polyak, 2006)	3.42	9	1.4×10^5	`eigh` failure

Stage (iii), interpretation. Table 30 distinguishes three failure modes that are not clearly separated in the benchmark with the fixed oracle budget, Table 28.

Rebuild cost. The abs-eig DSO-BlockHess (exact block Hessian with absolute eigenvalues), which scores 8.2 dB under the benchmark rule (three affordable steps within the geval budget), reaches **33.1 dB**, it needed greater budget to give a good result. The same holds in attenuated form for DSO-Adaptive ($12.6 \rightarrow 30.0$ dB). The second kind DSO family therefore does beat first kind spectral preconditioning at nano scale too, by 2–5 dB, consistent with its results on the saddle rich regression benchmark and on the 91.4M parameter fidelity study.

Algorithmic failure. The plain Newton DSO-BlockHess variant stays at 8.2 dB after 6.5×10^6 gradient equivalents because its unregularized Newton step on severely indefinite layer Hessians is attracted to saddles (since the eigenvalues are absolute, the saddle escape mechanism due to the Chebyshev second kind preconditioning does not function), the failure already diagnosed in Appendix H. Increasing the budget does not help it because it cannot escape saddles. Plain cubic Newton (with exact Hessian recomputations every step) crashes at step 9 when its own trajectory drives the dense Hessian into a state whose eigendecomposition fails to converge in LAPACK.

CubicKrylov performance. Among the methods evaluated on these benchmarks, each under a fixed gradient-equivalent budget, CubicKrylov leads at *both* budget levels, 31.66 dB at 17.7k gradient equivalents and 36.64 dB at 6.5×10^6. The lazy cubic Newton variants with $m{=}n$ are close to within 0.3–0.5 dB here, but only by spending the full 2,400 s runtime budget on 10^5 iterations, and the strongest DSO variant with exact Hessian recomputation remains 3.6 dB behind at $150\times$ the spent oracle cost.

ARC-global and the scale boundary. The one method that edges past CubicKrylov in this comparison is global-Krylov ARC, at **37.21 dB** after 16,391 accepted steps (21 rejections, and σ settles at its 10^{-8} floor) at only 4.6×10^5 gradient equivalents. At $n = 15$k the loss landscape is still benign enough that a *single* global regularization weight is adequate. Nearly every search step is accepted, the multiplicative update drives σ down to its lower clamp $\sigma_{\min} = 10^{-8}$, the cubic term becomes negligible, the whole-space Lanczos step is nearly an unregularized Newton step, and the method converges rapidly. The per-block ARC variants of Appendix M reach the same PSNR range (36.37 dB plain, 36.44 dB stabilized, 36.49 dB with the guarded gradient-based acceptance, and every σ_b ends at the same clamp), so at this scale it does not matter whether the regularization weight is global or per-block, and the block bookkeeping only adds sweep overhead. This makes the full-scale outcome the more instructive one. The identical global implementation stalls at 4.0 dB on the 91.4M parameter model (Appendix M), where a single σ must satisfy the acceptance test along all 91.4M directions at once, so rejections escalate it to $\sim10^{10}$ and the accepted steps shrink to noise level, while the per-block variant reaches 61.9 dB there. The nano comparison and the full-scale runs therefore lie on opposite sides of the model size at which the global-σ formulation stops scaling and per-block regularization constants become the enabling mechanism.

Among the methods of the benchmark with the fixed oracle budget, Table 28, the matrix-free CubicKrylov step leads at both budget levels tested, 31.66 dB at 17.7k gradient equivalents and 36.64 dB at 6.5×10^6, and the only entry of Table 30 ahead of it in this comparison, global-Krylov ARC at 37.21 dB, leads by less than 0.6 dB. The methods of Table 30 that have also been run unchanged at the 91.4M parameter scale, CubicKrylov, the per-block ARC variants of Appendix M, and the refined recurrence hybrids of Appendix N, all remain effective at that scale through the same mechanism of per-block regularization (Appendix M and the ESM-20 runs of Appendix N). These results are consistent with the nano ablation of Appendix H, which evaluates the same methods without the equal oracle budget imposed in this comparison.