NON-CONVEX TRAINING USING IMAGINARY PATHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most machine learning methods including deep learning are defined by non-convex optimization problems that are typically regularized for enhanced generalization yet fundamentally challenged by loss landscapes with local minima. We show that regularization by the ℓ_1 and ℓ_2 norms can be expressed in terms of a radial-phase decomposition for which we prove the existence of a complex path with a bounded loss leaving any nonzero real stationary point under the stated assumptions. We explore this by designing a novel complex lifting framework from $\mathbb{R}^P$ to $\mathbb{C}^P$ that introduces a bilinear skew-symmetric auxiliary loss. We impose a consensus constraint between the lifted and original parameterization and use the generic alternating direction methods of multipliers (ADMM) for the optimization defining the proposed Complex-Continuation ADMM (CC-ADMM) method for non-convex optimization. After finite time complex exploration, CC-ADMM projects the model onto the real parameter space and continues real-valued optimization, reducing its convergence analysis to that of the real optimization phase. We show that, under the stated assumptions, the CC-ADMM converges to a stationary point of the original loss with probability one. CC-ADMM uses only twice the number of parameters during training with no resulting inference overhead. Experiments on synthetic non-convex problems confirm the predicted exploration behavior and when applied on a variety of non-convex machine learning tasks including "grokking", as well as Vision Transformers and Llama-1B and 3B finetuning, we observe enhanced generalization performance. The CC-ADMM thereby provides a versatile and favorable framework for non-convex training.

1 INTRODUCTION

Non-convex optimization problems are ubiquitous in modern engineering, machine learning, and signal processing, yet their landscapes can be difficult to navigate because of local minima, saddle points, and flat plateaus. The dynamics of gradient methods near saddle points have been studied extensively (Lee et al., 2016; Du et al., 2017; Jin et al., 2017). Such geometric obstacles are a fundamental barrier to reliable optimization in high dimensions and motivate methods that explore additional directions.

In the context of deep learning, a substantial body of literature has shown that over-parameterization can produce benign landscapes under specified architecture and width assumptions: connected sub-level sets, absence of spurious valleys, and no-spurious-minima results for quadratic activations (Kawaguchi, 2016; Nguyen & Hein, 2017; Nguyen, 2019; Venturi et al., 2019; Du & Lee, 2018). Other results have established the convergence of gradient-based training for specific architectures, such as sufficiently wide networks (Allen-Zhu et al., 2019; Du et al., 2019). A complementary line of *mode connectivity* work has shown empirically and theoretically that separately trained solutions are often linked by low-loss curves in parameter space (Garipov et al., 2018; Draxler et al., 2018; Kuditipudi et al., 2019; Benton et al., 2021; Ainsworth et al., 2023). Empirical landscape studies have further characterized the geometry of trained minima (He et al., 2019; Li et al., 2018). However, this picture is far from complete and at best presents that optima are connected by low loss surfaces, but it is unclear how gradient based methods explore them. Finite (training) time stochastic (Random walk) arguments in high dimension do not suffice to show that all minima are visited whereas noisy gradient driven random walks are not symmetric and are influenced by the geometry of the landscape.

Typically, deep learning models are regularized by the ℓ_1 and ℓ_2 norms. We presently explore that such regularization can be expressed in terms of a radial-phase decomposition for which we prove

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

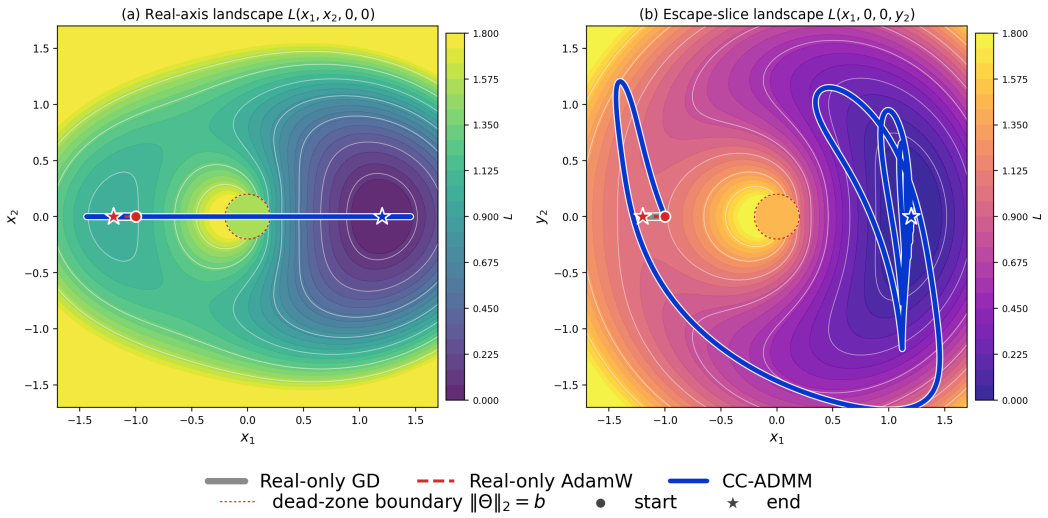

Figure 1: This example uses two real parameters $x = (x_1, x_2)$ and a modReLU-style radial activation applied to the whole parameter vector defined as $a(x) = \max\{r - b, 0\}\,\frac{x}{r}$ for $r = \|x\|_2 > 0$, with $b = 0.2$ and $a(0) = 0$. This activation outputs zero for $r \le b$, defining its *dead zone* (dashed circles). Writing the activation output as $a = a(x) = (a_1, a_2) \in \mathbb{R}^2$, the real loss defining this example's landscape is $\mathcal{L}_{\text{real}}(x) = (\|a\|_2 - 1)^2 + \frac{1}{2}\,(1 - a_1/\|a\|_2) + \frac{1}{2}x_2^2$, where $a_1/\|a\|_2$ is defined as zero when $a = 0$. These formulas omit the 10^{-8} numerical stabilizers used in the implementation. For training, we lift the parameters to $\Theta = x + iy \in \mathbb{C}^2$, where $x = \Re\Theta = (x_1, x_2)$ is a real part of Θ and $y = \Im\Theta = (y_1, y_2)$ denotes its imaginary part. We apply the same radial activation to Θ using $r = \|\Theta\|_2$, and replace a_1 by its real part $\Re a_1$ in the loss. To induce exploration, we add the proposed bilinear auxiliary $\lambda_k \mathcal{L}_M = 2\lambda_k(x_1 y_2 - x_2 y_1)$ and consensus penalty $\frac{\rho_k}{2}\|\Theta - Z + U\|_2^2$, where $Z \in \mathbb{R}^2$ is a separate real consensus variable and $U \in \mathbb{C}^2$ is the scaled dual variable. Consensus requires $x = Z$ and $y = 0$, equivalently $\Theta = Z$ when Z. During exploration, intermediate iterates may violate this constraint and have nonzero imaginary coordinates. Contours show two-dimensional slices of the task loss, excluding auxiliary and consensus penalty parts: (a) the real loss $\mathcal{L}_{\text{real}}(x_1, x_2)$, with $y_1 = y_2 = 0$, and (b) its complex extension on the (x_1, y_2) slice, with $x_2 = y_1 = 0$. All methods start from the real point $x = (-1, 0)$ and take 2000 steps. GD (grey) and AdamW (red) approach the real local minimum near $(-1.2, 0)$ with loss 1. Our CC-ADMM (blue) explores the imaginary y_2 direction and converges to the real consensus near the global minimum $(1.2, 0)$ as λ_k is annealed away. The limiting solution is real and is a stationary point of $\mathcal{L}_{\text{real}}$. Notably, in the real task domain of the panel (a), ADMM-CC (blue) path "tunnels" through the high loss region around the dead zone towards lower loss minimum. Circles and stars mark starts and endpoints, respectively. Further details accompany Figure 3 in the appendix.

the existence of a complex path with a bounded loss leaving any real stationary point. We note that such radial-phase decomposition is also relevant beyond deep learning: any nonconvex problem with a regularization based on a magnitude-driven loss component as commonly used also in signal processing, phase retrieval, inverse problems, and optimal control fit the current framework.

Having established the existence of such path, we propose a novel optimization framework that augments the optimization into the complex plane ($\mathbb{C}^n$) to bypass the topological barriers of learning in the real domain. Notably, we ensure that the original optimization is recovered when projected back to the real domain ($\mathbb{R}^n$) by imposing a consensus constraint in the complex optimization. Importantly, this constraint can be handled naturally using the Alternating Direction Method of Multipliers (ADMM), which has become a foundational tool to solve large-scale, structurally complex optimization problems (Boyd et al., 2011). ADMM has also been used for neural-network training (Taylor et al., 2016; Wang et al., 2019a) and weight pruning (Zhang et al., 2018). We consequently develop the Complex-Continuation ADMM (CC-ADMM) for non-convex training exploring imaginary paths for improved optimization.

Our approach is motivated by the geometric phenomenon of lifting: expanding the dimensionality or algebraic structure of the parameter space can structurally transform inescapable local minima into strict saddle points admitting directions of negative curvature (Ge et al., 2017; Ma et al., 2023). By leveraging the complex field, our framework constructs such escape directions deterministically, without relying on stochastic noise, approximations, or explicit objective smoothing. Crucially, this complex augmentation scales linearly in computational cost ($\mathcal{O}(2n)$), efficiently mitigating the issue of local minima.

Figure 1 illustrates the construction in its simplest non-trivial form. Two real minima sit on either side of a high-loss barrier. A real-domain only optimizer started in the suboptimal basin is trapped. The same iterate lifted to take complex values, rounds the barrier through an imaginary detour and converges to the better minima. Inference is unchanged: the complex excursion is a training-time tool that vanishes by the end of training, leaving a purely real-domain deployed model.

Notably, the following three properties make our construction broadly useful. It is *deterministic* - every step is a closed-form (or near-closed-form) update from the previous iterate, with no injected stochasticity. It is *cheap* - the complex extension doubles the parameter count during training only and incurs no inference overhead. It is *architecturally light*: the radial loss structure it exploits is already present in any modReLU-style activations (Arjovsky et al., 2016), while linear-layer biases and the affine parameters of batch and layer normalization provide learnable additive shifts (Ioffe & Szegedy, 2015; Ba et al., 2016).

In summary, our contributions are:

- We prove the existence of complex paths with explicit loss bounds from real strict local minima in the radial-phase settings, including escape paths beyond any bounded basin of attraction (Theorem 1 and Corollary 1.1).
- We introduce the CC-ADMM, a complex-primal/real-consensus ADMM based optimization method with a skew bilinear phase-invariant auxiliary loss, $\mathcal{L}_M$, that allows to navigate non-convex optimization landscape without getting stuck in local optima. It uses purely deterministic mechanisms (no injected parameter noise), and once annealed reverts at inference-time to the purely real objectives and parameters.
- We derive deterministic departure from a real stationary point under a nondegenerate coupling (Theorem 2), and convergence to a stationary point with probability one for SGD after projection to reals, under the stated assumptions (Theorem 3). Notably, these theorems describe *algorithmic dynamics*, not global optimality of the real problem.

We validate the escape mechanism on scalar and vector toy problems, provide image-classification benchmarks on MNIST(Lecun et al., 1998), CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and grokking experiments on algebraic datasets of increasing difficulty (Power et al., 2022), as well as experiments on Llamma 1B and 3B parameter LLMs finetuning considering Alpaca and WikiText datasets, (Taori et al., 2023; Merity et al., 2016). We compare CC-ADMM against their respective real-valued baselines. Code for reproducing all experiments is provided as supplementary material.

2 Related Work

Nonconvex ADMM and continuation. ADMM divides a constrained optimization problem into smaller updates, often called "blocks", using a penalty and dual variables to coordinate the updates. For nonconvex and nonsmooth problems with linear constraints, Wang et al. (2019b) establish conditions for convergence to stationary solutions. Later variants make the updates easier to compute or accelerate their progress. Hien et al. (2022) simplify each block update by minimizing a function that upper-bounds its objective, and use previous iterates to extrapolate the updates, with convergence results under stated assumptions. For rank-constrained model identification, Liu et al. (2023) adapt the penalty and use Anderson acceleration to combine recent updates, demonstrating faster convergence numerically. Yuan (2025) develops an ADMM continuation scheme that gradually increases the constraint penalty while reducing the smoothing parameter of a Moreau-envelope approximation to a nonsmooth objective term. These works address convergence to stationary solutions and the efficiency of the updates. Our focus is on how temporary additional parameter directions can enable exploration beyond a real local minimum.

Denoisers and learned priors in ADMM. Plug-and-play (PnP) methods reconstruct signals by alternating between fitting measurements and applying a denoiser. Venkatakrishnan et al. (2013) introduce this approach within ADMM, allowing a denoiser to supply prior information without an explicit regularization formula. Ryu et al. (2019) prove that PnP-ADMM converges to a fixed point when the data-fidelity term is strongly convex, the denoiser satisfies a suitable Lipschitz bound on its residual mapping, and the algorithm's step size satisfies their stated conditions. Park et al. (2025) show how pretrained score diffusion models can serve as PnP denoising priors without retraining. Shrestha & Fu (2026) integrate score-based denoising into ADMM using Gaussian noise injection and conditional Langevin steps to correct the denoiser inputs. These methods use signal priors to solve inverse problems. CC-ADMM uses an auxiliary loss $\mathcal{L}_M$ acting directly on model parameters to promote exploration during training.

Lifting and optimization landscapes. For matrix sensing, matrix completion, and robust PCA, Ge et al. (2017) identify conditions under which every local minimum of the studied formulations is globally optimal. Ma et al. (2023) study over-parameterization through tensor lifting. For rank-one matrix sensing, they show that, under their stated assumptions and for a sufficiently large odd lifting order, the lifted counterpart of a non-global local minimizer becomes a strict saddle. This provides a direction of negative curvature along which the lifted objective can decrease. These approaches provide concrete example of how adding variables can change geometry around a local minimum.

Complex variables and training-time exploration. Complex continuation has also been used to solve nonlinear equations. Jalali et al. (2008) connect real and complex solution branches to locate multiple solutions of equilibrium and stability equations. Jiang et al. (2014) apply ADMM to nonconvex polynomial optimization with real or complex variables, exploiting the polynomial's multi-block structure of their tensor form. In neural networks, Arjovsky et al. (2016) introduce unitary recurrent networks and modReLU, which adjusts each complex activation's magnitude while preserving its phase whenever the output is nonzero.

CC-ADMM uses imaginary parameter coordinates temporarily during training. A bilinear auxiliary loss $\mathcal{L}_M$ couples real and imaginary parameters to promote exploration, while the consensus constraint requires agreement with a real parameter vector. The auxiliary loss $\mathcal{L}_M$ is annealed away, the parameters are projected onto the real space, and optimization continues on the original real loss. Appendix B discusses further connections to complex-valued networks, mode connectivity, continuation, and symmetry-based optimization.

3 Problem setting

We minimize a real, regularized non-convex loss function $\mathcal{L}_{\text{real}} : \mathbb{R}^P \to \mathbb{R}$ over the real parameters $Z \in \mathbb{R}^P$ and consider throughout the multivariate setting $P \geq 2$. We assume that the loss admits a *radial-phase decomposition* on $\mathbb{R}^P$,

$$\mathcal{L}_{\text{real}}(Z) \;=\; \psi(\|Z\|_2) \;+\; \chi(Z), \tag{1}$$

with $\psi \in C^1([0,\infty))$, $\chi \in C^1(\mathbb{R}^P)$, and $\|\nabla\chi(Z)\|_2 \leq \varepsilon$ on a ball of radius $R_0 > 0$ around the origin, where $\varepsilon \geq 0$. We assume $\psi'(0) = 0$ so that the radial function $Z \mapsto \psi(\|Z\|_2)$ and its complex extension are continuously differentiable at the origin. The decomposition covers a smooth ℓ_2-regularized loss by setting $\chi(Z) := \mathcal{L}(f_Z)$, where f_Z is a model with real parameters Z and $\mathcal{L}$ is an output loss, e.g., cross entropy or squared loss, together with regularization $\psi(Z) := \mu\|Z\|_2^2/2$ with regularization coefficient $\mu \geq 0$. It also covers ℓ_1 regularization through restating it as a ℓ_2-problem by the use of the identity $\|z\|_1 = \min_{a\odot b=z}(\|a\|_2^2 + \|b\|_2^2)/2$ (Hoff, 2017; Ziyin & Wang, 2023; Jacobs & Burkholz, 2025), where $\odot$ denotes coordinate-wise multiplication. Let's also note that despite ReLU networks generally do not satisfy the C^1 assumption, they can be seen as a limit case of C^1 smooth *softplus* networks of the same width and depth (Zhang et al., 2024).

4 The Complex-Continuation ADMM (CC-ADMM)

The following theory establishes three statements under the stated assumptions: i) A bounded-loss complex path connects a nonzero real strict local minimum to its antipode (Theorem 1) and as an

important consequence there exists an escape path from any bounded loss basin around the local minima (Corollary 1.1). ii) By introducing an auxiliary loss $\mathcal{L}_M$ relying on any skew-symmetric matrix M s.t. $MZ_\star \neq 0$, the CC-ADMM primal step moves deterministically from a real local minimum $Z_\star$ into the complex domain(Theorem 2). iii) CC-ADMM based on SGD converges to a stationary point of the original loss after annealing and projection onto the reals (Theorem 3). Full proofs and more involved generalizations are available in Appendix C.

Complex extension of the real loss. We lift the parameter to $\Theta = x + iy \in \mathbb{C}^P$, where $x = \Re\Theta$, $y = \Im\Theta$ are real vectors and $\|\Theta\|_2^2 = \|x\|_2^2 + \|y\|_2^2$. Gradients and smoothness below refer to these $2P$ real coordinates. Define

$$\widetilde{\mathcal{L}}_{\text{real}}(\Theta) = \psi(\|\Theta\|_2) + \chi(\Re\Theta). \tag{2}$$

The extension is consistent ($\widetilde{\mathcal{L}}_{\text{real}}(x, 0) = \mathcal{L}_{\text{real}}(x)$) and has vanishing imaginary gradient at every real point: $\partial_y \widetilde{\mathcal{L}}_{\text{real}}(x, 0) = 0$.

This is the extension used in the path theorem and Corollary 1.1. A complex network's forward graph need not have this global radial form; its connection to the theory is discussed in Appendix D.1.

Theorem 1 (Bounded-loss complex continuation)**.** *Let $Z_\star$ be a strict local minimizer of $\mathcal{L}_{\text{real}}$ with $0 < R := \|Z_\star\|_2 \leq R_0$. For any unit $v \in \mathbb{R}^P$ orthogonal to $Z_\star$, the path*

$$\gamma(t) = R\big(\hat{Z}_\star \cos t + iv \sin t\big), \qquad 0 \leq t \leq \pi, \qquad \hat{Z}_\star := Z_\star/R,$$

is continuous in $\mathbb{C}^P$, satisfies $\gamma(0) = Z_\star$, $\gamma(\pi) = -Z_\star$ and $\|\gamma(t)\|_2 = R$, and obeys

$$\mathcal{L}_{\text{real}}(Z_\star) - \varepsilon R t \leq \widetilde{\mathcal{L}}_{\text{real}}(\gamma(t)) \leq \mathcal{L}_{\text{real}}(Z_\star) + \varepsilon R t.$$

The loss difference between the two real endpoints satisfies $|\mathcal{L}_{\text{real}}(-Z_\star) - \mathcal{L}_{\text{real}}(Z_\star)| \leq 2\varepsilon R$.

Proof sketch. The path trades real norm for imaginary norm while keeping $\|\gamma(t)\|_2 = R$. Thus, the radial term is constant and the loss difference is $\chi(Z_\star \cos t) - \chi(Z_\star)$. The mean-value argument applied on the real line segment bounds its absolute value by $\varepsilon\|Z_\star \cos t - Z_\star\|_2 = \varepsilon R(1 - \cos t)$. Since $1 - \cos t \leq t$ on $[0, \pi]$, the stated two-sided path bound follows. At $t = \pi$, $1 - \cos\pi = 2$ gives the endpoint bound $2\varepsilon R$. See Theorem 9 in the Appendix for the full proof. □

Corollary 1.1 (Escape beyond bounded basins)**.** *Under the C^1 assumptions in Section 3, every real strict local minimum with a bounded basin for a fixed real optimization rule admits a complex path to a real point outside that basin, with the explicit finite loss bound.*

Proof sketch. We enclose the basin in $\|Z - Z_\star\|_2 \leq D$, choose $R > D/2$, and set $c = Z_\star + Rd$ for unit d. For unit $v \perp d$, take $\gamma(t) = c - Rd\cos t + iRv \sin t$, $0 \leq t \leq \pi$. We show that its real endpoint $Z_\star + 2Rd$ lies beyond the basin. The loss bound follows from C^1 regularity on the compact region containing the path. The full proof is given in the Appendix C. □

Skew bilinear phase-invariant auxiliary. Having established the existence of a path with locally bounded loss increase, next we construct an auxiliary that initiates motion into complex coordinates.

We define the auxiliary loss as follows

$$\mathcal{L}_M(\Theta) = \Im\langle\Theta, M\Theta\rangle_{\mathbb{C}} = 2x^\top My, \qquad M^\top = -M. \tag{3}$$

Here $M \in \mathbb{R}^{P\times P}$ is fixed, yet arbitrary chosen skew-symmetric matrix ($M^\top = -M$), and the complex inner product is $\langle a, b\rangle_{\mathbb{C}} = \overline{a}^\top b$.

It is straightforward to show that $\mathcal{L}_M$ is phase-invariant ($\mathcal{L}_M(e^{i\phi}\Theta) = \mathcal{L}_M(\Theta)$), and its imaginary gradient at any real $\Theta = x$ is $\partial_y \mathcal{L}_M(x, 0) = -2Mx \in x^\perp$, which is nonzero whenever $x \notin \ker M$. With a positive weight, the auxiliary thus supplies a deterministic imaginary force at real points, while $\widetilde{\mathcal{L}}_{\text{real}}$ contributes none. Both gradient identities are derived in Lemmas 8 and 16 in the appendix and concrete choices of M are given in Remark 6 therein.

The CC-ADMM method. At iteration k, we minimize the training-time objective $\mathcal{L}_{\text{train}}(\Theta) := \widetilde{\mathcal{L}}_{\text{real}}(\Theta) + \lambda_k \mathcal{L}_M(\Theta)$ under the consensus constraint $\Theta = Z$, $Z \in \mathbb{R}^P$, by the Alternating Direction Method of Multipliers (ADMM), a suitable and widely used optimization algorithm for such linearly constrained problems (Glowinski & Marroco, 1975; Boyd et al., 2011).

Introducing the scaled dual $U \in \mathbb{C}^P$, the scaled augmented Lagrangian with consensus penalty $\rho_k > 0$ and auxiliary weight $\lambda_k \geq 0$ reads

$$\mathcal{L}_{\rho_k,\lambda_k}(\Theta, Z, U) = \mathcal{L}_{\text{train}}(\Theta) + \tfrac{\rho_k}{2}\|\Theta - Z + U\|_2^2 - \tfrac{\rho_k}{2}\|U\|_2^2, \qquad Z \in \mathbb{R}^P. \tag{4}$$

The last term is constant in (Θ, Z) and is kept for completeness of the scaled form. ADMM primal minimization and a scaled dual update yield the CC-ADMM iterations

$$\Theta^{k+1} \in \arg\min_{\Theta \in \mathbb{C}^P} \{\mathcal{L}_{\text{train}}(\Theta) + \tfrac{\rho_k}{2}\|\Theta - (Z^k - U^k)\|_2^2\}, \tag{5}$$

$$Z^{k+1} = \Re(\Theta^{k+1} + U^k), \tag{6}$$

$$U^{k+1} = U^k + \Theta^{k+1} - Z^{k+1}. \tag{7}$$

On the real consensus subspace, $\widetilde{\mathcal{L}}_{\text{real}}(Z) = \mathcal{L}_{\text{real}}(Z)$ and $\mathcal{L}_M(Z) = 0$. Notably, the task loss minimization is handled by the primal block, while the Z-update is a real projection and the dual accumulates its violation, i.e., imaginary direction excurse.

Theorem 2 (Deterministic departure from a real stationary point)**.** *Let $Z_\star$ be a real local minimizer, initialize $\Theta^k = Z^k = Z_\star$ and $U^k = 0$, and assume $MZ_\star \neq 0$ and $\lambda_k > 0$. Suppose that the gradient of the* complex extension*, viewed on $\mathbb{R}^{2P}$, is globally $\widetilde{L}$-Lipschitz and*

$$\rho_k > \widetilde{L} + 2\lambda_k \|M\|_{\text{op}}. \tag{8}$$

Then the primal subproblem has a unique minimizer with $\Im\Theta^{k+1} \neq 0$.

Proof sketch. The penalty condition makes the primal objective strongly convex. If its minimizer were real, imaginary stationarity would give $Mx^{k+1} = 0$. Real stationarity and $\nabla\mathcal{L}_{\text{real}}(Z_\star) = 0$ give $\|x^{k+1} - Z_\star\|_2 \leq (\widetilde{L}/\rho_k)\|x^{k+1} - Z_\star\|_2$, hence $x^{k+1} = Z_\star$ by the assumption (8). This contradicts $MZ_\star \neq 0$. Theorem 17 in the appendix gives the full proof and a local version with an interior-existence condition. □

After imaginary exploration during training the following result shows by using established SGD convergence theory that the CC-ADMM method converges to a real stationary point of the task loss.

Theorem 3 (Convergence under annealing and real projection)**.** *Assume $\mathcal{L}_{\text{real}}$ is coercive ($\mathcal{L}_{\text{real}}(Z) \to \infty$ as $\|Z\|_2 \to \infty$) and bounded below, with globally Lipschitz gradient. After a finite exploration phase ending at a fixed iteration k_0, set $\lambda_k = 0$, project the model onto $\mathbb{R}^P$, and disable consensus and dual updates. For $k \geq k_0$, use real SGD, $Z^{k+1} = Z^k - \eta_k g_k$, where $g_k \in \mathbb{R}^P$ is a stochastic estimate of $\nabla\mathcal{L}_{\text{real}}(Z^k)$. Assume these estimates are conditionally unbiased with uniformly bounded conditional noise variance, given the exploration history and all information available before sampling g_k. The deterministic learning rates satisfy $\eta_k > 0$, $\sum_{k\geq k_0}\eta_k = \infty$, $\sum_{k\geq k_0}\eta_k^2 < \infty$. Then, with probability one, $\mathcal{L}_{\text{real}}(Z^k)$ converges to a finite value, the real iterates are bounded, $\|\nabla\mathcal{L}_{\text{real}}(Z^k)\|_2 \to 0$, and every limit point Z^∞ satisfies $\nabla\mathcal{L}_{\text{real}}(Z^\infty) = 0$.*

Proof sketch. After the auxiliary loss vanishes and the model is projected to real parameters, the remaining iterations are SGD on $\mathcal{L}_{\text{real}}$. The finite exploration phase, followed by real projection, provides a valid starting point for SGD. Under the stated assumptions, convergence then follows from Proposition 3 of Bertsekas & Tsitsiklis (2000), which gives convergence of the loss and vanishing gradients with probability one, with the exploration history included in the conditioning information. Appendix C.6 restates this theorem and gives its full proof. □

In addition to the stochastic gradient descent, the separate Proposition 20 in the appendix covers exact proximal and deterministic gradient-descent updates. In the Experimental section, the reported learning experiments use $K = 5$ stochastic AdamW inner steps on the primal objective, at the matched baseline learning rate and schedule. After projection, CC-ADMM follows standard AdamW and inherits any convergence result whose assumptions cover the projected parameters, inherited optimizer state and subsequent training settings. The method is summarized in Algorithm 1 in the appendix.

Table 1: Grokking: optimizer steps required to reach test accuracy 0.99 (mean $\pm$ population standard deviation over successful runs) and measures time it takes for model to switch from overfitting to generalization (a.k.a. "grokking"). Each method succeeds in $5/5$ runs on every arithmetic task and $4/5$ on non-Abelian S_5. Bold marks the faster mean. Parameter counts are given in Table 11.

Method	$a+b$	$a-b$	$a\times b$	a/b	S_5
Baseline	$14\,966 \pm 4492$	$23\,365 \pm 6300$	$11\,274 \pm 4644$	$\mathbf{18\,556 \pm 3460}$	$102\,521 \pm 107\,380$
CC-ADMM	$\mathbf{9\,068 \pm 1421}$	$\mathbf{16\,133 \pm 5750}$	$\mathbf{9\,548 \pm 1677}$	$18\,767 \pm 1448$	$\mathbf{20\,700 \pm 22\,554}$

5 Results and Discussion

We evaluate the proposed CC-ADMM method on five problem families: (i) modular-arithmetic *grokking* on transformers, (ii) *from-scratch image classification* on CIFAR-100 (MLP and CNN), (iii) *rotation regression* on CIFAR-10 with ResNet-34, (iv) *instruction-tuning* billion-scale language models (Llama-3.2-1B and 3B on Alpaca), and (v) *Vision Transformer* classification (ViT-Tiny on CIFAR-100), trained from scratch with a basic augmentation recipe. Additional MNIST/CIFAR-10 results, detailed settings and dataset/model sources appear in Appendix E.

Method labels and training. The models reported per family depend on whether the Baseline is a full real model or a low-rank (LR) adapter. From-scratch families compare a real-only AdamW Baseline and CC-ADMM. The MLP/CNN use modReLU and a magnitude readout $|z_L|$, whereas their Baselines use ReLU and ViT uses a real forward map with auxiliary imaginary parameters. The Llama families compare three adapter architectures at outer rank 16: Baseline (QLoRA), CC-ADMM LR with an inner low-rank complex bottleneck, and CC-ADMM with a lifted input adapter factor. Table 11 in the Appendix further reports deployed counts and architectural differences. During the exploration phase, CC-ADMM takes $K = 5$ stochastic AdamW steps on the task loss plus auxiliary and consensus terms between real-consensus and dual updates. At the final phase after annealing these terms and dual updates are disabled the AdamW optimization continues on the real model. Appendix A.2 provides the implementation details.

5.1 Grokking on modular arithmetic

Setup: Following the grokking setting (Power et al., 2022; Agrawal & Kumar, 2025), we train a two-layer transformer on five tasks: addition, subtraction, multiplication and division over $\mathbb{Z}_{97}$, and non-Abelian permutation composition on S_5. Thirty percent of the modular token pairs are used for training and the remaining 70% for testing. We measure the *threshold step*-the first optimizer step at which test accuracy reaches 0.99-and the *grok rate*, the fraction of seeds reaching this threshold within the budget. Both methods share the optimizer and total step budget, while baselines use the same or more parameters than the CC-ADMM variants at deployment (Table 11).

Findings: Table 1 shows the largest reduction on the non-Abelian S_5 task: CC-ADMM reaches the threshold in approximately $5\times$ fewer steps than the Baseline ($20\,700$ vs. $102\,521$). Both models have one unsuccessful seed at the chosen budget, while CC-ADMM has a narrower distribution among successful runs. On the modular operations, the reduction is more modest: approximately $1.65\times$ on $a+b$, $1.45\times$ on $a-b$, $1.18\times$ on $a\times b$, and essentially a tie on a/b. To ablate the effect of the auxiliary, experiments in Appendix E.3 record $24/25$ successful "grokking" runs with the auxiliary loss on and $4/25$ without it across the five tasks, supporting the theory of the auxiliary loss inducing an escape/exploration mechanism.

5.2 Image classification on CIFAR-100

Setup: We train an MLP and CNN from scratch on CIFAR-100 with cross-entropy and AdamW (learning rate 10^{-3}, weight decay 10^{-4}, batch size 128, 100 epochs). The comparisons differ in architecture and deployed counts with each Baseline having more than 10% more real parameters to the baseline (Table 11). To further include augmented smoothed baselines, separate ResNet-18 runs are scheduled for 300, 900 and 2700 epochs, using MixUp 0.8, CutMix 1.0 augmentations and label

smoothing 0.1. Both ResNet-18 models deploy 11.22M parameters. We report test accuracy at the end of each training run. Additional MNIST and CIFAR10 runs are in the appendix, Table 5 & 6.

Results and Findings: Final-epoch CIFAR-100 accuracies are reported as mean ± population SD over three seeds, with CC-ADMM outperforming Baselines for both architectures (CC-ADMM vs. Baseline). At 100 epochs, the MLP achieves 0.2153 ± 0.0018 versus 0.2613 ± 0.0031 ($+4.60$ percentage points), and the CNN achieves 0.6104 ± 0.0034 versus 0.6519 ± 0.0016 ($+4.14$ points).

For augmented and smooth ResNet-18, the corresponding results (CC-ADMM vs. Baseline) are 0.7838 ± 0.0014 versus 0.7599 ± 0.0021 at 300 epochs, 0.8019 ± 0.0018 versus 0.8048 ± 0.0016 at 900 epochs, and 0.7944 ± 0.0005 versus 0.8036 ± 0.0015 at 2700 epochs. The ResNet-18 gains are therefore -2.40, $+0.28$, and $+0.92$ percentage points, respectively. The improved generalization thus comes at higher training costs of CC-ADMM: at 300 epochs, its logged wall-clock time, incl. evaluations, is $1.49\times$ the Baseline's and peak allocated memory during exploration is $1.60\times$.

5.3 Rotation regression on CIFAR-10 (ResNet-34)

Setup: We also test CC-ADMM on a non-classification objective: predicting the rotation angle of a randomly rotated CIFAR-10 image. A ResNet-34 with a two-dimensional regression head is trained from scratch with MSE loss on the $(\sin\theta, \cos\theta)$ output. We report test MSE and angular error in degrees. All models use AdamW (learning rate 10^{-3}, batch 128, 250 epochs, three seeds).

Findings: CC-ADMM reduces the final test MSE from 0.3920 ± 0.0046 to 0.3705 ± 0.0146 and the final angular error from $40.22° \pm 0.41$ to $37.38° \pm 1.38$, an approximately $3°$ absolute gain (Table 7). This extends the predictive-performance benefit of the method to a regression objective.

5.4 Instruction-tuning Llama-3.2 on Alpaca

Setup: One of the most common practical machine learning task nowadays is finetuning of pretrained large model. In what follows, we test CC-ADMM on finetuning the LLama 1B and 3B models. We use QLoRA-style instruction-tuning of frozen NF4-quantized Llama-3.2-1B and 3B on the Stanford Alpaca corpus (Taori et al., 2023), with adapters at all seven linear projections per transformer block. The three models share the outer adapter rank $r = 16$: Baseline (QLoRA) uses real linear adapter factors, CC-ADMM lifts the input adapter factor to complex parameters, and CC-ADMM LR uses a real input projection followed by a complex bottleneck of width $s = 8$. Both CC-ADMM variants use modReLU and magnitude pooling before a real output projection. CC-ADMM adds a small real bias to the Baseline adapter count, while CC-ADMM LR deploys at 73.39% and 74.64% of that count at 1B and 3B, respectively.

Table 2: Final real-model Llama results: held-out Alpaca cross-entropy and WikiText-103 perplexity (lower is better). Values are mean ± std. over seeds. The results in bold are significantly (3 std) better than baselines.

	Llama-3.2-1B		Llama-3.2-3B	
Method	Alpaca loss	WikiText PPL	Alpaca loss	WikiText PPL
Baseline (QLoRA)	1.4343 ± 0.0024	14.085 ± 0.046	1.2974 ± 0.0005	10.317 ± 0.031
CC-ADMM LR	**1.3991** ± 0.0007	**12.299** ± 0.031	**1.2646** ± 0.0007	**9.394** ± 0.011
CC-ADMM	**1.3949** ± 0.0006	**12.360** ± 0.016	**1.2633** ± 0.0003	**9.388** ± 0.018

Findings: Both CC-ADMM variants achieve lower final held-out Alpaca loss and OOD perplexity than the QLoRA Baseline at both scales (Table 2). At 1B, CC-ADMM has the lowest final Alpaca perplexity (4.035 vs. 4.197 for the Baseline), while CC-ADMM LR has the lowest final OOD perplexity (12.299 vs. 14.085, a 12.7% reduction). The two CC-ADMM variants have similar final OOD perplexity (9.388 and 9.394, respectively, within 3 std. apart), compared with 10.317 for the Baseline. This renders the CC-ADMM LR variant especially interesting, as it has achieved this with $1/5$ less parameters. Additionally, the elapsed training time in Figure 2 highlights that exploration possibly leading to better generalization comes with costs of training time overhead. Thus, CC-ADMM is the method suitable for problems where such a trade-off makes sense, i.e., large and non-convex optimizations.

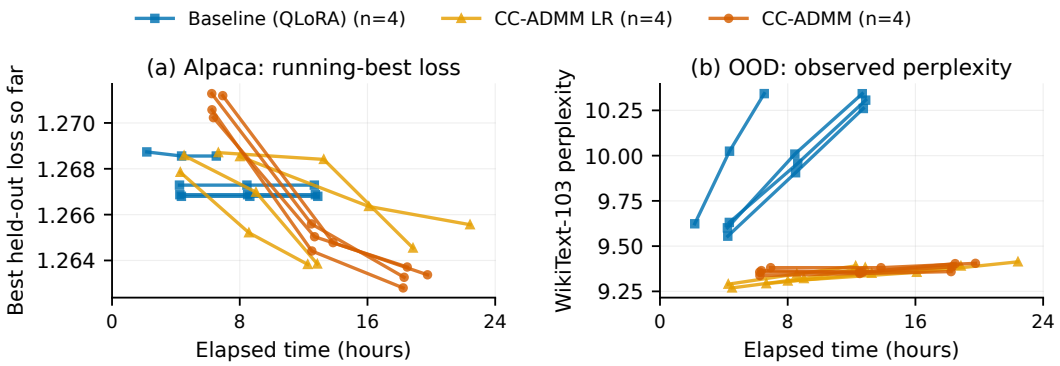

Figure 2: Llama-3.2-3B fine-tuning versus logged training and evaluation time (four runs per method). Left: lowest held-out Alpaca loss observed up to each evaluation. Right: WikiText-103 perplexity measured at each evaluation. Markers denote epoch-end evaluations. For both CC-ADMM variants, the first two evaluations use complex parameters, while the third evaluates the projected model after continued real-valued training. Table 2 reports final-epoch metrics over all completed runs and Appendix E.5 provides the complementary observed epoch-end trajectories and best/final comparisons.

Table 3: ViT-Tiny on CIFAR-100 (three seeds, mean $\pm$ sample standard deviation). Both models deploy 5.38M parameters with the same real forward map. Training state and computation differ.

Method	n	best test acc	final test acc	best test loss	final test loss
Baseline	3	0.5977 ± 0.0055	0.5964 ± 0.0047	1.6987 ± 0.0143	2.7572 ± 0.0707
CC-ADMM	3	$\mathbf{0.6101 \pm 0.0051}$	$\mathbf{0.6060 \pm 0.0050}$	$\mathbf{1.4773 \pm 0.0196}$	$\mathbf{2.0952 \pm 0.0424}$

5.5 ViT-Tiny from scratch on CIFAR-100

Setup: A ViT-Tiny (5.38M deployed parameters) is trained from scratch on CIFAR-100 with cross-entropy and AdamW (learning rate 5×10^{-4}, weight decay 0.05, batch 256, 200 epochs, cosine schedule with 2000 warmup steps, three seeds). The Baseline is a plain real ViT. CC-ADMM replaces every transformer-block linear layer with a complex-shadow layer: its imaginary parameters are trained through the auxiliary loss and consensus constraint, while the task forward map uses the same real linear operation as the Baseline. The augmentation pipeline is based on random crop and horizontal flip, without MixUp, CutMix, RandAugment or label smoothing.

Findings: CC-ADMM improves best test accuracy from 0.5977 to 0.6101 and final test accuracy from 0.5964 to 0.6060 (Table 3). Its final test loss is approximately 24% lower. Further architecture and recipe details appear in Appendix E.6, suggesting reduced late-training overfitting.

6 Conclusion

We introduced the **Complex-Continuation ADMM** (CC-ADMM), a training-time lift of a real network into a complex parameter space, governed by a phase-invariant auxiliary loss $\mathcal{L}_M$ and real ADMM consensus constraint. After finite exploration, the imaginary parameters are zeroed and frozen, leaving a purely real deployed model. Our theory establishes bounded-loss complex paths and deterministic departure from real stationary points under explicit assumptions, followed by convergence to stationary point with probability one for SGD on the recovered real domain. CC-ADMM thus provides a versatile tool for exploring non-convex optimization landscapes. **Limitations and future work.** The solution found by the proposed CC-ADMM method depends on the architecture and geometry of the loss region explored. This is in line with existing nonconvex ADMM analyses requiring conditions on the objective and constraints, together with suitable penalties and subproblem updates (Hong et al., 2016; Wang et al., 2019b; Li & Pong, 2015). Extending the analysis to wider settings and adaptive choices of M, ρ_k and λ_k, and alternative ways to return to the real parameters, are directions for future work.

REPRODUCIBILITY STATEMENT

The main text specifies the practical inner solver and consensus updates. The appendices give the mathematical assumptions and proofs, implementation details, training protocols, and additional results. The SGD convergence assumptions and the experimental AdamW settings are identified separately, and the experimental comparisons report their training and deployment costs. Reproducibility code is uploaded as a separate supplementary material alongside this paper.

AI USE STATEMENT

In this work, generative AI tools were used for generating data set and loss objective of a double-well landscape for synthetic experiment of Figure 1., obtaining code and datasets from existing github repositories, as a coding assistant and a feedback loop in writing the proofs. Additionally, we used generative AI tools for aid and polish writing. Specifically for checking and correcting english (grammar) and editing the paper to improve readability. Further we used AI for retrieval and research-papers discovery, novelty checking and finding related work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rayyan Abdalla. Complex-valued neural networks – theory and analysis, 2023. URL https://arxiv.org/abs/2312.06087.

Tathagat Agrawal and Manoj Kumar. Grokking in neural networks: A review. *SN Computer Science*, 6(6):627, Jul 2025. ISSN 2661-8907. doi: 10.1007/s42979-025-04182-z. URL https://doi.org/10.1007/s42979-025-04182-z.

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=CQsmMYmlP5T.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 242–252. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/allen-zhu19a.html.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1120–1128, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/arjovsky16.html.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Gregory W Benton, Wesley J Maddox, Sanae Lotfi, and Andrew Gordon Wilson. Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning*, 2021.

Dimitri P. Bertsekas and John N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, January 2000. ISSN 1095-7189. doi: 10.1137/s1052623497331063. URL http://dx.doi.org/10.1137/S1052623497331063.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. doi: 10.1561/2200000016. URL http://dx.doi.org/10.1561/2200000016.

Andrei Caragea, Dae Gwan Lee, Johannes Maly, Götz Pfander, and Felix Voigtlaender. Quantitative approximation results for complex-valued neural networks. *SIAM Journal on Mathematics of Data Science*, 4(2):553–580, May 2022. ISSN 2577-0187. doi: 10.1137/21m1429540. URL http://dx.doi.org/10.1137/21M1429540.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1309–1318. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/draxler18a.html.

Simon Du and Jason Lee. On the power of over-parametrization in neural networks with quadratic activation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1329–1338. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/du18a.html.

Simon Du, Chi Jin, Jason D Lee, Michael Jordan, Aarti Singh, and Barnabas Poczos. Gradient descent can take exponential time to escape saddle points. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f79921bbae40a577928b76d2fc3edc2a-Paper.pdf.

Simon S. Du, Xiyu Zhai, Barnabás Póczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=S1eK3i09YQ.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/be3087e74e9100d4bc4c6268cdbe8456-Paper.pdf.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1233–1242. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/ge17a.html.

Paul Geuchen and Felix Voigtlaender. Optimal approximation using complex-valued neural networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 1681–1737. Curran Associates, Inc., 2023. doi: 10.52202/075280-0084. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/05b69cc4c8ff6e24c5de1ecd27223d37-Paper-Conference.pdf.

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975. URL https://www.numdam.org/item/M2AN_1975__9_2_41_0/.

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/01d8bae291b1e4724443375634ccfa0e-Paper.pdf.

Le Thi Khanh Hien, Duy Nhat Phan, and Nicolas Gillis. Inertial alternating direction method of multipliers for non-convex non-smooth optimization. *Computational Optimization and Applications*, 83(1):247–285, Sep 2022. ISSN 1573-2894. doi: 10.1007/s10589-022-00394-8. URL https://doi.org/10.1007/s10589-022-00394-8.

Akira Hirose. *Complex-Valued Neural Networks*. Springer Berlin Heidelberg, 2012. ISBN 9783642276323. doi: 10.1007/978-3-642-27632-3. URL http://dx.doi.org/10.1007/978-3-642-27632-3.

P.D. Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. *Comput. Statist. Data Anal.*, 115:186–198, 2017. ISSN 0167-9473. doi: 10.1016/j.csda.2017.06.007. URL https://doi.org/10.1016/j.csda.2017.06.007.

Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, January 2016. ISSN 1095-7189. doi: 10.1137/140990309. URL http://dx.doi.org/10.1137/140990309.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

Tom Jacobs and Rebekka Burkholz. Mask in the mirror: Implicit sparsification. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 36221–36245, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5999c13eec1963e733794055034b9557-Paper-Conference.pdf.

Farhang Jalali, J. D. Seader, and Saeed Khaleghi. Global solution approaches in equilibrium and stability analysis using homotopy continuation in the complex domain. *Comput. Chem. Eng.*, 32(10):2333–2345, 2008. doi: 10.1016/J.COMPCHEMENG.2007.12.001. URL https://doi.org/10.1016/j.compchemeng.2007.12.001.

Bo Jiang, Shiqian Ma, and Shuzhong Zhang. Alternating direction method of multipliers for real and complex polynomial optimization models. *Optimization*, 63(6):883–898, March 2014. ISSN 1029-4945. doi: 10.1080/02331934.2014.895901. URL http://dx.doi.org/10.1080/02331934.2014.895901.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1724–1732. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/jin17a.html.

Kenji Kawaguchi. Deep learning without poor local minima. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/f2fc990265c712c49d51a18a32b39f0c-Paper.pdf.

Ken Kreutz-Delgado. The complex gradient operator and the cr-calculus, 2009. URL https://arxiv.org/abs/0906.4835.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/46a4378f835dc8040c8057beb6a2da52-Paper.pdf.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 0018-9219. doi: 10.1109/5.726791. URL http://dx.doi.org/10.1109/5.726791.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir (eds.), *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pp. 1246–1257, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL https://proceedings.mlr.press/v49/lee16.html.

Guoyin Li and Ting Kei Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015. doi: 10.1137/140998135. URL http://dx.doi.org/10.1137/140998135.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf.

Qingyuan Liu, Zhengchao Huang, Hao Ye, Dexian Huang, and Chao Shang. Accelerated nonconvex admm with self-adaptive penalty for rank-constrained model identification. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 1281–1287. IEEE, December 2023. doi: 10.1109/cdc49753.2023.10383766. URL http://dx.doi.org/10.1109/CDC49753.2023.10383766.

Ziye Ma, Igor Molybog, Javad Lavaei, and Somayeh Sojoudi. Over-parametrization via lifting for low-rank matrix sensing: Conversion of spurious solutions to strict saddle points. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23373–23387. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ma23f.html.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Quynh Nguyen. On connected sublevel sets in deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4790–4799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/nguyen19a.html.

Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2603–2612. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/nguyen17a.html.

Chicago Y. Park, Yuyang Hu, Michael T. McCann, Cristina Garcia-Cardona, Brendt Wohlberg, and Ulugbek S. Kamilov. Plug-and-play priors as a score-based method. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 49–54, Anchorage, AK, USA, September 2025. doi: 10.1109/ICIP55913.2025.11084503.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-play methods provably converge with properly trained denoisers. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5546–5557. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ryu19a.html.

Rajesh Shrestha and Xiao Fu. Taming score-based denoisers in admm: A convergent plug-and-play framework. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 13523–13568, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/16e18fa3b3add076c30f2a2598f03031-Paper-Conference.pdf.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: A scalable admm approach. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2722–2731, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/taylor16.html.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J. Pal. Deep complex networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1T2hmZAb.

Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948. IEEE, December 2013. doi: 10.1109/globalsip.2013.6737048. URL http://dx.doi.org/10.1109/GLOBALSIP.2013.6737048.

Luca Venturi, Afonso S. Bandeira, and Joan Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. *Journal of Machine Learning Research*, 20(133):1–34, 2019. URL http://jmlr.org/papers/v20/18-674.html.

Junxiang Wang, Fuxun Yu, Xiang Chen, and Liang Zhao. Admm for efficient deep learning with global convergence. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, New York, NY, USA, 2019a. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330936.

Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, Jan 2019b. ISSN 1573-7691. doi: 10.1007/s10915-018-0757-z. URL https://doi.org/10.1007/s10915-018-0757-z.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Ganzhao Yuan. Admm for nonconvex optimization under minimal continuity assumption. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 44235–44277, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6d6f9908ea35313dd7566f5ce8c6e815-Paper-Conference.pdf.

Huisheng Zhang, Xiaodong Liu, Dongpo Xu, and Ying Zhang. Convergence analysis of fully complex backpropagation algorithm based on wirtinger calculus. *Cognitive Neurodynamics*, 8 (3):261–266, Jun 2014. ISSN 1871-4099. doi: 10.1007/s11571-013-9276-7. URL https://doi.org/10.1007/s11571-013-9276-7.

Shijun Zhang, Jianfeng Lu, and Hongkai Zhao. Deep network approximation: Beyond relu to diverse activation functions. *Journal of Machine Learning Research*, 25(35):1–39, 2024. URL http://jmlr.org/papers/v25/23-0912.html.

Tianyun Zhang, Shaokai Ye, Kaiqi Zhang, Jian Tang, Wujie Wen, Makan Fardad, and Yanzhi Wang. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Liu Ziyin and Zihao Wang. spred: Solving l1 penalty with SGD. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 43407–43422. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ziyin23a.html.

A GLOSSARY OF TECHNICAL TERMS

For readers less familiar with the physics / complex-analysis vocabulary used throughout the paper, we collect short definitions of the following recurring terms: *gauge*, *skew gauge*, and *non-holomorphic*.

Gauge A *gauge* is a direction (or, more generally, a group action) in parameter space along which the observable quantity of interest does not change. The terminology comes from physics (a common global phase of a quantum state that does not change its physical predictions) and has been adopted in machine learning to describe any reparameterization that leaves the realized function, or the loss on every training input, invariant.

In our setting, the relevant example is the quadratic neuron $y = h^\top W\, h$ with $h \in \mathbb{R}^d$ and $W \in \mathbb{R}^{d\times d}$. For any square matrix A, the identity

$$h^\top A\, h \;=\; h^\top \operatorname{sym}(A)\, h + h^\top \operatorname{skew}(A)\, h \;=\; h^\top \operatorname{sym}(A)\, h,$$

holds because $h^\top S\, h = 0$ for every skew-symmetric $S = -S^\top$ and every h. Consequently, the output depends only on the symmetric part $\operatorname{sym}(W) = \frac{1}{2}(W + W^\top)$. The map $W \mapsto \operatorname{sym}(W)$ is a projection onto a *gauge slice*. Its pre-image is $W + \{S : S^\top = -S\}$, and moving along skew matrices is *motion along the gauge* - the parameter W changes, but the function does not.

Holomorphic vs. non-holomorphic A function $f : \mathbb{C} \to \mathbb{C}$ is *holomorphic* near a point z if it is complex-differentiable throughout a neighbourhood of z. For continuously differentiable real and imaginary parts, writing $z = x + iy$ and $f = u(x, y) + i\, v(x, y)$, this is equivalent to the Cauchy-Riemann equations

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \qquad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$$

holding throughout that neighborhood. Holomorphic functions are rigid: a holomorphic f is automatically infinitely differentiable and equal to its Taylor series on a disc.

Common operations used in complex-valued neural networks - in particular $|z|$, $\Re z$, $\Im z$, and modReLU $a(z) = \mathrm{ReLU}(|z| - b)\, z/|z|$ for $z \neq 0$, with $a(0) = 0$ - are generally *non-holomorphic*. For instance, $|z| = \sqrt{x^2 + y^2}$ has

$$\frac{\partial |z|}{\partial x} = \frac{x}{|z|}, \qquad \frac{\partial |z|}{\partial y} = \frac{y}{|z|},$$

whereas the Cauchy-Riemann equations (viewed as constraints on $|z|$ regarded as the real part of some complex-valued extension) would require the first quantity to match the derivative of the imaginary part, $\partial \Im(|z|)/\partial y = 0$, and the second equation also requires $y/|z| = 0$. Both requirements cannot hold at a nonzero z; at zero the modulus is not differentiable. Hence $|z|$ is nowhere holomorphic.

Consequences for training. The classical chain rule for complex-differentiable compositions does not apply to networks built from non-holomorphic blocks. Two equivalent standard workarounds exist:

1. **Wirtinger calculus.** Treat z and its conjugate $\bar{z}$ as independent variables and define

$$\frac{\partial}{\partial z} := \tfrac{1}{2}\left(\frac{\partial}{\partial x} - i\,\frac{\partial}{\partial y}\right), \qquad \frac{\partial}{\partial \bar{z}} := \tfrac{1}{2}\left(\frac{\partial}{\partial x} + i\,\frac{\partial}{\partial y}\right).$$

   The "non-holomorphic gradient" of a real-valued loss is $2\,\partial L/\partial \bar{z}$, and steepest-descent updates use this Wirtinger conjugate gradient (Kreutz-Delgado, 2009; Zhang et al., 2014).

2. **Real embedding.** Identify $\mathbb{C}^P \cong \mathbb{R}^{2P}$ via $\Theta = x + iy \;\leftrightarrow\; (x, y) \in \mathbb{R}^P \times \mathbb{R}^P$ and perform plain real-valued calculus on the embedded representation.

The two choices are mathematically equivalent. This paper uses the real embedding throughout. That is also the reason why the proofs of Lemma 15 and Theorem 17 are written in (x, y) coordinates. The implementation stores real and imaginary components as real tensors and differentiates them using ordinary *autograd*.

A.1 NOTES ON RADIAL-PHASE DECOMPOSITION OF THE LOSS INTO $\psi(\|Z\|_2)$ AND $\chi(Z)$

The radial-phase decomposition $\mathcal{L}_{\mathrm{real}}(Z) = \psi(\|Z\|_2) + \chi(Z)$ of Definition 7 is a conceptual helper for proving the existence of a continuous bounded-loss path in Theorem 9. It is a structural assumption on the loss landscape, together with the stated regional bounds. The CC-ADMM algorithm itself does not compute ψ or χ as distinct quantities, and the constants R_0 and ε are not estimated during training. Absorbing a nonradial term into χ may increase its gradient bound and hence the bound on loss variation.

The dynamics of CC-ADMM use a phase-invariant auxiliary loss with a nonzero imaginary gradient at a real stationary point $Z_\star$, supplied by $\mathcal{L}_M$ when $MZ_\star \neq 0$ (Lemma 16). With positive auxiliary loss weight and zero dual, this gives the departure result of Theorem 17 under its stated smoothness and update assumptions. The condition $MZ_\star \neq 0$ is separate from skewness: the implemented skew-circulant matrices have nontrivial kernels. The path theorem describes the loss along a constructed path; it does not assert that the optimizer tracks that path or reaches a better real minimum. Convergence after the finite real projection is treated separately in Theorem 3.

Radial structure in the implementation. Besides regularization, some of the architectures use the following activation-level structure:

- The classifier head reads the per-coordinate magnitude $|z_L| = \sqrt{z_{\mathrm{re},L}^2 + z_{\mathrm{im},L}^2 + \delta^2}$ of the last hidden layer, where $\delta > 0$ is a numerical smoothing constant. This readout is invariant under per-coordinate rotations of that hidden representation.
- The intermediate `modReLU` activation $\sigma_{\mathrm{modReLU}}(z) = z\,\mathrm{ReLU}(|z| + b)/|z|$, with its value at zero set to zero, is phase-equivariant: it scales the magnitude but leaves the phase untouched. Here $z \in \mathbb{C}$ and $b \in \mathbb{R}$ use the additive-bias convention, opposite to the threshold convention above.

These properties concern activations. They do not, by themselves, make the loss radial in the full vector of network parameters or give a small residual-gradient bound. The ViT forward map is real, and Llama lifts selected adapter modules, as detailed in the experimental setup.

A.2 THEORY–IMPLEMENTATION GLOSSARY

The following table makes the symbol-by-symbol correspondence explicit between the theoretical CC-ADMM and its implementation. For partial lifting, only selected modules have imaginary and consensus variables; the real-phase SGD theorem applies to the full vector of trainable real parameters, including the unlifted ones.

Table 4: Correspondence between the mathematical variables and the implementation.

Theory	Implementation	Notes / differences
$\Theta = x + iy \in \mathbb{C}^P$	$W^{(\ell)} = W_{\mathrm{re}}^{(\ell)} + iW_{\mathrm{im}}^{(\ell)} \in \mathbb{C}^{m_\ell \times n_\ell}$; biases $b_{\mathrm{re}}^{(\ell)} + ib_{\mathrm{im}}^{(\ell)} \in \mathbb{C}^{m_\ell}$	Theory: a single complex vector. Code: one block per lifted module, flattening non-output dimensions of convolution kernels. With biases, the lifted block has $P_{\mathrm{lift}} = \sum_\ell (m_\ell n_\ell + m_\ell)$ complex coordinates. Unlifted parameters remain real.
$Z \in \mathbb{R}^P$	$X^{(\ell)} \in \mathbb{R}^{m_\ell \times n_\ell}$, plus one block per bias	Real consensus variable, stored per layer. The imaginary component of Z is zero; the imaginary component of the primal center $Z - U$ is $-U_{\mathrm{im}}$.
$U \in \mathbb{C}^P$	$U_{\mathrm{re}}^{(\ell)}, U_{\mathrm{im}}^{(\ell)}$, stored separately	Scaled dual variable, with the same shapes as the corresponding weights and biases.
$\mathcal{L}_{\mathrm{real}}(Z)$	The real-parameter training objective	The experiments use classification cross-entropy, regression MSE or response-token cross-entropy. The task loss appears once in the Θ block; the other block enforces real consensus.

Theory	Implementation	Notes / differences
$\widetilde{\mathcal{L}}_{\mathrm{real}}(\Theta)$	The forward pass of the lifted model	The theory defines a particular additive radial extension. The implemented forward extension is architecture dependent and need not equal that formula.
$\psi(\|Z\|_2)$, $\chi(Z)$	Not computed separately	Used in the path theorem. Neither magnitude readouts nor modReLU alone verifies the regional bound $\|\nabla\chi\|_2 \le \varepsilon$.
$M^{(\ell)}$ skew	$M^{(\ell)} = S - S^\top$, where S shifts the output coordinates cyclically	Its norm is at most 2, equal to 2 when 4 divides m_ℓ. The constant vector lies in its kernel; for $m_\ell = 2$ the matrix is zero.
$\mathcal{L}_M(\Theta) = 2x^\top My$	$\mathcal{L}_M = \sum_\ell a_\ell[2\,\mathrm{tr}((W_{\mathrm{re}}^{(\ell)})^\top M^{(\ell)} W_{\mathrm{im}}^{(\ell)}) + 2(b_{\mathrm{re}}^{(\ell)})^\top M^{(\ell)} b_{\mathrm{im}}^{(\ell)}]$	Column-wise vectorization gives weight blocks $a_\ell(I_{n_\ell} \otimes M^{(\ell)})$ and bias blocks $a_\ell M^{(\ell)}$. The positive coefficients a_ℓ account for summing or averaging over lifted modules.
$\lambda_k \ge 0$	Annealed auxiliary loss weight	Auxiliary loss shutoff and the finite transition to real parameters can occur at different times. Schedules are experiment-specific.
$\rho_k > 0$	Consensus penalty during exploration	A positive penalty alone does not establish the curvature or learning-rate conditions of the convergence results.
Θ-update, (11)	$K = 5$ stochastic AdamW steps between consensus updates	The primal objective contains the task loss, the auxiliary loss, and the quadratic consensus constraint term. These are inexact updates; the exact-primal proof does not apply to them without further assumptions.
Z-update, (12)	$X^{(\ell)} \leftarrow W_{\mathrm{re}}^{(\ell)} + U_{\mathrm{re}}^{(\ell)}$	The projection exactly solves the real-consensus subproblem of the one-loss splitting. There is no second task-loss proximal subproblem.
U-update, (13)	$U_{\mathrm{re}}^{(\ell)} \leftarrow U_{\mathrm{re}}^{(\ell)} + W_{\mathrm{re}}^{(\ell)} - X^{(\ell)}$; $U_{\mathrm{im}}^{(\ell)} \leftarrow U_{\mathrm{im}}^{(\ell)} + W_{\mathrm{im}}^{(\ell)}$	The basic dual update uses imaginary consensus constraint target zero. Optional controller modifications are described in Appendix D.
Real phase	Zero and freeze imaginary parameters; disable continuation updates	The task optimizer continues on the real model. Theorem 3 treats SGD under its stated conditions; the experiments use AdamW.

Vectorizing the auxiliary loss shows that its gradient structure agrees with Lemma 16: $\partial_{W_{\mathrm{im}}^{(\ell)}} \mathcal{L}_M(W_{\mathrm{re}}, 0) = -2a_\ell M^{(\ell)} W_{\mathrm{re}}^{(\ell)}$. MLP/CNN and literal grokking use sums ($a_\ell = 1$). Besides, the other implementations average over lifted modules. The real projection and basic dual updates match (31)–(32). The implementation does not solve the primal exactly, and the finite real transition is imposed explicitly. In the strong-recipe ResNet comparison, exploration ends at epoch 165 at each training budget.

A.3 Synthetic trajectory settings

For Figure 1, let $w = (x_1, x_2, y_1, y_2)$, $\delta = 10^{-8}$, $r = \sqrt{\|w\|_2^2 + \delta^2}$, $a = w\max(r - 0.2, 0)/(r + \delta)$, and $q = \sqrt{\|a\|_2^2 + \delta^2}$. The implemented task loss is

$$\mathcal{T}(w) = (q-1)^2 + \tfrac{1}{2}\left(1 - \frac{a_1}{q+\delta}\right) + \tfrac{1}{2}x_2^2.$$

This is a joint norm gate, not an elementwise activation. Ignoring numerical smoothing, the real minima are near $(x_1, x_2) = (-1.2, 0)$ and $(1.2, 0)$, with losses 1 and 0, respectively. The negative real minimum is already a saddle of the complex extension. Inside the gate, the loss is approximately $1.5 + x_2^2/2$, so the whole real-plane disc is not flat.

All methods start at $w = (-1, 0, 0, 0)$ and take 2000 steps with learning rate 0.02. Real gradient descent (GD) and real AdamW optimize $\mathcal{T}(x_1, x_2, 0, 0)$; AdamW uses zero weight decay and default moments $(0.9, 0.999)$. The CC trace takes one full-batch gradient step on $\mathcal{T} + \lambda_k\mathcal{L}_M + \rho_k\|\Theta - Z +$

$U\|^2/2$, followed by the real projection and basic dual update. The initial consensus equals the real initialization and the dual is zero. With $\sigma(u) = (1 + e^{-u})^{-1}$,

$$\lambda_k = \tfrac{1}{4}[1 + \cos(\pi \min\{k/1400, 1\})], \qquad \rho_k = 0.01 + 9.99\,\sigma\left(\frac{k/2000 - 0.7}{0.05}\right).$$

The plotted values are recorded before each update. The toy retains consensus updates throughout; it does not perform the explicit restart of Theorem 3. Its full complex trajectory avoids the gate's dead zone even when its real projection crosses that disc. The task loss depends on the full norm in its angular denominator and therefore is not the specific additive extension of Theorem 1. Escape and return in this run are numerical observations.

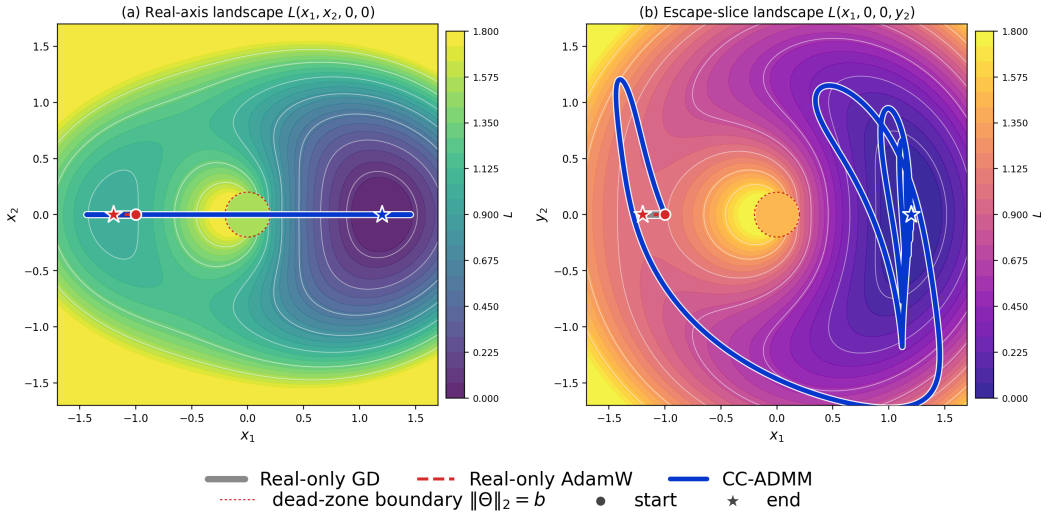

Figure 3: Extended view of Figure 1: **two-dimensional $P = 2$ complex-continuation toy example.** Panel (a) renders the real slice (x_1, x_2); panel (b) renders the slice (x_1, y_2) used during the imaginary excursion. All three optimizers start at the shared real point $(-1, 0)$. Real-only GD and AdamW stay on the real axis and approach the negative real local minimum. In this run, CC-ADMM leaves the real axis through y_2 and approaches the positive real minimum. Its projection crosses the activation dead-zone disc, while the full complex trajectory avoids it. The settings and scope of this numerical illustration are given above.

B Additional related work

Complex-valued networks and their optimization. Complex-valued networks use complex weights and activations to represent information through both magnitude and phase. Their foundations and applications are described in the monograph of Hirose (2012) and the survey preprint of Abdalla (2023). Arjovsky et al. (2016) introduce unitary recurrent networks, while Trabelsi et al. (2018) develop building blocks like complex convolution, batch normalization and initialization for deep complex networks. Training does not require every operation to have a complex derivative: the real-valued loss can be differentiated with respect to the real and imaginary parts of each parameter. Wirtinger calculus expresses these derivatives following Cauchy-Riemann (CR) equations of complex calculus, as explained in the technical report of Kreutz-Delgado (2009). Using this calculus, Zhang et al. (2014) establish convergence results for complex backpropagation under assumptions on the activations, boundedness of the weights and step size. Their analysis assumes analytic activations and does not cover modReLU. CC-ADMM uses complex coordinates temporarily during exploration and returns to a real model for the final optimization phase.

Magnitude-based activations and approximation. ModReLU changes a complex activation's magnitude while preserving its phase whenever the output is nonzero (Arjovsky et al., 2016; Trabelsi

et al., 2018). Using the threshold convention, it is

$$a(z) = \max\{|z| - b, 0\}\frac{z}{|z|} \quad \text{for } z \neq 0, \qquad a(0) = 0.$$

For $b \geq 0$, inputs in the disc $|z| \leq b$ produce zero output. On the real axis, this becomes $a(x) = \operatorname{sign}(x)\max\{|x| - b, 0\}$, which generally differs from ReLU. Caragea et al. (2022) bound the approximation error of deep modReLU networks for smooth target functions. Geuchen & Voigtlaender (2023) obtain approximation rates for "shallow" single-hidden-layer complex networks under activation assumptions that also include modReLU. These results describe what the networks can represent and how the approximation error decreases with network size. They do not determine whether training finds the required weights. The radial behavior of an individual activation also does not establish the radial-phase assumptions on the full parameter loss in our path theorem. In our experiments, the MLP/CNN comparisons change activations and readouts relative to their real baselines, while the ViT construction retains a real forward map. Section 5 specifies these choices.

Network structure and parameter symmetries. The geometry of a training loss depends on the architecture and objective. For deep linear networks with squared loss, Kawaguchi (2016) establishes conditions under which every local minimum is globally optimal. Nguyen (2019) gives conditions on width, activations, and training data under which sublevel sets are connected: parameter vectors below a given loss level can be joined without exceeding that level. Du & Lee (2018) show how sufficient width can remove non global local minima in one-hidden-layer networks with quadratic activation and fixed output weights, under their loss and regularization (weight decay) assumptions. These results rely on the particular network structures studied. When a continuous parameter symmetry preserves both the model output and any regularization term, it gives a path of constant loss. Moving along that path does not lower the loss. In CC-ADMM, the skew auxiliary loss supplies a nonzero imaginary gradient when $Mx \neq 0$, providing the departure mechanism analyzed in Theorem 2.

C Method: Complex-Continuation ADMM with a skew phase-invariant auxiliary

C.1 Problem setting

We consider supervised learning with real inputs and real outputs. Let $Z \in \mathbb{R}^P$ be the parameters of a real-valued network and $\mathcal{L}_{\text{real}}(Z)$ a standard task loss (cross-entropy, MSE, ...). Our goal is to minimize $\mathcal{L}_{\text{real}}$. When the real minima lie in distinct connected components of a sub=level set, a continuous path between them must traverse higher-loss regions. Our aim is a *deterministic continuation mechanism* that enlarges the optimization geometry during training while retaining the real model and task loss for deployment. Throughout, $P \geq 2$ and we identify $\mathbb{C}^P$ with $\mathbb{R}^{2P}$: gradients, Lipschitz constants and convexity refer to these real coordinates.

C.2 Auxiliary complex parameterization

Let $\Theta = x + iy \in \mathbb{C}^P$ be a complex auxiliary parameter, used only at training time. The real parameters $Z \in \mathbb{R}^P$ define the deployed network and the task loss $\mathcal{L}_{\text{real}}(Z)$. The two are coupled by the equality constraint $\Theta = Z$, $Z \in \mathbb{R}^P$. During training we solve

$$\min_{\Theta \in \mathbb{C}^P,\, Z \in \mathbb{R}^P} \mathcal{L}_{\text{train}}(\Theta) := \widetilde{\mathcal{L}}_{\text{real}}(\Theta) + \lambda_k \mathcal{L}_M(\Theta) \quad \text{subject to } \Theta = Z, \tag{9}$$

with an iteration-dependent weight $\lambda_k \geq 0$ annealed to zero.

The auxiliary loss $\mathcal{L}_M$ is phase-invariant in the sense of Definition 4 below. On the feasible set, $\mathcal{L}_M(Z) = 0$ and the objective is exactly $\mathcal{L}_{\text{real}}(Z)$. There is one task-loss term, evaluated in the complex primal block accompanied by the consensus constraint update, which is a real projection.

C.3 The skew bilinear phase-invariant auxiliary loss

Definition 4 (Phase invariance). *A function $L : \mathbb{C}^P \to \mathbb{R}$ is* phase-invariant *if*

$$L(e^{i\phi}\Theta) = L(\Theta) \qquad \textit{for every } \phi \in \mathbb{R},\ \Theta \in \mathbb{C}^P.$$

Not every phase-invariant function suffices for our construction: the mechanism of Section C.5 requires the auxiliary loss gradient to have a nonzero imaginary component at real Z. Lemma 15 shows that phase invariance forces that imaginary gradient to lie in $Z^\perp \subset \mathbb{R}^P$. A differentiable purely radial choice $f(\|\Theta\|_2)$ has zero imaginary gradient at real points. We therefore use the following skew bilinear form, which provides a nonzero imaginary gradient under an explicit non-degeneracy condition (Lemma 16).

Definition 5 (Skew bilinear auxiliary loss)**.** *Let $M \in \mathbb{R}^{P\times P}$ be skew-symmetric, $M^\top = -M$. Define*

$$\mathcal{L}_M(\Theta) := \Im \langle \Theta, M\Theta\rangle_{\mathbb{C}} = 2x^\top My, \qquad \Theta = x + iy, \quad x, y \in \mathbb{R}^P.$$

The complex inner product conjugates its first argument. The fixed matrix M is part of the algorithm's specification. A nonzero force at $Z_\star$ requires $MZ_\star \neq 0$ and a positive auxiliary loss weight (Theorem 17).

Remark 6 (Choice of M)**.** *A canonical choice is $M = e_1e_2^\top - e_2e_1^\top$, where e_j is the jth coordinate vector. This matrix is skew with $\ker M = \operatorname{span}\{e_1, e_2\}^\perp$. Any $Z_\star$ with a nonzero component in the e_1–e_2 plane satisfies $MZ_\star \neq 0$.*

More generally, for any nonzero $x \in \mathbb{R}^P$, choose a unit $v \perp x$ and set $M = v\hat{x}^\top - \hat{x}v^\top$, where $\hat{x} = x/\|x\|_2$. Then $M^\top = -M$ and $Mx = \|x\|_2 v \neq 0$. Every odd-dimensional real skew matrix is singular, so no such matrix can give a nonzero force at every nonzero point in odd dimension.

As a vartiant for the cyclic shift S on n coordinates, discussed in this appendix, the un-normalized skew-circulant $M = S - S^\top$ has eigenvalues $2i\sin(2\pi j/n)$ for $j = 0, \ldots, n-1$, and $\|M\|_{\mathrm{op}} = 2\max_j |\sin(2\pi j/n)| \leq 2$. Equality holds when n is divisible by four. With such a choice the constant vector lies in its kernel for every n, and this construction is zero when $n = 2$. The current analysis fixes M. Learning or changing it during training would require checking the corresponding assumptions again and is left as future work.

CC-ADMM ITERATIONS

Complex extension of the real loss. For the Θ-primal step we evaluate a complex extension of the real task loss. Given the radial-phase decomposition $\mathcal{L}_{\mathrm{real}}(Z) = \psi(\|Z\|_2) + \chi(Z)$ in Definition 7, define

$$\widetilde{\mathcal{L}}_{\mathrm{real}}(\Theta) = \psi(\|\Theta\|_2) + \chi(\Re\Theta). \tag{10}$$

The extension is consistent with the original loss: $\widetilde{\mathcal{L}}_{\mathrm{real}}(X + 0i) = \mathcal{L}_{\mathrm{real}}(X)$ for every $X \in \mathbb{R}^P$, and has zero imaginary gradient at every real point (Lemma 8).

We split (9) over Θ and Z and enforce $\Theta = Z$ with a scaled dual $U \in \mathbb{C}^P$. The CC-ADMM updates are, with $\rho_k > 0$,

$$\Theta^{k+1} \in \arg\min_{\Theta\in\mathbb{C}^P} \{\mathcal{L}_{\mathrm{train}}(\Theta) + \tfrac{\rho_k}{2}\|\Theta - (Z^k - U^k)\|_2^2\}, \tag{11}$$

$$Z^{k+1} = \arg\min_{Z\in\mathbb{R}^P} \tfrac{\rho_k}{2}\|Z - (\Theta^{k+1} + U^k)\|_2^2 = \Re(\Theta^{k+1} + U^k), \tag{12}$$

$$U^{k+1} = U^k + \Theta^{k+1} - Z^{k+1}. \tag{13}$$

The Z-update is a real projection. The Θ-update is, in general, a non-quadratic minimization in $(\Re\Theta, \Im\Theta)$ because $\widetilde{\mathcal{L}}_{\mathrm{real}}$ is non-convex. Nonetheless, the radial-phase decomposition makes its imaginary gradient vanish at real points, which will be essential in the proof of the Theorem 17 (departure from real stationary point).

These are standard scaled ADMM updates for a fixed objective and penalty. During exploration the implementation varies λ_k and ρ_k while storing U as displayed. Holding the unscaled multiplier fixed instead would require rescaling U when ρ_k changes. The implementation uses a finite number of stochastic primal steps (Section D.1). We also provide the exact-update results below concerning their stated coefficients and assumptions.

C.4 RADIAL-PHASE STRUCTURE AND COMPLEX EXTENSION

Definition 7 (Radial-phase decomposition)**.** *A real loss $\mathcal{L}_{\mathrm{real}} : \mathbb{R}^P \to \mathbb{R}$ admits a* radial-phase decomposition *if it can be written as*

$$\mathcal{L}_{\mathrm{real}}(Z) = \psi(\|Z\|_2) + \chi(Z), \qquad Z \in \mathbb{R}^P,$$

where $\psi : [0, \infty) \to \mathbb{R}$ is C^1 with $\psi'(0) = 0$, and $\chi : \mathbb{R}^P \to \mathbb{R}$ is C^1.

The radial term depends only on the norm and is therefore constant on every sphere centered at the origin. Its complex extension $\psi(\|\Theta\|_2)$, $\Theta \in \mathbb{C}^P$, has the same property and remains constant along the fixed-radius path constructed below.

We additionally require that, for some $R_0 > 0$ and $\varepsilon \geq 0$, $\|\nabla\chi(Z)\|_2 \leq \varepsilon$, whenever $\|Z\|_2 \leq R_0$.

Let's note, that in the definition above only the gradient bound $\|\nabla\chi(Z)\|_2 \leq \varepsilon$ is restricted to the ball. Further, the condition $\psi'(0) = 0$ ensures that the radial term is C^1 at the origin, including in the complex extension viewed as a function of real and imaginary coordinates. We do not require ψ to be strictly convex or to attain its minimum at any particular radius.

The complex extension is the $\widetilde{\mathcal{L}}_{\text{real}}$ from (10), restated here: $\widetilde{\mathcal{L}}_{\text{real}}(\Theta) := \psi(\|\Theta\|_2) + \chi(\Re\Theta)$. The radial part is already invariant under global phase rotation, and χ breaks that invariance only through the real part. Three consequences are immediate for the dynamics of CC-ADMM:

Lemma 8 (Real-restriction and zero imaginary gradient of $\widetilde{\mathcal{L}}_{\text{real}}$ at real points)**.** *Assume $\mathcal{L}_{\text{real}}$ admits the radial–phase decomposition of Definition 7, and let $\widetilde{\mathcal{L}}_{\text{real}}(\Theta) = \psi(\|\Theta\|_2)+\chi(\Re\Theta)$ be its complex extension.*

Writing $\Theta = x + iy$ with $x, y \in \mathbb{R}^P$:

(i) (Real-restriction.) $\widetilde{\mathcal{L}}_{\text{real}}(x, 0) = \mathcal{L}_{\text{real}}(x)$ for every $x \in \mathbb{R}^P$.

(ii) (Real-gradient agreement.) $\partial_x\widetilde{\mathcal{L}}_{\text{real}}(x, 0) = \nabla\mathcal{L}_{\text{real}}(x)$ for every $x \in \mathbb{R}^P$ (at which $\mathcal{L}_{\text{real}}$ is differentiable).

(iii) (Vanishing imaginary gradient at real points.) $\partial_y\widetilde{\mathcal{L}}_{\text{real}}(x, 0) = 0$ for every $x \in \mathbb{R}^P$.

Proof. (i) At $y = 0$, $\|\Theta\|_2 = \|x\|_2$ and $\Re\Theta = x$, so $\widetilde{\mathcal{L}}_{\text{real}}(x, 0) = \psi(\|x\|_2) + \chi(x) = \mathcal{L}_{\text{real}}(x)$. (ii) Differentiate (i) in x. To prove (iii) let's write

$$r = \|\Theta\|_2 = \sqrt{\|x\|_2^2 + \|y\|_2^2}.$$

For $r > 0$, the chain rule gives, for each $j = 1, \ldots, P$,

$$\frac{\partial}{\partial y_j}\psi(r) = \psi'(r)\frac{\partial r}{\partial y_j} = \psi'(r)\frac{y_j}{r}.$$

Since $\chi(x)$ is independent of y, it follows that

$$\nabla_y\widetilde{\mathcal{L}}_{\text{real}}(x, y) = \psi'(r)\frac{y}{r}, \qquad r > 0.$$

This is zero at $y = 0$ whenever $x \neq 0$. At $(x, y) = (0, 0)$, the condition $\psi'(0) = 0$ ensures that the radial term is differentiable with zero gradient:

$$\lim_{\substack{h \to 0 \\ h \neq 0}} \frac{|\psi(\|h\|_2) - \psi(0)|}{\|h\|_2} = 0, \qquad h \in \mathbb{R}^{2P}.$$

Thus, $\nabla_y\widetilde{\mathcal{L}}_{\text{real}}(x, 0) = 0$ for every $x \in \mathbb{R}^P$. □

This lemma is what allows the closed-form-free version of Theorem 17: the imaginary-direction first-order condition of the Θ-primal subproblem can be analyzed without ever solving the (non-quadratic) full subproblem, because the only term contributing to ∂_y at a real iterate with a real proximal center is the auxiliary loss $\lambda_k\mathcal{L}_M$.

Any-radius observation. The construction below uses the sphere through $Z_\star$, of radius $R := \|Z_\star\|_2$. Along this sphere, $\psi(R)$ is constant and exactly cancels its contribution to $\mathcal{L}_{\text{real}}(Z_\star)$. The conclusion therefore holds at every nonzero real strict local minimizer within the assumed ball, regardless of where ψ has its possibly non-unique minima.

Theorem 9 (Bounded-loss complex continuation)**.** *Suppose $\mathcal{L}_{\mathrm{real}}$ admits the radial–phase decomposition of Definition 7 on a ball of radius R_0. Let $Z_\star \in \mathbb{R}^P$ be any strict local minimizer of $\mathcal{L}_{\mathrm{real}}$ with $0 < \|Z_\star\|_2 \le R_0$, and set*

$$R := \|Z_\star\|_2 > 0.$$

Then for any unit vector $v \in \mathbb{R}^P$ orthogonal to $Z_\star$, the continuous path

$$\gamma(t) \;=\; R\big(\,\hat{Z}_\star \cos t \;+\; i\, v \sin t\,\big), \qquad t \in [0,\pi], \quad \hat{Z}_\star := Z_\star/R,$$

satisfies

(i) $\gamma(0) = Z_\star$ *and* $\gamma(\pi) = -Z_\star$

(ii) $\|\gamma(t)\|_2 = R$ *for every* $t \in [0,\pi]$*, hence* $\psi(\|\gamma(t)\|_2) = \psi(R)$ *is constant along the path*

(iii) $\Re\big(\gamma(t)\big) \neq Z_\star$ *for every* $t \in (0,\pi]$

(iv) the two-sided loss bound

$$\mathcal{L}_{\mathrm{real}}(Z_\star) - \varepsilon R(1-\cos t) \le \widetilde{\mathcal{L}}_{\mathrm{real}}(\gamma(t)) \le \mathcal{L}_{\mathrm{real}}(Z_\star) + \varepsilon R(1-\cos t). \tag{14}$$

Since $1-\cos t \le \min\{t, 2\}$ on $[0,\pi]$, these imply the bounds $\mathcal{L}_{\mathrm{real}}(Z_\star) \pm \varepsilon R t$ and an endpoint loss difference of at most $2\varepsilon R$. In particular, if the antipodal point $-Z_\star$ satisfies $\mathcal{L}_{\mathrm{real}}(-Z_\star) < \mathcal{L}_{\mathrm{real}}(Z_\star)$, then γ is a continuous path in $\mathbb{C}^P$ from $Z_\star$ to a strictly lower-loss real point, with the loss along γ never exceeding $\mathcal{L}_{\mathrm{real}}(Z_\star) + \varepsilon\pi R$.

Proof. The real and imaginary coordinates contribute separately to the complex norm. Since $\|\hat{Z}_\star\|_2 = \|v\|_2 = 1$,

$$\|\gamma(t)\|_2^2 \;=\; R^2\big(\|\hat{Z}_\star\|_2^2 \cos^2 t + \|v\|_2^2 \sin^2 t\big) \;=\; R^2(\cos^2 t + \sin^2 t) \;=\; R^2,$$

which proves (ii) and gives $\psi(\|\gamma(t)\|_2) = \psi(R)$ as a constant in t.

The real part is $\Re(\gamma(t)) = R\cos t\, \hat{Z}_\star$, so $\Re(\gamma(0)) = R\hat{Z}_\star = Z_\star$ and $\Re(\gamma(\pi)) = -R\hat{Z}_\star = -Z_\star$, proving (i). For $t \in (0,\pi]$ we have $\cos t < 1$, hence $\Re(\gamma(t)) = R\cos t\, \hat{Z}_\star \neq R\hat{Z}_\star = Z_\star$, proving (iii).

For (iv), the line segment from $Z_\star$ to $\Re(\gamma(t))$ lies inside the ball $\{\|Z\|_2 \le R\} \subseteq \{\|Z\|_2 \le R_0\}$ (using $|\cos t| \le 1$ and $\|Z_\star\|_2 = R \le R_0$), so the gradient bound on χ applies along the whole segment. The mean value theorem gives

$$|\chi(\Re(\gamma(t))) - \chi(Z_\star)| \;\le\; \varepsilon\, \|\Re(\gamma(t)) - Z_\star\|_2 \;=\; \varepsilon R\,(1-\cos t) \;\le\; \varepsilon R\, t.$$

The radial terms cancel exactly:

$$\widetilde{\mathcal{L}}_{\mathrm{real}}(\gamma(t)) - \mathcal{L}_{\mathrm{real}}(Z_\star) = \chi(\Re\gamma(t)) - \chi(Z_\star).$$

The absolute-value estimate proves both inequalities in (14). This proves (iv). The final claim is then immediate from $\gamma(\pi) = -Z_\star$ and the assumed strict inequality on $\mathcal{L}_{\mathrm{real}}(-Z_\star)$. □

The same construction and bounds hold for any nonzero real point within the assumed ball.

Corollary 1.1 (Escape beyond bounded basins, restated)**.** *Under the C^1 assumptions in Section 3, every real strict local minimum with a bounded basin for a fixed real optimization rule admits a complex path to a real point outside that basin, with the explicit finite loss bound.*

Proof. Let $\mathcal{B}$ be the basin of $Z_\star$ for the fixed real optimization rule, with its optimizer-state initialization fixed. Choose $0 \le D < \infty$ such that $\mathcal{B} \subseteq \{Z : \|Z - Z_\star\|_2 \le D\}$. For any $R > D/2$, choose unit vectors $d, v \in \mathbb{R}^P$ with $d^\top v = 0$, and set $c = Z_\star + Rd$.

The continuous path

$$\gamma(t) = c - Rd\cos t + iRv\sin t, \qquad 0 \le t \le \pi,$$

satisfies $\gamma(0) = Z_\star$ and $\gamma(\pi) = Z_\star + 2Rd \notin \mathcal{B}$. Indeed, its endpoint is at distance $2R > D$ from $Z_\star$. Its real projection is outside the basin whenever $R(1-\cos t) > D$.

Put $S = \|c\|_2 + R$ and define

$$a_S := \max_{0 \le r \le S} |\psi'(r)|, \qquad b_S := \max_{\|Z\|_2 \le S} \|\nabla \chi(Z)\|_2.$$

Both constants are finite: ψ' is continuous on $[0, S]$, and $\nabla\chi$ is continuous on the closed real ball of radius S and these domains are compact. To identify the region on which these bounds are used, recall that for real vectors a, b, the complex Euclidean norm satisfies $\|a + ib\|_2^2 = \|a\|_2^2 + \|b\|_2^2$.

Since d and v are unit vectors,

$$\|\gamma(t) - c\|_2^2 = R^2\|d\|_2^2 \cos^2 t + R^2\|v\|_2^2 \sin^2 t = R^2.$$

Thus the path lies on the complex sphere of radius R centered at c. The triangle inequality gives

$$\|\gamma(t)\|_2 \le \|c\|_2 + \|\gamma(t) - c\|_2 = S, \qquad \|\Re\gamma(t)\|_2 \le \|\gamma(t)\|_2 \le S.$$

These are the closed balls of radius S centered at the origin in $\mathbb{C}^P$ and $\mathbb{R}^P$, respectively. The starting point satisfies $\|Z_\star\|_2 = \|\gamma(0)\|_2 \le S$. By convexity of the real ball, the entire line segment from $Z_\star$ to $\Re\gamma(t)$ lies in that ball, so the bound b_S applies throughout the segment.

To compute the displacement from the starting point, substitute $c = Z_\star + Rd$:

$$\gamma(t) - Z_\star = Rd(1 - \cos t) + iRv \sin t.$$

Taking the squared complex norm and using the half-angle identity yields

$$\begin{aligned}\|\gamma(t) - Z_\star\|_2^2 &= R^2\left[(1 - \cos t)^2 + \sin^2 t\right] \\ &= 2R^2(1 - \cos t) = 4R^2 \sin^2(t/2).\end{aligned}$$

Because $0 \le t \le \pi$, we have $\sin(t/2) \ge 0$, and hence $\|\gamma(t) - Z_\star\|_2 = 2R\sin(t/2)$. The real displacement is $\Re\gamma(t) - Z_\star = Rd(1 - \cos t)$ and since $\|d\|_2 = 1$ and $1 - \cos t \ge 0$, its norm is $\|\Re\gamma(t) - Z_\star\|_2 = R(1 - \cos t)$.

Applying the scalar mean-value bound to ψ, the gradient bound to χ on the real line segment, and the reverse triangle inequality, we obtain, for the original complex extension,

$$\begin{aligned}\left|\widetilde{\mathcal{L}}_{\text{real}}(\gamma(t)) - \mathcal{L}_{\text{real}}(Z_\star)\right| &\le a_S\left|\|\gamma(t)\|_2 - \|Z_\star\|_2\right| + b_S\|\Re\gamma(t) - Z_\star\|_2 \\ &\le 2a_S R\sin(t/2) + b_S R(1 - \cos t) \\ &\le 2R(a_S + b_S).\end{aligned} \tag{15}$$

Thus, every finite $R > D/2$ supplies a path to a real point outside the bounded basin, without requiring that endpoint to have lower loss. The constants depend on the chosen radius and the low-loss bound depends on it. Unlike the origin-centered path in Theorem 1, this shifted path needs not keep the original radial term constant, which is why its variation appears explicitly in (15). □

We note that Theorem 9 is a geometric statement in the sense that it constructs an explicit continuous path of bounded extra loss between $Z_\star$ and its antipodal point $-Z_\star$. It does not by itself describe how an algorithm would explore γ, and as stated it asserts nothing about lower-loss real points whose magnitude differs from $\|Z_\star\|_2$. The next corollary removes this last restriction by re-centering the radial-phase decomposition.

The construction of Theorem 9 relies on $\|\gamma(t)\|_2$ being constant on the sphere through $Z_\star$. Two distinct real points $Z_\star$ and $\hat{Z}$ in $\mathbb{R}^P$ ($P \ge 2$) need not lie on a common sphere centered at the origin (their norms can differ), but they always lie on a common sphere centered at their midpoint $c = (Z_\star + \hat{Z})/2$. The natural extension of the theorem therefore replaces the radial-phase decomposition centered at 0 by one centered at c. The proof reduces to Theorem 9 after the change of variable $\tilde{Z} = Z - c$, since this change of variable preserves all gradients of $\mathcal{L}_{\text{real}}$ and all critical-point structure.

Definition 10 (c-centered radial-phase decomposition)**.** *For $c \in \mathbb{R}^P$, suppose $\mathcal{L}_{\text{real}}(Z) = \psi^{(c)}(\|Z - c\|_2) + \chi^{(c)}(Z)$, where the functions are C^1, $(\psi^{(c)})'(0) = 0$, and $\|\nabla\chi^{(c)}(Z)\|_2 \le \varepsilon^{(c)}$ on $\|Z - c\|_2 \le R$. Its centered complex extension is $\widetilde{\mathcal{L}}^{(c)}(\Theta) = \psi^{(c)}(\|\Theta - c\|_2) + \chi^{(c)}(\Re\Theta)$.*

Remark 11 (Centering an ℓ_2-regularized loss)**.** *For $\mathcal{L}_{\mathrm{real}}(Z) = \mu\|Z\|_2^2/2 + \chi(Z)$, the exact centered decomposition uses*

$$\psi^{(c)}(r) = \mu r^2/2, \qquad \chi^{(c)}(Z) = \chi(Z) + \mu\langle Z, c\rangle - \mu\|c\|_2^2/2.$$

For this quadratic radial term, re-centering preserves the original complex extension exactly: for $\Theta = x + iy$ and real c,

$$\widetilde{\mathcal{L}}^{(c)}(x+iy) = \frac{\mu}{2}\big(\|x-c\|_2^2 + \|y\|_2^2\big) + \chi^{(c)}(x) = \frac{\mu}{2}\big(\|x\|_2^2 + \|y\|_2^2\big) + \chi(x) = \widetilde{\mathcal{L}}_{\mathrm{real}}(x+iy).$$

Thus $\nabla\chi^{(c)} = \nabla\chi + \mu c$ and $\varepsilon^{(c)} \le \varepsilon + \mu\|c\|_2$ wherever the original residual bound holds. The required domain must contain the centered ball. An origin-centered bound known only on a smaller ball is insufficient. An analogous identity applies to an ℓ_1 objective only after its Hadamard-factorization reformulation in that factor space, as described in the main body of the paper. We note though, that centering can make the loss band large possibly increasing a barrier along the path.

Corollary 12 (Origin-shifted path existence)**.** *Let $Z_\star \in \mathbb{R}^P$ be a strict local minimizer of $\mathcal{L}_{\mathrm{real}}$ and let $\hat{Z} \in \mathbb{R}^P$ satisfy $Z_\star \neq \hat{Z}$ and $\mathcal{L}_{\mathrm{real}}(\hat{Z}) \le \mathcal{L}_{\mathrm{real}}(Z_\star)$.*

Define

$$c := \tfrac{1}{2}(Z_\star + \hat{Z}), \qquad R := \tfrac{1}{2}\|Z_\star - \hat{Z}\|_2 > 0, \qquad \hat{Z}_\star := \frac{Z_\star - c}{R} \in \mathbb{R}^P \quad (\|\hat{Z}_\star\|_2 = 1).$$

Suppose $\mathcal{L}_{\mathrm{real}}$ admits a c-centered radial-phase decomposition of radius R (Definition 10) with parameters $(\psi^{(c)}, \chi^{(c)}, \varepsilon^{(c)})$. Then for any unit vector $v \in \mathbb{R}^P$ orthogonal to $\hat{Z}_\star$, the continuous path

$$\gamma(t) = c + R\big(\hat{Z}_\star \cos t + i\, v \sin t\big), \qquad t \in [0, \pi],$$

satisfies

- *(i)* $\gamma(0) = Z_\star$ *and* $\gamma(\pi) = \hat{Z}$
- *(ii)* $\|\gamma(t) - c\|_2 = R$ *for every* $t \in [0, \pi]$, *hence* $\psi^{(c)}(\|\gamma(t) - c\|_2) = \psi^{(c)}(R)$ *is constant along the path*
- *(iii)* $\Re(\gamma(t)) \neq Z_\star$ *for every* $t \in (0, \pi]$
- *(iv)* $\big|\widetilde{\mathcal{L}}^{(c)}(\gamma(t)) - \mathcal{L}_{\mathrm{real}}(Z_\star)\big| \le \varepsilon^{(c)} R(1 - \cos t) \le \varepsilon^{(c)} Rt$.

In particular, when $\mathcal{L}_{\mathrm{real}}(\hat{Z}) < \mathcal{L}_{\mathrm{real}}(Z_\star)$ the path γ connects $Z_\star$ to a strictly lower-loss real point, and the loss along γ never exceeds $\mathcal{L}_{\mathrm{real}}(Z_\star) + \varepsilon^{(c)}\pi R$.

Proof. Define the translated loss $\mathcal{L}^\circ(\tilde{Z}) := \mathcal{L}_{\mathrm{real}}(\tilde{Z} + c)$. Because translation by c is a bijection of $\mathbb{R}^P$ onto itself that commutes with differentiation, $\nabla\mathcal{L}^\circ(\tilde{Z}) = \nabla\mathcal{L}_{\mathrm{real}}(\tilde{Z} + c)$ for every $\tilde{Z}$. In particular, $\tilde{Z}_\star := Z_\star - c$ is a strict local minimizer of $\mathcal{L}^\circ$ (since $Z_\star$ is one of $\mathcal{L}_{\mathrm{real}}$), and $\|\tilde{Z}_\star\|_2 = R$.

Inserting the assumed c-centered decomposition into the definition of $\mathcal{L}^\circ$ gives, for every $\tilde{Z} \in \mathbb{R}^P$,

$$\mathcal{L}^\circ(\tilde{Z}) = \psi^{(c)}(\|\tilde{Z}\|_2) + \chi^{(c)}(\tilde{Z} + c) =: \tilde{\psi}(\|\tilde{Z}\|_2) + \tilde{\chi}(\tilde{Z}),$$

with $\tilde{\psi} := \psi^{(c)}$ and $\tilde{\chi}(\tilde{Z}) := \chi^{(c)}(\tilde{Z} + c)$.

The chain rule yields $\nabla\tilde{\chi}(\tilde{Z}) = \nabla\chi^{(c)}(\tilde{Z} + c)$, so $\|\nabla\tilde{\chi}(\tilde{Z})\|_2 = \|\nabla\chi^{(c)}(\tilde{Z} + c)\|_2 \le \varepsilon^{(c)}$ on $\{\|\tilde{Z}\|_2 \le R\}$, the re-labeled image of $\{\|Z - c\|_2 \le R\}$ under translation. Hence, $\mathcal{L}^\circ$ satisfies Definition 7 on a ball of radius $R_0 := R$ around 0, with parameters $(\tilde{\psi}, \tilde{\chi}, \varepsilon^{(c)})$.

Because $\hat{Z}_\star$ is a unit vector and $v \perp \hat{Z}_\star$ with $\|v\|_2 = 1$, all hypotheses of Theorem 9 hold for the translated loss $\mathcal{L}^\circ$ and minimizer $\tilde{Z}_\star = R\,\hat{Z}_\star$. Applying that theorem yields the path

$$\tilde{\gamma}(t) = R\big(\hat{Z}_\star \cos t + i\, v \sin t\big), \qquad t \in [0, \pi],$$

satisfying $\tilde{\gamma}(0) = \tilde{Z}_\star$, $\tilde{\gamma}(\pi) = -\tilde{Z}_\star$, $\|\tilde{\gamma}(t)\|_2 = R$, $\Re\tilde{\gamma}(t) \neq \tilde{Z}_\star$ for $t \in (0, \pi]$, and the bound $|\widetilde{\mathcal{L}}^\circ(\tilde{\gamma}(t)) - \mathcal{L}^\circ(\tilde{Z}_\star)| \le \varepsilon^{(c)} R(1 - \cos t) \le \varepsilon^{(c)} Rt$, where $\widetilde{\mathcal{L}}^\circ(\tilde{\Theta}) := \tilde{\psi}(\|\tilde{\Theta}\|_2) + \tilde{\chi}(\Re\tilde{\Theta})$.

Next we translate back to original space. Define $\gamma(t) := c + \tilde{\gamma}(t)$. Since the translation is by a real constant, $\Im\gamma(t) = \Im\tilde{\gamma}(t)$ and $\Re\gamma(t) = c + \Re\tilde{\gamma}(t)$. Therefore:

- $\gamma(0) = c + \tilde{Z}_\star = Z_\star$ and $\gamma(\pi) = c - \tilde{Z}_\star = c - (Z_\star - c) = 2c - Z_\star = \hat{Z}$, proving (i)
- $\|\gamma(t) - c\|_2 = \|\tilde{\gamma}(t)\|_2 = R$, proving (ii)
- $\Re\gamma(t) - Z_\star = (\Re\tilde{\gamma}(t) - \tilde{Z}_\star) \neq 0$ for $t \in (0, \pi]$, proving (iii)
- by the definitions of $\widetilde{\mathcal{L}}^{(c)}$ and $\widetilde{\mathcal{L}}^{\circ}$,

$$\begin{aligned}\widetilde{\mathcal{L}}^{(c)}(\gamma(t)) &= \psi^{(c)}(\|\gamma(t) - c\|_2) + \chi^{(c)}(\Re\gamma(t)) \\ &= \tilde{\psi}(\|\tilde{\gamma}(t)\|_2) + \tilde{\chi}(\Re\tilde{\gamma}(t)) = \widetilde{\mathcal{L}}^{\circ}(\tilde{\gamma}(t)),\end{aligned}$$

  and $\mathcal{L}^{\circ}(\tilde{Z}_\star) = \mathcal{L}_{\text{real}}(Z_\star)$.

  The two-sided bound from Theorem 9 therefore gives $|\widetilde{\mathcal{L}}^{(c)}(\gamma(t)) - \mathcal{L}_{\text{real}}(Z_\star)| \leq \varepsilon^{(c)} R(1 - \cos t) \leq \varepsilon^{(c)} Rt$, proving (iv).

The final claim follows from $\gamma(\pi) = \hat{Z}$ and the assumed strict inequality on $\mathcal{L}_{\text{real}}(\hat{Z})$. At $t = \pi$, the endpoint improvement is at most $2\varepsilon^{(c)} R$. □

Remark 13. *A magnitude-threshold activation such as the modReLU is radial in its own scalar pre-activation. We note, that its threshold changes the active radius, and a preceding bias translates that pre-activation. While valid locally on a infinitesimally small open neighborhood, it does not establish an approximate radial symmetry of an entire network's loss in its full parameter space, or a small residual after an arbitrary parameter-space translation. Such a decomposition and its bound must be checked for the loss and domain in question.*

Remark 14 (A consistently centered auxiliary loss and extension)**.** *To obtain the translated version of the iteration, replace both the extension and auxiliary loss by their centered versions: $\mathcal{L}^{(c)}_{\text{train}}(\Theta) = \widetilde{\mathcal{L}}^{(c)}(\Theta) + \lambda_k \mathcal{L}_M(\Theta - c)$, where $\mathcal{L}_M(\Theta - c) = 2(x-c)^\top My$. Under $(\tilde{\Theta}, \tilde{Z}, \tilde{U}) = (\Theta - c, Z - c, U)$, the squared penalty and projection translate exactly and the dual residual is unchanged. Thus the translated iteration has the same form as* (11)–(13)*, for the translated loss.*

The departure theorem applies with $M(Z_\star - c) \neq 0$ and its own smoothness and existence assumptions. This conjugacy does not show that its iterates follow the path of Corollary 12.

C.5 Phase invariance and deterministic imaginary motion

Lemma 15 (Phase-invariance identity)**.** *Let $L : \mathbb{C}^P \to \mathbb{R}$ be C^1 and phase-invariant (Def. 4). Writing $\Theta = x + iy$ with $x, y \in \mathbb{R}^P$, the identity*

$$\big\langle \partial_y L(x, 0),\, x \big\rangle = 0 \qquad \textit{holds for every } x \in \mathbb{R}^P. \tag{16}$$

Equivalently, $\partial_y L(x, 0) \in x^\perp \subset \mathbb{R}^P$.

Proof. Phase invariance asserts $L(x'(\phi), y'(\phi)) = L(x, y)$ with $x'(\phi) = x\cos\phi - y\sin\phi$ and $y'(\phi) = x\sin\phi + y\cos\phi$. By invariance, differentiating both sides with respect to ϕ at $\phi = 0$ gives zero. Evaluating at $y = 0$ gives $\partial_\phi x'|_{\phi=0, y=0} = 0$ and $\partial_\phi y'|_{\phi=0, y=0} = x$, whence by the chain rule $0 = \langle \partial_y L(x, 0), x\rangle$. □

Under the standing assumption $P \geq 2$, the hyperplane $x^\perp \subset \mathbb{R}^P$ is non-trivial whenever $x \neq 0$, so (16) permits, but does not require, a non-zero imaginary gradient. The skew bilinear auxiliary loss $\mathcal{L}_M$ realizes this non-zero gradient when $Mx \neq 0$.

Lemma 16 (Properties of the skew auxiliary loss)**.** *For the $\mathcal{L}_M$ of Definition 5,*

(i) $\mathcal{L}_M$ is real-valued and phase-invariant;

(ii) $\partial_x \mathcal{L}_M = 2My$ and $\partial_y \mathcal{L}_M = -2Mx$, so $\partial_y \mathcal{L}_M(x, 0) \neq 0$ exactly when $Mx \neq 0$;

(iii) its real-coordinate Hessian is $2\left(\begin{smallmatrix} 0 & M \\ -M & 0 \end{smallmatrix}\right)$, with operator norm $2\|M\|_{\text{op}}$.

Proof. (i) Write the complex quadratic form as $\Theta^H M\Theta = (x-iy)^\top M(x+iy) = x^\top Mx + y^\top My + i\big(x^\top My - y^\top Mx\big)$. For skew M, $x^\top Mx = 0$ and $y^\top My = 0$, because $u^\top Mu = (u^\top Mu)^\top = u^\top M^\top u = -u^\top Mu$.

Similarly $y^\top Mx = -x^\top My$. Hence, $\Theta^H M\Theta = 2i\, x^\top My$, so $\mathcal{L}_M(\Theta) = \Im(\Theta^H M\Theta) = 2x^\top My \in \mathbb{R}$. Phase invariance is immediate from $(e^{i\phi}\Theta)^H M(e^{i\phi}\Theta) = e^{-i\phi}e^{i\phi}\Theta^H M\Theta = \Theta^H M\Theta$.

(ii) Direct differentiation of $\mathcal{L}_M = 2x^\top My$ gives $\partial_x \mathcal{L}_M = 2My$ and $\partial_y \mathcal{L}_M = 2M^\top x$. For skew M, $2M^\top x = -2Mx$. The inner product $\langle \partial_y \mathcal{L}_M(x,0), x\rangle = -2x^\top Mx = 0$ by skewness, consistent with Lemma 15. $\partial_y \mathcal{L}_M(x,0) = 0$ if and only if $Mx = 0$ if and only if $x \in \ker M$.

(iii) Differentiating once more gives the real-coordinate Hessian $2\left(\begin{smallmatrix} 0 & M \\ -M & 0 \end{smallmatrix}\right)$. Its square is block diagonal with both blocks $-4M^2 = 4M^\top M$, proving the operator-norm formula. □

Theorem 17 (Deterministic departure at a real stationary point)**.** *Let $Z_\star$ be a real local minimizer and suppose $\Theta^k = Z^k = Z_\star$, $U^k = 0$, $MZ_\star \neq 0$ and $\lambda_k > 0$. Assume the extension* (10) *has a globally $\widetilde{L}$-Lipschitz gradient. If*

$$\mu_k := \rho_k - \widetilde{L} - 2\lambda_k \|M\|_{\mathrm{op}} > 0, \tag{17}$$

then the primal subproblem has a unique global minimizer Θ^{k+1} and $\Im\Theta^{k+1} \neq 0$. Moreover,

$$\|\Theta^{k+1} - Z_\star\|_2 \leq \frac{2\lambda_k \|MZ_\star\|_2}{\mu_k}.$$

A local version holds if the extension gradient is $\widetilde{L}$-Lipschitz on a neighbourhood of the closed complex ball $\overline{B}_\delta(Z_\star)$, $0 < \delta < \|Z_\star\|_2$, and $2\lambda_k\|MZ_\star\|_2 < \mu_k\delta$: there is a unique minimizer on that ball, it is interior and non-real, and it is a local minimizer of the unconstrained subproblem.

Proof. Write the primal objective as $Q(\Theta) = \widetilde{\mathcal{L}}_{\mathrm{real}}(\Theta) + \lambda_k \mathcal{L}_M(\Theta) + \rho_k\|\Theta - Z_\star\|_2^2/2$. *Step 1: existence and uniqueness.* A differentiable function with $\widetilde{L}$-Lipschitz gradient is $\widetilde{L}$-weakly convex. Lemma 16 therefore makes Q μ_k-strongly convex. Under the global assumption, it is coercive and has a unique minimizer. Since $\nabla Q(Z_\star) = (0, -2\lambda_k MZ_\star)$, strong monotonicity of ∇Q gives the displayed distance bound.

For the local version, continuity gives a minimizer on the closed ball. At any boundary point w,

$$\langle \nabla Q(w), w - Z_\star\rangle \geq \mu_k\delta^2 - 2\lambda_k\|MZ_\star\|_2\delta > 0.$$

Moving from w toward the center decreases Q, so a boundary point cannot minimize it. The minimizer is interior and unique by the strong convexity on the ball and the same distance estimate applies.

Step 2: the first-order conditions at $y = 0$ force $x = Z_\star$. Suppose for contradiction that $y^{k+1} = 0$. The first-order condition $\partial_y Q(x^{k+1}, y^{k+1}) = 0$ at $y^{k+1} = 0$ reads, in real coordinates,

$$\partial_y \widetilde{\mathcal{L}}_{\mathrm{real}}(x^{k+1}, 0) \;+\; 2\lambda_k\, M^\top x^{k+1} \;+\; \rho_k \cdot 0 \;=\; 0.$$

By Lemma 8(iii) the first term is identically zero, and using $M^\top = -M$ and $\lambda_k > 0$ the equation reduces to

$$M\, x^{k+1} \;=\; 0, \qquad \text{i.e.} \qquad x^{k+1} \in \ker M. \tag{18}$$

At the same time, the first-order condition $\partial_x Q(x^{k+1}, 0) = 0$ provides

$$\nabla \mathcal{L}_{\mathrm{real}}(x^{k+1}) \;+\; 2\lambda_k\, M \cdot 0 \;+\; \rho_k\,(x^{k+1} - Z_\star) \;=\; 0,$$

where we used Lemma 8(ii) for the x-gradient of the extension at $y = 0$. Rearranging,

$$x^{k+1} - Z_\star \;=\; -\frac{1}{\rho_k}\,\nabla \mathcal{L}_{\mathrm{real}}(x^{k+1}). \tag{19}$$

Since $\nabla \mathcal{L}_{\mathrm{real}}(Z_\star) = 0$ ($Z_\star$ is a local minimizer) and $\nabla \mathcal{L}_{\mathrm{real}}$ is $\widetilde{L}$-Lipschitz on the relevant real domain, taking norms in (19) gives

$$\|x^{k+1} - Z_\star\|_2 \;\leq\; \frac{\widetilde{L}}{\rho_k}\,\|x^{k+1} - Z_\star\|_2.$$

By (17) we have $\widetilde{L}/\rho_k < 1$, so this inequality forces $\|x^{k+1} - Z_\star\|_2 = 0$, i.e. $x^{k+1} = Z_\star$.

Step 3: contradiction. Combining (18) with $x^{k+1} = Z_\star$ yields $MZ_\star = 0$, contradicting the standing hypothesis $MZ_\star \neq 0$. Therefore, $y^{k+1} \neq 0$, and $\Im(\Theta^{k+1}) \neq 0$. □

Remark (exploration and adaptive restarts). Complex exploration need not produce an immediately lower-loss real point: its purpose is to enable exploration of other basins before real-valued optimization continues. Theorem 17 establishes departure from the real subspace and Corollary 1.1 establishes existence of a path beyond a bounded basin. Whether the algorithm reaches another basin depends on its trajectory and the ADMM algorithm's iterative steps during the optimization.

If real-valued refinement returns near the starting stationary point, as assessed by parameter proximity together with training loss, a possible extension is to restart exploration with a different skew-symmetric matrix $\widehat{M}$ satisfying $\widehat{M}Z_\star \neq 0$ and/or learning rate. These two factors define a "slack energy/relaxation" within the ADMM procedure that can absorb a possible increase of the task loss needed to surpass the loss barrier along the escape paths. In fact, the freedom of choice of the skew-symmetric matrix M defining the auxiliary provides an interesting ground for deeper research possibilities outlined in the following remark.

Remark 18 (Auxiliary form strength and future work)**.** *The auxiliary loss initiates complex exploration without requiring stochastic noise. At a real stationary initialization $\Theta = Z = Z_\star$, $U = 0$, a plain gradient step gives $y^+ = 2\eta\lambda_k M Z_\star$. Thus departure occurs when $\lambda_k > 0$ and $MZ_\star \neq 0$. Subsequent imaginary and real updates are coupled through $\nabla_x \mathcal{L}_M = 2My$.*

During exploration, a decrease in the auxiliary loss can compensate for an increase in the task loss. More precisely, consider a primal update from $\Theta = Z_\star$ with $Z = Z_\star$ and $U = 0$, holding λ_k and ρ_k fixed. Any resulting Θ^+ that decreases the primal objective satisfies

$$\widetilde{\mathcal{L}}_{\mathrm{real}}(\Theta^+) - \mathcal{L}_{\mathrm{real}}(Z_\star) \leq -\lambda_k \mathcal{L}_M(\Theta^+) - \frac{\rho_k}{2}\|\Theta^+ - Z_\star\|_2^2.$$

A sufficiently negative auxiliary loss contribution can therefore offset both a task-loss increase and the consensus penalty. This inequality is conditional on primal-objective descent and while it does not assert descent at every stochastic update it opens the future research directions to be explored.

The magnitude of this contribution depends on the spectrum and orientation of M. A real skew-symmetric matrix has eigenvalues $\pm i\sigma_j$, with possible zeros, and $\sigma_{\max} = \|M\|_{\mathrm{op}}$. Since $-iM$ is Hermitian and $\mathcal{L}_M(\Theta) = \Theta^H(-iM)\Theta$, for any fixed norm $r > 0$,

$$\min_{\|\Theta\|_2 = r} \lambda_k \mathcal{L}_M(\Theta) = -\lambda_k \sigma_{\max} r^2.$$

Thus, larger eigenvalue magnitudes permit a larger negative auxiliary loss contribution in suitable directions. They do not ensure that the iterates reach those directions. Along a particular path, the auxiliary bilinear term may vanish or change sign, so merely increasing its magnitude does not necessarily make that path a descent trajectory. Moreover, scaling M is equivalent to scaling λ_k, which is readily done by tuning the learning rate in the current setup.

In addition to the auxiliary term, the dual records consensus constraint violations and changes the subsequent primal updates. Its contribution can oppose or balance the auxiliary loss force, so dual accumulation alone does not establish continued escape

These observations motivate future work on selecting the orientation of M from local loss geometry, while controlling its norm and coordinating the auxiliary and penalty schedules. The objective would be to direct exploration toward another basin and establish sufficient conditions for crossing its boundary before the final real projection. Such a result would directly connect the update dynamics to the bounded-loss paths constructed here.

Proposition 19 (Projection and dual accumulation)**.** *For* (12)–(13)*,*

$$Z^{k+1} = \Re\Theta^{k+1} + \Re U^k, \qquad \Re U^{k+1} = 0, \qquad \Im U^{k+1} = \Im U^k + \Im\Theta^{k+1}.$$

If $U^0 = 0$, then $\Im U^{k+1} = \sum_{j=1}^{k+1} \Im\Theta^j$. Under the departure theorem at $k = 0$, $\Im U^1 \neq 0$.

Proof. For $W = W_R + iW_I$ and real Z, $\|Z - W\|_2^2 = \|Z - W_R\|_2^2 + \|W_I\|_2^2$, whose unique minimizer is W_R. Substitute this projection into the real and imaginary parts of the dual update, and telescope the imaginary recursion. □

C.6 STATIONARITY AFTER A FINITE EXPLORATION PHASE

After annealing and the finite real projection, convergence reduces to that of the optimizer used on $\mathcal{L}_{\text{real}}$. The following restatement gives Theorem 3 with exactly the same assumptions and conclusions, followed by its full proof. The separate Proposition 20 covers exact proximal and deterministic gradient-descent updates.

Theorem 3 (Convergence under annealing and real projection, restated)**.** *Assume $\mathcal{L}_{\text{real}}$ is coercive ($\mathcal{L}_{\text{real}}(Z) \to \infty$ as $\|Z\|_2 \to \infty$) and bounded below, with globally Lipschitz gradient. After a finite exploration phase ending at a fixed iteration k_0, set $\lambda_k = 0$, project the model onto $\mathbb{R}^P$, and disable consensus and dual updates. For $k \geq k_0$, use real SGD, $Z^{k+1} = Z^k - \eta_k g_k$, where $g_k \in \mathbb{R}^P$ is a stochastic estimate of $\nabla\mathcal{L}_{\text{real}}(Z^k)$. Assume these estimates are conditionally unbiased with uniformly bounded conditional noise variance, given the exploration history and all information available before sampling g_k. The deterministic learning rates satisfy $\eta_k > 0$, $\sum_{k\geq k_0} \eta_k = \infty$, $\sum_{k\geq k_0} \eta_k^2 < \infty$. Then, with probability one, $\mathcal{L}_{\text{real}}(Z^k)$ converges to a finite value, the real iterates are bounded, $\|\nabla\mathcal{L}_{\text{real}}(Z^k)\|_2 \to 0$, and every limit point Z^∞ satisfies $\nabla\mathcal{L}_{\text{real}}(Z^\infty) = 0$.*

Proof. Let $\mathcal{F}_k$ be the history of the algorithm before sampling g_k, including the finite exploration phase. Then Z^k is $\mathcal{F}_k$-measurable, and the theorem's conditional gradient assumptions mean that, almost surely,

$$\mathbb{E}[g_k \mid \mathcal{F}_k] = \nabla\mathcal{L}_{\text{real}}(Z^k), \tag{20}$$

$$\mathbb{E}[\|g_k - \nabla\mathcal{L}_{\text{real}}(Z^k)\|_2^2 \mid \mathcal{F}_k] \leq \sigma^2 < \infty \tag{21}$$

for a fixed constant σ^2.

Once the auxiliary term has vanished and the model has been projected to real parameters, the remaining iterations are ordinary SGD on $\mathcal{L}_{\text{real}}$.

We can write $s_k = -\nabla\mathcal{L}_{\text{real}}(Z^k)$ and $w_k = \nabla\mathcal{L}_{\text{real}}(Z^k) - g_k$, so that $Z^{k+1} = Z^k + \eta_k(s_k + w_k)$.

Now, we are ready to apply Proposition 3 of Bertsekas & Tsitsiklis (2000). Its descent-direction conditions hold because $-\langle\nabla\mathcal{L}_{\text{real}}(Z^k), s_k\rangle = \|\nabla\mathcal{L}_{\text{real}}(Z^k)\|_2^2$ and $\|s_k\|_2 \leq 1 + \|\nabla\mathcal{L}_{\text{real}}(Z^k)\|_2$.

Equations (20)–(21) give $\mathbb{E}[w_k \mid \mathcal{F}_k] = 0$ and $\mathbb{E}[\|w_k\|_2^2 \mid \mathcal{F}_k] \leq A(1 + \|\nabla\mathcal{L}_{\text{real}}(Z^k)\|_2^2)$ with $A = \max\{1, \sigma^2\}$, as required.

Its conclusion is either loss divergence to $-\infty$ or convergence of the loss to a finite value and vanishing gradients, with probability one.

The assumed lower bound on the loss excludes the first alternative. On each sample path where the loss converges, its values are bounded above. Coercivity therefore bounds the iterates. If $Z^{k_j} \to Z^\infty$, continuity of the gradient gives $\nabla\mathcal{L}_{\text{real}}(Z^\infty) = \lim_j \nabla\mathcal{L}_{\text{real}}(Z^{k_j}) = 0$.

Thus, the finite exploration phase, followed by real projection, provides a valid starting point for the SGD theory above (absolute convergence w.r.t. starting point) resulting in the stationary point limit, just shown, with the finite exploration history included in $\mathcal{F}_{k_0}$. □

According to the Theorem 3, a sufficient learning-rate schedule is for instance $\eta_{k_0+j} = a/(j+1)^q$ with $a > 0$ and $1/2 < q \leq 1$. We also further note that the theorem concerns convergence to a stationary point. It does not assert convergence of the full parameter sequence to a single point or a global minimum. Setting $g_k = \nabla\mathcal{L}_{\text{real}}(Z^k)$ also covers deterministic gradient descent with the stated learning rates. In addition, the separate result below covers exact proximal and constant-step gradient-descent updates.

Proposition 20 (Exact and gradient-descent updates after real projection)**.** *Assume $\mathcal{L}_{\text{real}}$ is coercive and bounded below and $\widetilde{\mathcal{L}}_{\text{real}}$ of* (10) *has a globally $\widetilde{L}$-Lipschitz gradient, with complex variables identified with $\mathbb{R}^{2P}$.*

After a finite exploration phase, set $\Theta^{k_0} = Z^{k_0} \in \mathbb{R}^P$ and $U^{k_0} = 0$. For all $k \geq k_0$, set $\lambda_k = 0$ and $\rho_k = \rho > \widetilde{L}$. Using (12)-(13) *we have either:*

(a) the exact primal update (11) *or*

(b) for a fixed $\tau \geq \widetilde{L}$, the prox-linear update

$$\Theta^{k+1} = \arg\min_{\Theta} \left\{ \left\langle \nabla \widetilde{\mathcal{L}}_{\text{real}}(\Theta^k), \Theta - \Theta^k \right\rangle_{\mathbb{R}^{2P}} + \tfrac{\tau}{2}\|\Theta - \Theta^k\|_2^2 + \tfrac{\rho}{2}\|\Theta - (Z^k - U^k)\|_2^2 \right\}. \tag{22}$$

Then $\Theta^k = Z^k \in \mathbb{R}^P$ and $U^k = 0$ for every $k \geq k_0$. The iterates are bounded and satisfy $\|\nabla \mathcal{L}_{\text{real}}(Z^k)\|_2 \to 0$. Every limit point is stationary for the original real loss.

Proof. We first show that the updates remain real and reduce to proximal point or gradient descent, then establish descent and stationarity. These real updates are special cases of the proximal-gradient iteration discussed by Li & Pong (2015, Section 4); we give the argument for the assumptions stated here.

The extension in (10) satisfies

$$\widetilde{\mathcal{L}}_{\text{real}}(x + iy) = \widetilde{\mathcal{L}}_{\text{real}}(x - iy), \qquad \widetilde{\mathcal{L}}_{\text{real}}(x) = \mathcal{L}_{\text{real}}(x).$$

Consequently, its imaginary gradient vanishes at every real point, and its real gradient there equals $\nabla \mathcal{L}_{\text{real}}$. The real loss also inherits the gradient Lipschitz constant $\widetilde{L}$.

(a) Exact primal update. Suppose $\Theta^k = Z^k \in \mathbb{R}^P$ and $U^k = 0$.

Since the extension has a $\widetilde{L}$-Lipschitz gradient and $\rho > \widetilde{L}$, the primal objective

$$\widetilde{\mathcal{L}}_{\text{real}}(\Theta) + \frac{\rho}{2}\|\Theta - Z^k\|_2^2$$

is $(\rho - \widetilde{L})$-strongly convex and coercive, so it has a unique minimizer. The objective is unchanged by complex conjugation. Its minimizer and its conjugate must therefore coincide, making the minimizer real. Equations (12)-(13) then give $Z^{k+1} = \Theta^{k+1}$ and $U^{k+1} = 0$. Induction proves these identities for every $k \geq k_0$.

The update is thus exactly

$$Z^{k+1} = \arg\min_{z \in \mathbb{R}^P} \left\{ \mathcal{L}_{\text{real}}(z) + \frac{\rho}{2}\|z - Z^k\|_2^2 \right\}.$$

Comparing with $z = Z^k$ gives

$$\mathcal{L}_{\text{real}}(Z^{k+1}) + \frac{\rho}{2}\|Z^{k+1} - Z^k\|_2^2 \leq \mathcal{L}_{\text{real}}(Z^k).$$

The optimality condition is

$$\nabla \mathcal{L}_{\text{real}}(Z^{k+1}) = -\rho(Z^{k+1} - Z^k).$$

(b) Prox-linear update. The quadratic subproblem has the explicit solution

$$\Theta^{k+1} = \frac{\tau\Theta^k + \rho(Z^k - U^k) - \nabla \widetilde{\mathcal{L}}_{\text{real}}(\Theta^k)}{\tau + \rho}.$$

If $\Theta^k = Z^k \in \mathbb{R}^P$ and $U^k = 0$, the zero imaginary gradient makes this update real. Projection leaves it unchanged and the dual remains zero. By induction, all subsequent updates are

$$Z^{k+1} = Z^k - \frac{1}{\tau + \rho}\nabla \mathcal{L}_{\text{real}}(Z^k).$$

The descent inequality for a function with Lipschitz gradient (Bertsekas & Tsitsiklis, 2000, Eq. (2.4)) gives

$$\mathcal{L}_{\text{real}}(Z^{k+1}) \leq \mathcal{L}_{\text{real}}(Z^k) - \left(\tau + \rho - \frac{\widetilde{L}}{2}\right)\|Z^{k+1} - Z^k\|_2^2,$$

whose coefficient is positive. The update also gives

$$\nabla \mathcal{L}_{\text{real}}(Z^k) = -(\tau + \rho)(Z^{k+1} - Z^k).$$

In both cases, the loss is nonincreasing. Coercivity and continuity make its initial sublevel set compact, so the iterates are bounded. Summing the corresponding descent inequality and using the lower bound on the loss yields

$$\sum_{k \geq k_0} \|Z^{k+1} - Z^k\|_2^2 < \infty.$$

Thus $Z^{k+1} - Z^k \to 0$. The gradient identities above, with fixed ρ and τ, imply $\|\nabla \mathcal{L}_{\mathrm{real}}(Z^k)\|_2 \to 0$. Continuity of the gradient then shows that every limit point is stationary for $\mathcal{L}_{\mathrm{real}}$. □

Algorithm 1 CC-ADMM training: active exploration and real tail

Require: Complex initial parameters Θ^0; fixed skew M; schedules $\lambda_k \geq 0$, $\rho_k > 0$; exploration boundary T_{explore}; $K = 5$; task-specific forward model and matched baseline optimizer schedule.

$Z^0 \leftarrow \Re\Theta^0$, $U^0 \leftarrow 0$
for $k = 0, \ldots, T_{\text{explore}} - 1$ **do**
  Hold Z^k, U^k fixed until the next consensus update
  Take K stochastic AdamW steps in Θ on the task loss plus $\lambda(t)\mathcal{L}_M(\Theta) + \frac{\rho(t)}{2}\|\Theta - Z^k + U^k\|_2^2$, using the configured schedules at each inner step
  $Z^{k+1} \leftarrow \Re(\Theta^{k+1} + U^k)$
  $U^{k+1} \leftarrow U^k + \Theta^{k+1} - Z^{k+1}$
end for
Set imaginary parameters to zero and freeze them
Disable auxiliary loss, consensus constraint and dual updates
Continue the real forward model with the baseline optimizer and schedule
return the trained real parameters

D Details on CC-ADMM Method and Implementation

This is an extended version of the CC-ADMM method described in the main body, including the algorithm details. We minimize the training-time objective $\mathcal{L}_{\text{train}}(\Theta)$ under the consensus constraint $\Theta = Z$, $Z \in \mathbb{R}^P$, by ADMM in scaled form. Introducing the scaled dual $U \in \mathbb{C}^P$, the scaled augmented Lagrangian with consensus penalty $\rho_k > 0$ and auxiliary weight $\lambda_k \geq 0$ reads

$$\mathcal{L}_{\rho_k,\lambda_k}(\Theta, Z, U) = \mathcal{L}_{\text{train}}(\Theta) + \tfrac{\rho_k}{2}\|\Theta - Z + U\|_2^2 - \tfrac{\rho_k}{2}\|U\|_2^2, \qquad Z \in \mathbb{R}^P. \tag{23}$$

The last term is constant in (Θ, Z). ADMM primal minimization and the scaled dual update yield

$$\Theta^{k+1} \in \arg\min_{\Theta}\{\mathcal{L}_{\text{train}}(\Theta) + \tfrac{\rho_k}{2}\|\Theta - (Z^k - U^k)\|_2^2\}, \tag{24}$$

$$Z^{k+1} = \Re(\Theta^{k+1} + U^k), \tag{25}$$

$$U^{k+1} = U^k + \Theta^{k+1} - Z^{k+1}. \tag{26}$$

The practical primal solve uses $K = 5$ stochastic AdamW steps, with an empirically chosen schedule and a finite penalty cap. Auxiliary shutoff and the later return to real parameters is a separate phase of the method as described in the Algorithm 1.

Compared to the experiments the Algorithm 1 gives the base projection and dual form. Any possible experiment-specific warmup gates, schedules and dual momentum are reported with the experiments. Here t counts inner steps. We also note, that the schedules need not remain constant across the K inner steps and that the auxiliary schedule can reach zero before T_{explore}. After the real projection, the choice of optimizer determines convergence in the final phase and Theorem 3 establishes the SGD case under its stated noise and learning-rate assumptions.

D.1 Training loss in the implementation

The Θ-subproblem is not solved exactly: the implementation takes $K = 5$ stochastic AdamW steps, while the consensus and dual updates are closed-form. At each inner step during active exploration, backpropagation computes the gradient of

$$\mathcal{L}_{\text{impl}}(\theta; X, U) = \mathcal{L}_{\text{task}}(f_\theta(x), y) + \tfrac{\rho(t)}{2}\mathcal{R}_{\text{ADMM}}(\theta; X, U) + \lambda(t)\mathcal{L}_M(\theta), \tag{27}$$

where $\theta = \bigcup_\ell \{W_{\text{re}}^{(\ell)}, W_{\text{im}}^{(\ell)}, b_{\text{re}}^{(\ell)}, b_{\text{im}}^{(\ell)}\}$ collects the trainable complex weights and biases of the lifted layers. The unlifted parameters remain real. For a literal complex network with a magnitude readout, a representative task term is

$$\mathcal{L}_{\text{task}}(f_\theta(x), y) = \text{CE}(W_{\text{cls}}|z_L| + b_{\text{cls}}, y), \quad |z|_i = \sqrt{z_{\text{re},i}^2 + z_{\text{im},i}^2 + \delta^2}, \tag{28}$$

where CE is the cross entropy, z_L is the last hidden activation, and $W_{\text{cls}}, b_{\text{cls}}$ are the real classifier parameters.

Here (x, y) denotes an input-target pair and $\delta > 0$ is a numerical smoothing constant, distinct from the path residual ε. This formula is representative of the magnitude-readout models, however, in the experiments the architectures are specified with each experiment. With $X^{(\ell)}$ the real consensus block is given by,

$$\mathcal{R}_{\mathrm{ADMM}} = \sum_{\ell} \left[\|W_{\mathrm{re}}^{(\ell)} - X^{(\ell)} + U_{\mathrm{re}}^{(\ell)}\|_F^2 + \|W_{\mathrm{im}}^{(\ell)} + U_{\mathrm{im}}^{(\ell)}\|_F^2 \right] + \text{corresponding bias terms}, \quad (29)$$

$$\mathcal{L}_M(\theta) = \sum_{\ell} a_\ell \left[2\,\mathrm{tr}\Big((W_{\mathrm{re}}^{(\ell)})^\top M^{(\ell)} W_{\mathrm{im}}^{(\ell)} \Big) + 2 (b_{\mathrm{re}}^{(\ell)})^\top M^{(\ell)} b_{\mathrm{im}}^{(\ell)} \right]. \quad (30)$$

Each $M^{(\ell)}$ acts on the lifted layer's output coordinates. The fixed coefficient a_ℓ records the implemented aggregation: the MLP, CNN and literal grokking models sum layer terms ($a_\ell = 1$), whereas ViT, fully lifted ResNet and Llama average over m lifted layers ($a_\ell = 1/m$). These coefficients can be absorbed into the effective block-diagonal skew matrix. Its norm then includes that scaling. After K inner steps, the controller applies

$$X^{(\ell)} \leftarrow W_{\mathrm{re}}^{(\ell)} + U_{\mathrm{re}}^{(\ell)}, \quad (31)$$

$$U_{\mathrm{im}}^{(\ell)} \leftarrow U_{\mathrm{im}}^{(\ell)} + W_{\mathrm{im}}^{(\ell)}, \quad (32)$$

together with analogous bias updates following from (29) and (30). Without additional momentum or other controller modifications, the real dual update resets $U_{\mathrm{re}}^{(\ell)}$ to zero. The next outer iteration's real penalty therefore measures displacement from the preceding projected real weights. The imaginary dual supplies the imaginary consensus coupling. Any controller momentum is an implementation variant outside the exact updates proved above.

Forward model and theoretical extension. The radial formula (10) is an explicitly defined mathematical extension. Complex affine layers, modReLU and magnitude pooling do not in general make an entire network loss radial in all its parameters. At differentiable points, a conjugation-invariant implemented loss satisfies the weaker identity $\mathcal{L}_{\mathrm{impl,task}}(x, y) = \mathcal{L}_{\mathrm{impl,task}}(x, -y)$, which implies a zero imaginary gradient at $y = 0$. Non-smooth activations and stochastic adaptive inner updates require separate analysis.

On real inputs, modReLU with threshold b restricts to $\mathrm{sign}(x)\,\mathrm{ReLU}(|x| - b)$ and magnitude pooling also remains a magnitude operation. Thus, zeroing imaginary weights defines a real restriction but need not reproduce a ReLU baseline's nonlinearity. If required, an implementation can possibly evaluate that restriction using real arithmetic.

E Additional results

This appendix reports additional results and experimental details. The from-scratch tables compare the real-only AdamW Baseline with CC-ADMM. The Llama experiments compare three models: Baseline (LoRA), CC-ADMM LR and CC-ADMM, as in the main body.

We report best-during-training and chronological final-epoch metrics separately. Best values are selected separately for each reported evaluation metric and may come from different epochs, including the complex exploration phase. However, final metrics come from the last epoch evaluation after projection and continued real-valued training. Standard deviations are across seeds and captions specify population or sample standard deviation.

E.1 MNIST and CIFAR-10

Both MNIST models achieve high test accuracy ($> 97\%$). CC-ADMM and the Baseline are close for the CNN, while the MLP Baseline is slightly better. On CIFAR-10, CC-ADMM improves MLP accuracy and slightly trails the CNN Baseline at the reported training budgets.

E.2 Training loss and checkpoint interpretation

Table 8 reports the mean logged minibatch task loss within the final chronological epoch, averaged across seeds. This quantity excludes the auxiliary and consensus penalties. It summarizes the sampled

Table 5: MNIST ($150/60$ epochs for the two cells, three seeds). Best and chronological-final test accuracies, mean $\pm$ population standard deviation.

Model	Method	Best accuracy	Final-epoch accuracy
MLP	Baseline	0.9840 ± 0.0006	0.9839 ± 0.0007
MLP	CC-ADMM	0.9823 ± 0.0000	0.9819 ± 0.0000
CNN	Baseline	0.9965 ± 0.0002	0.9961 ± 0.0005
CNN	CC-ADMM	0.9964 ± 0.0002	0.9960 ± 0.0003

Table 6: CIFAR-10 ($250/100$ epochs for the two cells, three seeds). Best and chronological-final test accuracies, mean $\pm$ population standard deviation

Model	Method	Best accuracy	Final-epoch accuracy
MLP	Baseline	0.5615 ± 0.0034	0.5579 ± 0.0025
MLP	CC-ADMM	0.5885 ± 0.0015	0.5869 ± 0.0020
CNN	Baseline	0.9199 ± 0.0014	0.9189 ± 0.0015
CNN	CC-ADMM	0.9185 ± 0.0006	0.9179 ± 0.0002

training trajectory, including augmentation, rather than a separate full-training-set evaluation. In the MLP, CNN and ViT comparisons, CC-ADMM has higher final logged training loss despite higher best test accuracy, indicating that the test and generalization gains need not accompany lower training task loss and are at least partially products of exploration and the CC-ADMM objective (task + auxiliary loss + consensus).

E.3 Auxiliary-off controls

The additional control sets $\lambda_0 = 0$ while retaining the imaginary parameters, real-consensus constraint projection and dual updates. Comparing auxiliary-on and auxiliary-off models tests the contribution of the skew auxiliary loss within the lifted architecture. Both models were run together in a separate cohort from the primary real-Baseline comparison.

Across the five grokking tasks, the auxiliary-on model reaches the 0.99 accuracy threshold in $24/25$ runs, compared with $4/25$ with the auxiliary-off. Threshold-step means are conditional on success. In CIFAR-100 the MLP benefits from the auxiliary, whereas the CNN's auxiliary-off mean is higher and the seed ranges overlap. The auxiliary loss's benefit therefore depends on the architecture and task.

E.4 Parameter counts and training cost

Let P_{tot} be the total number of trainable real parameters retained at deployment and fP_{tot} the lifted subset. The frozen Llama backbone is excluded. Counting the primal real/imaginary entries, real consensus constraint copies and complex dual entries gives $P_{\mathrm{tot}}(1 + 4f)$ scalar variables. The implementation additionally stores previous-state snapshots, giving another $3fP_{\mathrm{tot}}$ scalars. Including primal gradients and AdamW's two moment arrays gives approximately $P_{\mathrm{tot}}(4 + 10f)$ scalars, compared with $4P_{\mathrm{tot}}$ for a real Baseline with the same parameter count, before activations, allocator overhead, and skew-matrix buffers. Adapter/head-only lifting reduces this storage overhead in proportion to the lifted fraction. These counts describe training state, floating-point operation counts additionally depend on the forward and backward computations.

The memory measurement uses separate processes on one H100 GPU, batch 128, fp32 and AdamW, after 12 optimizer steps so both optimizer and controller state have been allocated. It measures the active fully lifted phase. Peak allocated memory is $1.60\times$ the Baseline, whereas explicitly counted state is $3.54\times$. The former includes activations and other allocator-managed tensors.

Table 14 reports the larger real ResNet-34 control for Results 5.2. Its parameter count is close to the training-time weight count of lifted ResNet-18, and it gives higher best accuracy at all three budgets. These comparisons show the trade-off between training cost, predictive performance and deployed

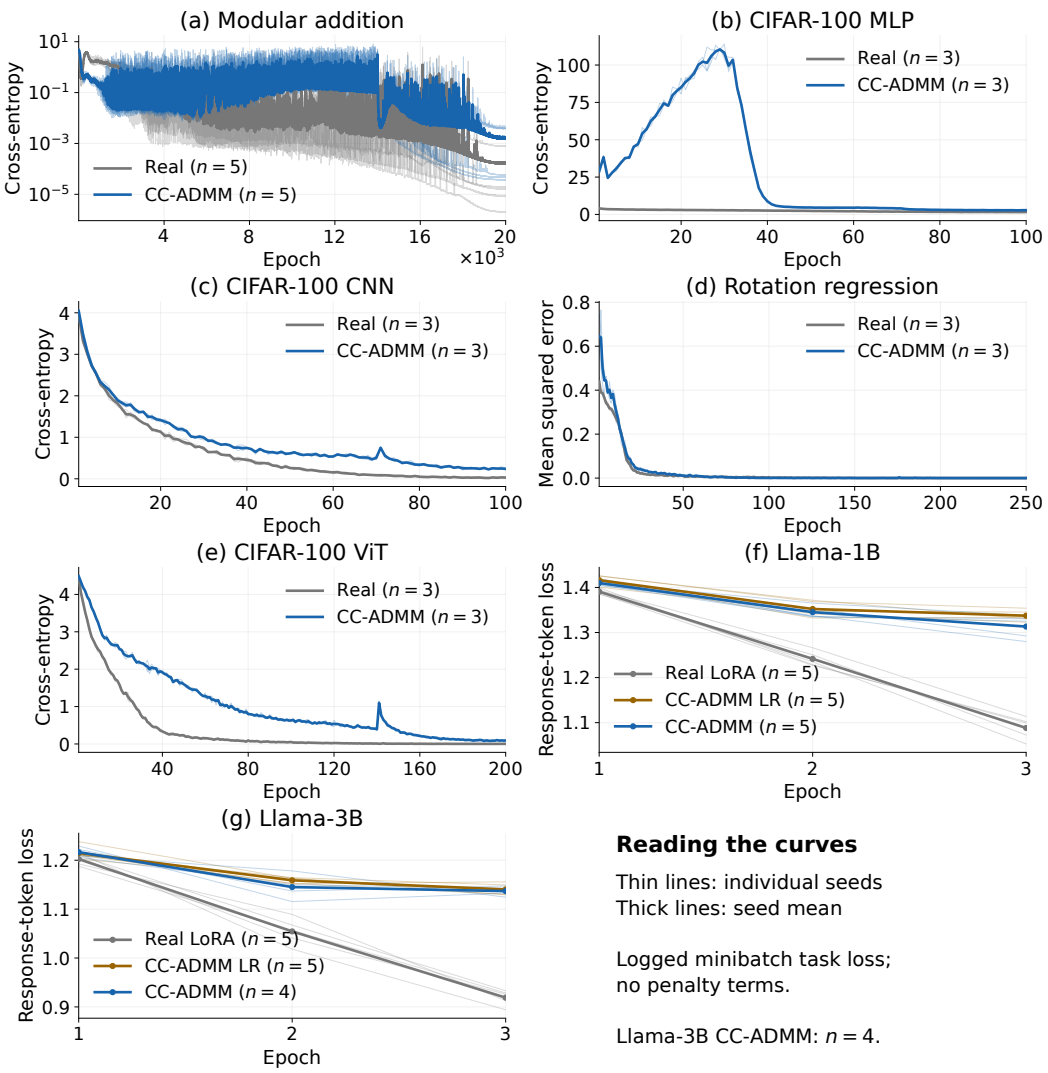

Figure 4: Logged training task-loss trajectories for modular addition, CIFAR-100 MLP/CNN/ViT, rotation regression, and Llama-1B/3B. Each point averages finite logged minibatch task-loss values (i.e., auxiliary loss and consensus constraint excluded) within an epoch and thin lines show individual seeds and thick lines their mean. These are minibatch summaries, not full-training-set evaluations or augmented objectives. Llama-3B CC uses four complete runs. Modular addition represents the grokking family and its loss axis is logarithmic.

Table 7: CIFAR-10 rotation regression with ResNet-34 (three seeds, mean $\pm$ sample standard deviation). Best MSE is the minimum over training epochs. Final MSE and angular error use the chronological final epoch, excluding the best-checkpoint reload row. The angular error is the minimum-arc distance in degrees.

Method	Best MSE	Final MSE	Final angular error
Baseline	0.3112 ± 0.0045	0.3920 ± 0.0046	40.22 ± 0.41
CC-ADMM	0.3041 ± 0.0071	0.3705 ± 0.0146	37.38 ± 1.38

Table 8: Final-epoch mean logged minibatch task loss, excluding auxiliary and consensus penalties (three seeds, mean $\pm$ sample standard deviation). This is not a full-training-set evaluation and is not measured at the best test checkpoint.

Model	Baseline	CC-ADMM
CIFAR-100 MLP	1.625 ± 0.042	2.830 ± 0.061
CIFAR-100 CNN	0.030 ± 0.004	0.239 ± 0.021
ViT-Tiny CIFAR-100	0.004 ± 0.003	0.091 ± 0.014

size: the larger real control has a larger deployed model, while CC-ADMM incurs additional training cost relative to the real ResNet-18.

E.5 Instruction-tuning Llama-3.2

Setup. We instruction-tune Llama-3.2-1B and Llama-3.2-3B on the Stanford Alpaca corpus (Taori et al., 2023), a public collection of approximately 52k `(instruction, input, output)` triples. Following the QLoRA recipe (Dettmers et al., 2023), the base model is loaded in NF4 4-bit quantization and frozen. Trainable parameters are LoRA-style adapters (Hu et al., 2022) of rank $r = 16$, inserted at all seven linear projections of every transformer block: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj` and `down_proj`. We hold out a deterministic 1000-row slice for evaluation and train on the remaining approximately 51k rows for three epochs (AdamW, learning rate 2×10^{-4}, weight decay zero, effective batch size 32, maximum sequence length 512).

Three models share the outer rank $r = 16$. **Baseline (LoRA)** is the standard real adapter with $r(d_{\text{in}} + d_{\text{out}})$ parameters per site and scaling $\alpha/r = 32/16$. **CC-ADMM** is a complex adapter $\text{ComplexLinear}(d_{\text{in}} \to r) \to B_{\text{real}}$, whose deployed real bias adds r parameters per site. **CC-ADMM LR** is the sandwich $A_{\text{real}} \to \text{ComplexLinear}(r \to s) \to B_{\text{real}}$ with inner bottleneck $s = 8 < r$, giving $rd_{\text{in}} + rs + s + sd_{\text{out}}$ deployed parameters, including its real bias. The CC-ADMM adapter deltas are unscaled and use modReLU and magnitude pooling. Summed over all seven projections per block, Baseline/CC-ADMM/CC-ADMM LR adapter counts are $11\,272\,192/11\,273\,984/8\,272\,768$ at 1B and $24\,313\,856/24\,316\,992/18\,147\,360$ at 3B. The LR models therefore use 73.39% and 74.64% of the Baseline adapter count, respectively. The frozen backbone is excluded from these counts.

Evaluation. After every epoch we compute cross-entropy on the 1000-row held-out Alpaca slice with *response-only masking*: instruction-and-template tokens have label -100. We report *Alpaca perplexity* $\text{ppl}_{\text{ID}} = \exp(\bar{\mathcal{L}}_{\text{Alpaca}})$, where $\bar{\mathcal{L}}_{\text{Alpaca}}$ is the mean response-token cross-entropy in nats. As a separate *out-of-distribution* probe of language-modeling retention, we compute next-token perplexity ppl_{OOD} on the WikiText-103 raw test split (Merity et al., 2016) (1024-token windows, 200k-token budget). The model never sees WikiText during fine-tuning.

For both CC-ADMM variants, the auxiliary coefficient decreases linearly from 0.5 to zero during approximately the first 8.75% of training at 1B and 4.375% at 3B. The auxiliary schedule uses the minibatch counter, whereas the projection boundary uses the optimizer-update counter. The gradient accumulation therefore gives different annealing durations at the two scales. The imaginary adapter weights and biases are zeroed and frozen after approximately 70% of the optimizer updates (2.1 epochs).

Table 9: Additional auxiliary ablation: first optimizer step reaching 0.99 test accuracy, with successful runs out of five in parentheses. Means and population standard deviations concern successful runs; a single success has no reported standard deviation. "Not reached" denotes censoring at the run budget, set to $20\,000$ epochs. Both models retain imaginary parameters, consensus and dual updates. These additional runs are reported separately from Table 1, whose addition and division summaries differ.

Operation	Auxiliary on	Auxiliary off ($\lambda_0 = 0$)
$a + b$	$9\,154 \pm 1\,231$ $(5/5)$	$41\,856$ $(1/5)$
$a - b$	$16\,133 \pm 5\,750$ $(5/5)$	Not reached $(0/5)$
$a \times b$	$9\,548 \pm 1\,677$ $(5/5)$	$36\,510$ $(1/5)$
a/b	$22\,598 \pm 5\,931$ $(5/5)$	Not reached $(0/5)$
S_5	$20\,700 \pm 22\,554$ $(4/5)$	$229\,988 \pm 44\,872$ $(2/5)$

Table 10: Additional CIFAR-100 auxiliary ablation: best-during-training test accuracy (three seeds, mean $\pm$ population standard deviation). The lifted architecture and consensus remain active in both models.

Model	Auxiliary on	Auxiliary off ($\lambda_0 = 0$)
CIFAR-100 MLP	0.2623 ± 0.0019	0.2379 ± 0.0021
CIFAR-100 CNN	0.6590 ± 0.0042	0.6664 ± 0.0052

Training then continues on the real task loss with AdamW, with the auxiliary and consensus terms and dual updates disabled.

Final metrics are taken from the third-epoch evaluation. Best metrics are the minima over the three epoch-end evaluations, computed separately for each metric. For every completed run of both CC-ADMM variants, the best Alpaca loss occurs at the final real-model evaluation, whereas the best WikiText-103 perplexity occurs before projection. Both are reported.

Findings at 1B. With the full seven-projection target set, both CC-ADMM models outperform the LoRA Baseline on final in-domain Alpaca and OOD WikiText-103 perplexity. CC-ADMM is strongest in-domain, with Alpaca perplexity 4.035 vs. 4.197 for the Baseline; CC-ADMM LR sits between them at 4.052. The OOD results show a similar pattern: the Baseline's WikiText-103 perplexity climbs from a best-during-training value of approximately 12.90 to a final 14.085, while both CC-ADMM variants remain close to their best values, finishing at 12.299 for CC-ADMM LR and 12.360 for CC-ADMM. This is consistent with a regularization benefit during instruction-tuning.

Findings at 3B. At 3B, CC-ADMM again has the lowest mean final Alpaca loss among the three methods. Both CC-ADMM variants outperform the LoRA Baseline on the final in-domain loss and OOD perplexity: final loss is 1.2974 for the Baseline, 1.2646 for CC-ADMM LR and 1.2633 for CC-ADMM. Final OOD perplexity is 10.317, 9.394 and 9.388, respectively. As at 1B, the Baseline's OOD perplexity drifts upward from its best-during-training value ($9.611 \rightarrow 10.317$), while both CC-ADMM variants remain near their best values. Figure 2 reports the best-so-far Alpaca loss (corresponding to oracle. i.e., best test loss selected "early stopping" for Baseline models) and observed WikiText-103 perplexity. Figures 5 and 6 show the measurements at each epoch end and show that 3 Baseline runs start overfitting already at epoch 2, while CC-ADMM keeps exploring and improving.

Epoch-end trajectories and checkpoint selection. Figures 5 and 6 show the values measured at each epoch end, complementing the running-best Alpaca curve in Figure 2 in the main body. Across all completed runs, the Baseline's mean per-run best and final Alpaca losses are 1.4003 and 1.4343 at 1B, and 1.2675 and 1.2974 at 3B. Alpaca loss decreases at every recorded epoch in all completed CC-ADMM and CC-ADMM LR runs, so their best observed Alpaca losses occur at the final epoch, conveying the message that longer exploration is beneficial for CC-ADMM, while "standard finetuning" starts to overfit in longer runs. The best values in the tables are selected separately for each metric within each run and then aggregated across runs. Note that they need not

Table 11: Reported deployed parameter counts and approximate lifted fractions. f is the fraction of the CC-ADMM deployed parameters with an imaginary twin. The MLP/CNN comparisons differ in activation and readout as well as optimization; their Baselines use ReLU, whereas the lifted models use modReLU with threshold 0.5 and a magnitude readout. ViT uses the same real linear forward map in both models. Counts are not optimizer/controller storage.

Family	Baseline deployed	CC-ADMM deployed	Reported f
CIFAR-100 MLP	2 412 132	2 149 988	0.98
CIFAR-100 CNN	1 853 988	1 591 844	0.25
Grokking ($\mathbb{Z}_{97}$)	423 011	373 411	0.0221
Grokking (S_5)	428 922	379 322	0.0218
ViT-Tiny	5 380 132	5 380 132	0.9905
ResNet-18 strong recipe	11.22M	11.22M	$\simeq 1$

Table 12: Memory for the fully lifted CIFAR-100 ResNet-18: batch 128, fp32, AdamW, one H100 GPU, separate processes, measured after 12 training steps so optimizer and controller state are allocated. Units are MB as reported. State subtotals are tensor counts; peaks are allocator measurements, not sums of per-component peaks.

Component	Baseline	CC-ADMM	Ratio
Weights	44.88	89.72	2.00×
Gradients	44.88	89.72	2.00×
AdamW moments	89.76	179.45	2.00×
Model buffers (including skew matrices)	0.04	7.04	—
ADMM controller state	0	269.05	—
State subtotal	179.56	634.98	3.54×
Peak allocated	849.29	1358.40	1.60×
Peak reserved	947.91	1484.78	1.57×

describe the same checkpoint. For both CC-ADMM variants, the first two evaluations use complex parameters, while the third follows projection and continued real-valued training.

E.6 Vision Transformer from scratch (ViT-Tiny on CIFAR-100)

Setup. A ViT-Tiny (image size 32, patch size 4, embedding dimension 192, depth 12, three heads) is trained from scratch on CIFAR-100 with cross-entropy and AdamW using the recipe in Section 5.5. The Baseline is a plain real ViT. CC-ADMM replaces every transformer-block linear layer (qkv, projection and two MLP layers, 48 sites in total) with a complex-shadow linear layer. It uses W_{re} in the forward pass, while W_{im} is trained through the auxiliary loss and consensus constraint and zeroed at the real-phase transition. After this transition, both models have 5.38M real parameters and the same forward architecture.

The augmentation pipeline is the standard CIFAR pair (`RandomCrop(32, padding=4)` and horizontal flip). We train for 200 epochs without MixUp, CutMix, RandAugment or label smoothing. This within-recipe comparison holds the deployed architecture and augmentation fixed while changing the training method. Published results using different ViT/CCT architectures and augmentation recipes are not directly comparable.

Findings. CC-ADMM improves the best test accuracy by 1.24 percentage points ($0.5977 \rightarrow 0.6101$) and reduces the best test loss by approximately 0.22 (Table 3). The final test loss is approximately 24% lower. The training-loss trajectories are shown in Figure 4; the higher final logged task loss alongside the improved test performance is consistent with reduced overfitting.

F Existing assets and computational resources

This section lists the external datasets, models and software used in the experiments, with their source links.

Table 13: Total logged wall-clock time in minutes (three seeds, mean $\pm$ sample standard deviation), including logged evaluations. The autoencoder and plain-56 are timing-only benchmarks here. Each ratio divides the means; hardware and pipeline are shared within a comparison.

Model	Epochs	Baseline (min)	CC-ADMM (min)	Ratio
MNIST autoencoder	200	7.17 ± 0.17	12.74 ± 0.30	$1.78\times$
CIFAR-10 plain-56 (SGD)	200	28.53 ± 1.48	52.39 ± 2.84	$1.84\times$
CIFAR-100 ResNet-18	300	32.34 ± 5.31	48.32 ± 3.72	$1.49\times$

Table 14: Larger real ResNet-34 control for the CIFAR-100 experiment in Section 5.2, using the same strong recipe and three seeds. Values are best test accuracy, mean $\pm$ population std. This model deploys 21.33M parameters, close to the lifted ResNet-18's 22.43M training-time weights but larger than its 11.22M real deployment model.

Epochs	ResNet-34 real control
300	0.8095 ± 0.0022
900	0.8213 ± 0.0029
2700	0.8141 ± 0.0004

PRETRAINED MODELS

Llama-3.2-1B and Llama-3.2-3B. Meta's base models are used as frozen NF4-quantized backbones for the Alpaca instruction-tuning experiments. The model identifiers are `meta-llama/Llama-3.2-1B` and `meta-llama/Llama-3.2-3B`; model cards and terms are available at `https://huggingface.co/meta-llama/Llama-3.2-1B` and `https://huggingface.co/meta-llama/Llama-3.2-3B`. Use is governed by the Llama 3.2 Community License and Acceptable Use Policy.

ViT-Tiny from scratch. The model is built with `timm.models.vision_transformer.VisionTransformer`, using `pretrained=False`; no external pretrained weights are used.

DATASETS

Stanford Alpaca. The Llama instruction-tuning dataset (Taori et al., 2023) contains 52 002 `(instruction, input, output)` triples generated with `text-davinci-003`. Its identifier is `tatsu-lab/alpaca` (`https://huggingface.co/datasets/tatsu-lab/alpaca`); the provider lists a CC BY-NC 4.0 license.

WikiText-103 (raw). We use the held-out test split (Merity et al., 2016), configuration `wikitext-103-raw-v1`, with 1024-token windows and a 200k-token evaluation budget (`https://huggingface.co/datasets/Salesforce/wikitext`). The model never sees this corpus during fine-tuning; the dataset card supplies its attribution and licensing terms.

CIFAR-10 / CIFAR-100. The image-classification and rotation experiments use these datasets (Krizhevsky, 2009), distributed at `https://www.cs.toronto.edu/~kriz/cifar.html`.

MNIST. The handwritten-digit dataset (`https://yann.lecun.com/exdb/mnist/`) is used in the additional MLP/CNN comparisons.

CODE PACKAGES AND COMPUTE

The implementation is written in Python with PyTorch (Paszke et al., 2019). Hugging Face `transformers`, `peft` and `datasets` support the Llama instruction-tuning pipeline; `bitsandbytes` provides NF4 quantization, and `timm` (Wightman, 2019) provides ViT model components.

Table 15: Llama-3.2-1B: best/final-epoch metrics, five seeds, mean ± sample standard deviation. CC-ADMM adds a small real bias to the Baseline adapter weight count; CC-ADMM LR has 73.39% of the Baseline adapter count. The quantized backbone is frozen in every model.

Metric	Baseline (LoRA) (n=5)	CC-ADMM LR (n=5)	CC-ADMM (n=5)
best test_loss	1.4003 ± 0.0019	1.3991 ± 0.0007	$\mathbf{1.3949 \pm 0.0006}$
final test_loss	1.4343 ± 0.0024	1.3991 ± 0.0007	$\mathbf{1.3949 \pm 0.0006}$
best Alpaca PPL	4.056 ± 0.008	4.052 ± 0.003	$\mathbf{4.035 \pm 0.002}$
final Alpaca PPL	4.197 ± 0.010	4.052 ± 0.003	$\mathbf{4.035 \pm 0.002}$
best OOD PPL	12.895 ± 0.309	$\mathbf{12.142 \pm 0.021}$	12.287 ± 0.014
final OOD PPL	14.085 ± 0.046	$\mathbf{12.299 \pm 0.031}$	12.360 ± 0.016

Table 16: Llama-3.2-3B: best/final-epoch metrics, mean ± population standard deviation. Baseline and CC-ADMM LR have five evaluated runs, while CC-ADMM has four.

Metric	Baseline (LoRA) (n=5)	CC-ADMM LR (n=5)	CC-ADMM (n=4/5)
best test loss	1.2675 ± 0.0007	1.2646 ± 0.0007	$\mathbf{1.2633 \pm 0.0003}$
final test loss	1.2974 ± 0.0005	1.2646 ± 0.0007	$\mathbf{1.2633 \pm 0.0003}$
best OOD PPL	9.611 ± 0.032	$\mathbf{9.289 \pm 0.012}$	9.358 ± 0.015
final OOD PPL	10.317 ± 0.031	9.394 ± 0.011	$\mathbf{9.388 \pm 0.018}$

Experiments use university HPC clusters with GPU-accelerated nodes and batch scheduling. The archived jobs use one NVIDIA H100 GPU per job. Appendix E.4 gives measured wall-clock time and GPU memory for the specified workloads and protocols.

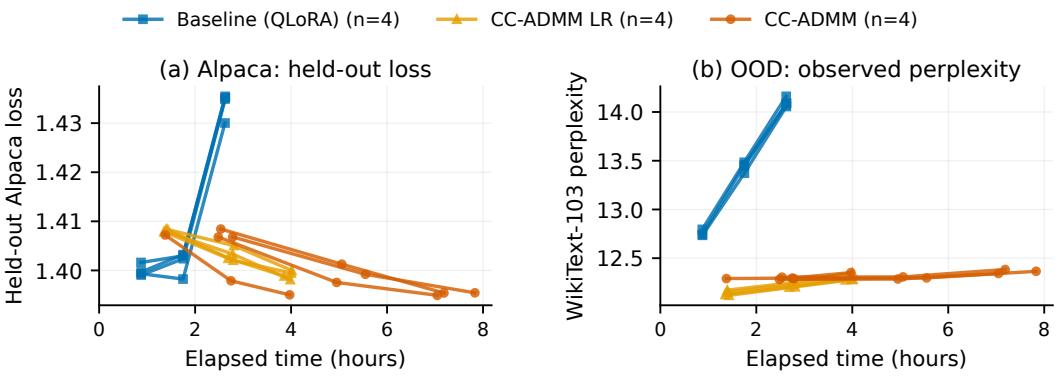

Figure 5: Llama-3.2-1B: held-out Alpaca cross-entropy (left) and WikiText-103 perplexity (right) versus logged training and evaluation time, with four runs per method. Markers show epoch-end measurements and straight segments connect evaluations from the same run.

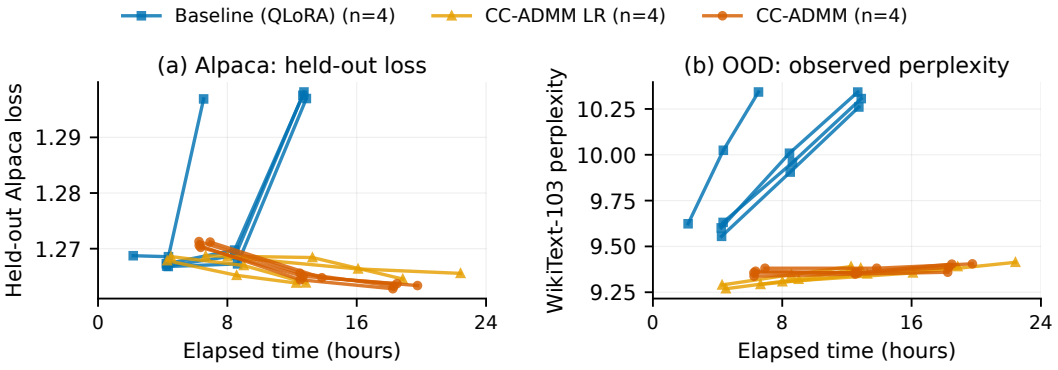

Figure 6: Llama-3.2-3B: held-out Alpaca cross-entropy (left) and WikiText-103 perplexity (right) versus logged training and evaluation time, with four runs per method. Markers show epoch-end measurements and straight segments connect evaluations from the same run.