# EXPECTATION CURVATURE: BEYOND THE HESSIAN IN NON-SMOOTH LOSS LANDSCAPES

## **Anonymous authors**

Paper under double-blind review

#### **ABSTRACT**

Second-order methods seek to exploit loss curvature, but in deep networks the Hessian often fails to approximate it well, especially near sharp gradient transitions induced by common activation functions. We introduce an analytic framework that characterizes *curvature of expectation*, showing how such transitions generate pseudorandom gradient perturbations that combine into a glass-like structure, analogous to amorphous solids. From this perspective we derive: (i) the density of gradient variations and bounds on expected loss changes, (ii) optimal kernels and sampling schemes to estimate both Hessian and glass curvature from ordinary gradients, and (iii) quasi-Newton updates that unify these curvature terms with exactness conditions under Nesterov acceleration. To probe their empirical role, we implement ALICE, a lightweight diagnostic that inserts curvature estimates into controlled updates, revealing which terms genuinely influence optimization. In this way, our results support further optimization research: they introduce a new theoretical picture of nonsmooth loss landscapes that can catalyze future advances in pruning, quantization, and curvature-aware training.

#### 1 Introduction

First-order optimizers such as SGD and Adam remain standard practice in deep learning, yet their efficiency is limited by the lack of curvature information. Second-order methods promise faster convergence by incorporating local curvature through the Hessian (Becker et al., 1988; LeCun et al., 1989; Bottou et al., 2018). However, deep networks with rectified linear units (ReLUs) and related activations introduce steep gradient transitions that break the smoothness assumptions underpinning Hessian-based methods. In these regimes, instantaneous second derivatives are poorly suited for extrapolation, leading to unreliable curvature estimates.

Across many parameters, each activation defines a boundary in parameter space. Crossing such a boundary changes gradient flow abruptly, producing pseudorandom perturbations that accumulate throughout the network. Collectively, these boundaries form a *gradient glass*: a structure analogous to amorphous solids, comprising small domains of local alignment despite global disorder at larger scales. Intuitively, parameters that influence early computations encounter many downstream boundaries and thus exhibit stronger glass-like gradient behavior, while parameters closer to the output layer admit smoother, Hessian-dominated loss analysis. Figure 1 illustrates this idea in two dimensions: gray lines mark activation boundaries, and blue arrows indicate pseudorandom gradient shifts encountered upon crossing them. A hypothetical training trajectory (violet) passes through successive domains. The right panel shows how gradient and loss vary along the trajectory, with expectation curves providing consistent bounds on the range of fluctuations.

An Empirical Motivation To test whether Hessian curvature explains these effects, we measured how gradient variations scale with perturbation distance. Let  $\mathcal{L}(\theta)$  denote average training loss with gradient  $g(\theta)$  for parameters  $\theta \in \mathbb{R}^d$ . Perturbing parameters by a random Rademacher vector  $\boldsymbol{\delta}$  of fixed distance  $\lambda$ , we measure gradient variations

$$v(\lambda) = \mathbb{E}\left[\gamma^2\right] \propto \lambda^p \quad \text{where} \quad \gamma = g(\mu + \lambda \delta) - g(\mu).$$
 (1)

Here and going forward, all vector powers are taken elementwise. If variations were dominated by the Hessian, we would expect quadratic scaling (p = 2). Yet experiments with ResNet18 on

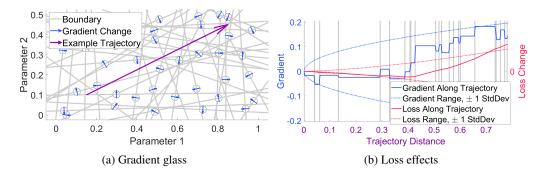


Figure 1: Illustration of gradient glass in 2D (left). Gray lines are domain boundaries due to changing a ReLU state. Blue arrows show pseudorandom gradient perturbations. The resulting expectation curvature (right) bounds how the loss evolves under parameter displacements.

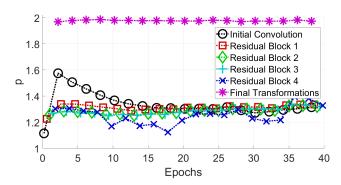


Figure 2: Scaling of gradient variations with perturbation distance. A pure Hessian contribution gives p=2. Observed exponents p<2 indicate additional curvature effects beyond the Hessian.

CIFAR-10 reveal consistently smaller exponents (Figure 2), indicating subquadratic growth. No low-order Taylor expansion can yield p < 2, suggesting that gradient discontinuities contribute an additional source of curvature beyond the Hessian. We further observe that p=2 dependence is recovered for parameters that follow the final activation layers, underscoring the connection between curvature behavior and activation-induced gradient discontinuities.

**Contributions** We present an analytic framework for *curvature of expectation*, capturing how loss evolves under parameter perturbations in the presence of activation-induced gradient discontinuities. Our main contributions are:

- 1. a theoretical model of *gradient glass*, quantifying the density of gradient variations induced by ReLU-like activation boundaries,
- 2. derivation of an optimal kernel for unbiased, minimum-variance diagonal estimation from randomized matrix–vector products,
- 3. identification of the optimal perturbation distribution, showing that Rademacher samples minimize estimator variance and recover Hutchinson trace estimation as a special case,
- 4. analytic bounds on expected loss changes, yielding a 3/2 power-law dependence that complements Hessian-based quadratic growth,
- 5. a modified quasi-Newton update that unifies Hessian and glass curvature terms, producing per-coordinate effective curvature with provable stability properties, and
- 6. exactness conditions under Nesterov acceleration, linking momentum and damping parameters to guarantee correction of hidden linear gradient components.

Together, these results establish a principled theory of curvature in nonsmooth networks, explain long-standing discrepancies in Hessian-based methods, and provide concrete algorithms for making this hidden structure measurable and actionable.

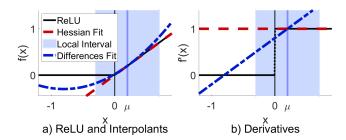


Figure 3: ReLU extrapolation. Hessian fitting cannot account for gradient discontinuities, whereas curvature matching gradient changes provides a better fit on a local interval.

**Alice** To ground the theory in experiment, we introduce ALICE, a streamlined probe designed to expose the role of curvature extraction. Alice functions as an investigative instrument: it isolates Hessian and glass contributions through controlled updates, providing clear evidence of when curvature-of-expectation matters. Our experimental results with Alice show that curvature terms are not only theoretically coherent, but empirically accessible, opening paths for future work in pruning, quantization, regularization, and the design of curvature-aware training methods.

# 2 RELATED WORK

**Second-Order Optimization** Second-order methods approximate curvature to accelerate training beyond first-order updates. Early work incorporated Hessian information into multilayer perceptrons, computing diagonal terms for pruning (LeCun et al., 1989) and improving convergence through quasi-Newton (QN) updates (Becker et al., 1988). A comprehensive survey is provided by Bottou et al. (2018). Recent techniques avoid explicit Hessian construction by using matrix-vector products, enabling scalable spectral analysis and visualization of loss landscapes (Yao et al., 2020).

Active Developments Curvature remains a focus of active research due to its link with generalization. Plasticity and catastrophic forgetting have been tied to Hessian eigenvalues (Lyle et al., 2023; Kong et al., 2023). Other studies connect curvature to transfer performance (Hemati et al., 2023), adversarial robustness (Li and Spratling, 2023), and federated optimization (Sen et al., 2023). Sharpness-Aware Minimization (SAM) leverages Hessian eigenvalue control to improve flatness (Kaur et al., 2023), while suppression of curvature along training trajectories can guide convergence to flat minima (Lee et al., 2023a;b). Extensions combining QN methods with Nesterov acceleration (Ninomiya, 2017; Indrapriyadarsini et al., 2020) and variational inference (Duersch, 2024) further illustrate the breadth of ongoing developments.

Hessian Shortcomings and Glass Analogies Despite these advances, Hessian-based curvature is not reliable near sharp activation transitions. For ReLU-like units, extrapolation based on instantaneous second derivatives clearly cannot anticipate the gradient discontinuities (Figure 3). Empirical studies confirm the jagged loss topography that emerges in quantization (Frumkin et al., 2023) and the *v*-shaped loss structures observed along random directions (Li et al., 2024). These observations align with results from statistical physics, where spin-glass analogies explain the proliferation of saddle points and the role of curvature rectification (Dauphin et al., 2014; Parisi, 2007). Later work connects glass-like behavior to generalization phase transitions (Choromanska et al., 2015; Spigler et al., 2019). Our approach builds on this perspective by modeling gradient discontinuities as a *gradient glass*, and by deriving analytic tools to exploit the resulting curvature of expectation for optimization.

# 3 EXTRACTING AND EXPLOITING GLASS-LIKE CURVATURE OF EXPECTATION

We now present our framework to model curvature in networks with gradient discontinuities induced by ReLUs. Each ReLU introduces a parameter boundary where the gradient shifts abruptly, and collectively these create a glass-like structure in the loss landscape. We quantify this effect by deriving the density of gradient variations and show how it complements Hessian curvature. Using randomized gradient evaluations, we obtain optimal estimators for both glass and Hessian diagonals, derive bounds on expected loss changes, and construct a modified quasi-Newton step. We also establish exactness conditions for Nesterov acceleration under this structure. These results provide the foundation for curvature-aware optimization in nonsmooth settings. All proofs appear in Appendix A.

**Theorem 1** (Glass from ReLUs). Consider a network with many ReLUs. Hold network inputs fixed and assume each pre-activation is locally linear in small parameter perturbations  $\theta = \mu + \delta$ :

$$y(\boldsymbol{\mu} + \boldsymbol{\delta}) = y(\boldsymbol{\mu}) + \boldsymbol{\delta}^{\top} \hat{\boldsymbol{\gamma}} \quad \text{with} \quad \hat{\boldsymbol{\gamma}} = \nabla_{\boldsymbol{\mu}} y(\boldsymbol{\mu}).$$
 (2)

Let the upper bound on pre-activation shifts be  $|\delta^{\top}\hat{\gamma}| \leq \psi$ , and define the set of near-threshold units  $S_{\psi} = \{k : |y^{(k)}(\mu)| < \psi\}$ . Within this set, suppose pre-activations are uniformly distributed in  $[-\psi, \psi]$ .

If gradient jumps from individual activation flips are independent, zero-mean pseudorandom variables, then the vector of gradient-variations  $v(\delta)$  given by Equation (1) satisfies

$$\mathbf{v}(\boldsymbol{\delta}) \leq \mathbf{R}|\boldsymbol{\delta}| \quad \text{where} \quad \mathbf{R}_{ij} = \frac{1}{2\psi} \sum_{k \in \mathcal{S}_{\psi}} \left( \frac{d\mathcal{L}(\boldsymbol{\mu})}{dz^{(k)}} \right)^2 \hat{\boldsymbol{\gamma}}_i^{(k)2} |\hat{\boldsymbol{\gamma}}_j^{(k)}|.$$
 (3)

The density matrix R quantifies additional gradient variation per unit length, beyond the Hessian. Any activation with ReLU-like derivative discontinuities will induce such terms, and averaging over inputs diminishes but does not eliminate their effects. Pointwise Hessians cannot detect them.

Because diagonal entries scale cubically with the same coordinate while off-diagonals are only quadratic-linear products, the largest coordinates of  $\hat{\gamma}^{(k)}$  push their mass disproportionately onto the diagonal. Summing over many such contributions amplifies this diagonal dominance, making a diagonal approximation of R a reasonable simplification in practice.

**Theorem 2** (Optimal Kernel for Diagonal Estimation). Let M be a linear operator accessible through matrix–vector products  $\mathbf{y} = M\delta$ . Let  $p(\delta) = \prod_{i=1}^d p(\delta_i)$  be a product distribution with i.i.d. zero-mean, unit-variance coordinates. For each  $i \in [d]$ , define the relative off-diagonal mass  $\omega_i^2 = \sum_{j \neq i} M_{ij}^2/m_i^2$  where  $m = \operatorname{diag}(M)$ . Then any kernel  $\kappa_i$  that yields an unbiased estimator of  $m_i$  from a single product satisfies

$$\mathbb{E}[\kappa_i(\boldsymbol{\delta}_i)\,\boldsymbol{y}_i] = \boldsymbol{m}_i. \tag{4}$$

Among all such kernels, the unique minimum-variance choice is

$$\kappa_i^*(\boldsymbol{\delta}_i) = \frac{c^{-1}\,\boldsymbol{\delta}_i}{\boldsymbol{\delta}_i^2 + \boldsymbol{\omega}_i^2}, \qquad c = \int \frac{\boldsymbol{\delta}_i^2}{\boldsymbol{\delta}_i^2 + \boldsymbol{\omega}_i^2} \, dp(\boldsymbol{\delta}_i). \tag{5}$$

This applies to any i.i.d. sample density. When the perturbation distribution is also optimized, the kernel simplifies further.

**Theorem 3** (Optimal Perturbation Density). Among product perturbation densities with i.i.d. factors  $p(\delta) = \prod_{i=1}^d p(\delta_i)$ , each with zero mean and unit variance, the variance-minimizing choice in Theorem 2 is the Rademacher distribution on each factor, i.e. probability 1/2 of  $\pm 1$ . Then Equation (5) simplifies to  $\kappa^*(\delta_i) = \delta_i$ .

This result recovers Hutchinson-style trace estimation as a special case, but here derived formally to permit estimation of diagonals for both R as well as the Hessian.

**Theorem 4** (Curvature of Expectation in Glass Loss). As we displace parameters by  $\delta$ , the increase in loss  $\Delta \mathcal{L}(\delta) = \mathcal{L}(\mu + \delta) - \mathcal{L}(\mu)$  is maximized if a local floor enforces  $\Delta \mathcal{L}(\delta_k) \geq 0$ . Using the diagonal approximation  $\mathbf{v} \approx \boldsymbol{\rho} * |\delta|$  with  $\boldsymbol{\rho} \succeq 0$ , and treating coordinates independently, the expected loss increase is bounded by

$$\mathbb{E}[\Delta \mathcal{L}(\boldsymbol{\delta})] \le \sqrt{\frac{2}{3\pi}} \, \boldsymbol{\rho}^{\frac{1}{2}T} |\boldsymbol{\delta}|^{\frac{3}{2}}. \tag{6}$$

Without a local floor, the symmetry of gradient perturbations would lead to flat expectation. The floor ensures perturbations only increase expected loss, consistent with "reflection" phenomena observed empirically (Li et al., 2024).

**Theorem 5** (Modified Quasi-Newton Step). Combining the bound of Equation (6) with the gradient g and nonnegative Hessian diagonal  $h \succeq 0$  gives

 $\mathcal{L}(oldsymbol{\mu} + oldsymbol{\delta}) \leq \mathcal{L}(oldsymbol{\mu}) + oldsymbol{\delta}^{ op} \Big( oldsymbol{g} + rac{1}{2} oldsymbol{h} st oldsymbol{\delta} \Big) + \sqrt{rac{2}{3\pi} \, oldsymbol{
ho}^{ op} |oldsymbol{\delta}|^3}.$ 

The minimizer is  $\boldsymbol{\delta} = -\boldsymbol{q} * \boldsymbol{\bar{h}}^{-1}$  with modified Hessian

$$\bar{\mathbf{h}} = \hat{\mathbf{h}} + \mathbf{h} + \sqrt{\hat{\mathbf{h}} * (\hat{\mathbf{h}} + 2\mathbf{h})} + \varepsilon, \quad \hat{\mathbf{h}} = 3 \, \boldsymbol{\rho} * (4\pi |\mathbf{g}| + \varepsilon)^{-1},$$
 (7)

where  $\varepsilon > 0$  ensures numerical stability.

Here  $\varepsilon > 0$  ensures numerical stability. The effective curvature interpolates between Hessian and glass terms, shortening optimal steps in both regimes.

Having established how glassy gradient fluctuations lead to a modified quasi-Newton step with diagonal correction (Theorem 5), the natural question is how such steps interact with momentum and acceleration schemes. In particular, Nesteroy's method separates the parameter update from the evaluation point, and so provides an opportunity to test whether the quadratic loss model with  $\bar{h}$  can be made exactly consistent with the accelerated gradient dynamics. The following theorem shows that a specific choice of update and evaluation fractions aligns the quasi-Newton model with Nesterov acceleration, while ensuring contraction of residual gradient errors.

Theorem 6 (Exact Nesterov Accelerated Quasi-Newton). In Nesterov acceleration, parameters and gradient evaluations evolve by

$$\mu^{(s+1)} = \mu^{(s)} + \varphi \, \delta^{(s)}, \qquad \qquad \nu^{(s+1)} = \mu^{(s)} + \omega \, \delta^{(s)}, \qquad (8)$$

$$\mathbf{g}^{(s+1)} = \beta_1 \mathbf{g}^{(s)} + (1 - \beta_1) g(\mathbf{\nu}^{(s)}), \tag{9}$$

where  $\boldsymbol{\delta}^{(s)}$  is the tentative update,  $\varphi \leq 1$  is the update fraction, and  $\omega \geq \varphi$  is the evaluation fraction.

Suppose we approximate the loss quadratically with the modified Hessian  $\bar{h}$  from Equation (7), so that the local gradient is

$$g(\boldsymbol{\mu}^{(s)} + \boldsymbol{\delta}) = \boldsymbol{g}^{(s)} + \bar{\boldsymbol{h}} * \boldsymbol{\delta}, \tag{10}$$

with optimal step  $\delta^{(s)} = -q^{(s)} * \bar{h}^{-1}$ . Yet assume the true dependence is linear but unknown

$$g^*(\boldsymbol{\mu}^{(s)} + \boldsymbol{\delta}) = g^*(\boldsymbol{\mu}^{(s)}) + \boldsymbol{H}\boldsymbol{\delta}$$
(11)

that remains valid through  $\delta^{(s)}$ . Then choosing

$$\varphi = 1 - \beta_1, \quad \omega = 1, \tag{12}$$

ensures that the momentum update Equation (9) matches the quadratic model Equation (10), while capturing the unknown linear dependence in Equation (11). Moreover, if

$$g^{(s)} = g^*(\mu^{(s)}) + \gamma^{(s)}$$
 then  $g^{(s+1)} = g^*(\mu^{(s+1)}) + \beta_1 \gamma^{(s)}$ , (13)

so errors are reduced by factor  $\beta_1$  each step.

This links the damping factor and momentum parameter, showing that  $\beta_1$  governs memory, damping, and correction of off-diagonal effects simultaneously.

Summary Theorems 1–6 provide a compact pipeline: ReLU-induced gradient jumps produce a measurable glass density (Theorem 1); diagonal elements of both glass and Hessian operators can be estimated optimally from randomized matrix-vector products when using Rademacher samples (Theorem 2, 3); the glass term tightens expected loss growth to a 3/2 power law (Theorem 4), which directly yields a modified per-coordinate curvature used in quasi-Newton updates (Theorem 5); finally, damping and Nesterov-style evaluation choices make these updates stable and exact for hidden linear corrections (Theorem 6). Full proofs appear in Appendix A.

# Algorithm 1 Alice Topography Update

Input: evaluation center:  $\nu$ ; gradient function:  $g(\theta)$ . Input and Output: running averages: g,  $\rho$ , and h. Hyperparameters:  $\lambda$ ,  $\beta_1$ ,  $\beta_2$ .

1: Draw Rademacher t.

- 2: Evaluate  $g^{(\pm)} = g(\boldsymbol{\nu} \pm \lambda \boldsymbol{t})$  and  $g^{(0)} = g(\boldsymbol{\theta})$ .
- 3:  $\mathbf{g} \leftarrow \beta_1 \mathbf{g} + (1 \beta_1) \mathbf{g}^{(0)}$ .
- 4:  $\mathbf{h} \leftarrow \beta_2 \mathbf{h} + (1 \beta_2) \frac{1}{2\lambda} |\mathbf{g}^{(+)} \mathbf{g}^{(-)}|$ .
- 5:  $\rho \leftarrow \beta_2 \rho + (1 \beta_2) \frac{2}{\lambda} \left( \frac{1}{2} (\boldsymbol{g}^{(+)} + \boldsymbol{g}^{(-)}) \boldsymbol{g}^{(0)} \right)^2$ .

### 4 EXPERIMENTS AND RESULTS

To assess our theoretical framework empirically, we implement *Alice*, a lightweight optimization testbed. Alice is intended to enable diagnostic probing of curvature effects during optimization by implementing both Hessian- and glass-based curvature approximations using ordinary gradient evaluations and inserting them into controlled training updates. This allows us to measure which curvature contributions are informative for different architectures, to test the predictions of Theorems 4–6, and to situate glass curvature alongside Hessian curvature in practical settings.

#### 4.1 ALICE AS A PROBE

By construction, Alice is minimal: it strips away auxiliary design choices in optimizers so as to isolate the effect of curvature terms. Its role is to provide a consistent baseline for comparing Hessian, glass, and combined curvature in practice. While its performance is competitive, this is incidental—the purpose is not to propose "yet another optimizer," but to demonstrate how curvature-of-expectation can be estimated and how it shapes optimization dynamics. In downstream work, more sophisticated methods may leverage these terms differently; Alice provides the measurement baseline against which such designs can be assessed.

**Empirical Questions.** Our experiments are organized to probe three aspects of the theory: (i) Do glass curvature terms improve predictive accuracy of loss changes (Theorem 4)? (ii) How do combined Hessian + glass updates affect step size and stability (Theorem 5)? (iii) Does Nesterov acceleration capture hidden linear gradient structure as predicted (Theorem 6)? All results should be read in this diagnostic light.

**Estimating Curvature.** To compute the three parameter-length quantities g,  $\rho$ , and h, Alice uses three gradient evaluations anchored at  $\nu$  to remain consistent with Theorem 6,

$$g^{(\pm)} = g(\boldsymbol{\nu} \pm \lambda \boldsymbol{\delta}) = g(\boldsymbol{\nu}) \pm \lambda \boldsymbol{H} \boldsymbol{\delta} + \boldsymbol{\gamma}^{(\pm)} \quad \text{and} \quad \boldsymbol{g}^{(0)} = g(\boldsymbol{\nu}).$$
 (14)

These capture the average linear dependencies as matrix-vector multiplies  $H\delta$ , while  $\gamma^{(\pm)}$  reveal glass density via

$$H\delta = \mathbb{E}\left[\left(g^{(+)} - g^{(-)}\right)/(2\lambda)\right],\tag{15}$$

$$\mathbf{R}|\boldsymbol{\delta}| = \mathbb{E}\left[\left(\frac{1}{2}(\boldsymbol{g}^{(+)} + \boldsymbol{g}^{(-)}) - \boldsymbol{g}^{(0)}\right)^2 / (2\lambda)\right]. \tag{16}$$

Algorithm 1 summarizes the running update. Further implementation details appear in Appendix B.

#### 4.2 EXPERIMENTAL RESULTS

**Nesterov Acceleration.** We now test whether the exactness property of Theorem 6 is observed in practice. Figure 4 examines the effect of NAQ when applied with different curvature terms:  $h_{\rm abs}$  (absolute Hessian from Line 4) and  $\rho$  (glass density from Line 5). We also include an RMS form  $h_{\rm rms}$  similar to AdaHessian (Yao et al., 2021).

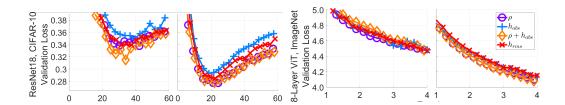


Figure 4: Alice with different curvature terms: a) full QN steps/  $\alpha=1.0,\,w=1.0,$  b) NAQ steps (Theorem 6)/  $\varphi=1.0,\,w=1.0.$ 

	Settings	7	Test Accurac	y
φ	Terms	Min	Median	Max
1.0	ρ	90.6%	91.2%	91.5%
1.0	$h_{ m abs}$	89.7%	90.1%	90.8%
1.0	$\rho + h_{\mathrm{abs}}$	91.0%	91.8%	92.7%
1.0	$h_{ m rms}$	90.5%	90.8%	91.2%
0.1	ρ	92.1%	92.6%	92.7%
0.1	$h_{ m abs}$	91.8%	91.9%	92.1%
0.1	$\rho + h_{\rm abs}$	92.1%	92.4%	92.8%
0.1	$h_{ m rms}$	91.8%	92.3%	92.7%

Settings		Test Loss		
$\overline{\varphi}$	Terms	Min	Median	Max
1.0	ρ	4.49	4.49	4.55
1.0	$h_{ m abs}$	4.45	4.49	4.73
1.0	$\rho + h_{\mathrm{abs}}$	4.41	4.48	4.48
1.0	$h_{ m rms}$	4.46	4.47	4.65
0.1	ρ	4.08	4.09	4.12
0.1	$h_{ m abs}$	4.13	4.13	4.14
0.1	$\rho + h_{\rm abs}$	4.05	4.07	4.11
0.1	$h_{ m rms}$	4.14	4.15	4.15

Table 1: ResNet18 / CIFAR-10: comparison of Table 2: ViT / Imagenet: comparison of curvature terms under Nesterov acceleration.

Our ViT model uses linear-complexity attention (Shen et al., 2021) trained on  $64 \times 64$  downsampled Imagenet (Deng et al., 2009; Chrabaszcz et al., 2017). The first column of Figure 4 shows full quasi-Newton steps; the second applies the NAQ coefficients. Results in Fig. 4 and Tables 1–2 confirm these expectations. For ResNet18 on CIFAR-10, Nesterov acceleration ( $\varphi=0.1$ ) consistently outperforms the unaccelerated case across ten random seeds, with the gradient-glass term  $\rho$  yielding the best median accuracy and the combination  $\rho+h_{\rm abs}$  achieving the best maximum accuracy. Although the RMS Hessian approximation trails slightly, its performance remains competitive. For the ViT on downsampled ImageNet, the same pattern holds: acceleration again provides a clear gain, and the combined  $\rho+h_{\rm abs}$  term delivers the lowest minimum and median losses across five seeds. Taken together, these experiments support the theoretical claim that NAQ corrects hidden linear gradient components and that glass curvature contributes measurably to improved generalization.

**Exploration and Stability.** Theorem 5 predicts that curvature-aware steps can shrink optimally, but may also overshoot if not bounded. In practice, we observe that the raw quasi-Newton step can become excessively large and exceed the range where curvature extrapolation is reliable. To stabilize training, Alice applies step-length bounds  $\lambda_{\min}$  and  $\lambda_{\max}$ , interpretable as Adam-style learning rates. Between these bounds, NAQ steps are used. Conveniently, by setting  $\varphi = \omega = 1$  and  $\lambda_{\min} = \lambda_{\max} = 1$ learn-rate recovers Adam exactly, so hyperparameters exist for which Alice will not underperform this baseline. Figure 5 illustrates this effect using the Tensorized Transformer (Ma et al., 2019) on WikiText-103 (Merity et al., 2016): increasing  $\lambda_{max}$  accelerates early loss reduction, confirming the value of stability constraints. Table 3 provides additional statistics. Results show that the stability bound  $\lambda_{\max}$  is the decisive factor: all curvature terms achieve markedly lower perplexity when  $\lambda_{\max}$ is increased from  $1.0 \times 10^{-3}$  to  $2.5 \times 10^{-3}$ . Indeed, the minimum perplexity at the smaller bound is higher than the maximum perplexity at the larger bound in nearly every case. While different curvature computations shift the relative best values—the glass term  $\rho$  attains the lowest absolute minimum and  $h_{\rm abs}$  achieves the lowest median—the dominant effect comes from the exploration limit imposed by  $\lambda_{\rm max}$ . These findings highlight that step-size safeguards are not merely a precaution but a key enabler of stable and efficient curvature exploitation.

**Method Comparisons.** Finally, to situate curvature effects in familiar training scenarios, we compare Alice against standard baselines. Figure 6 reports results on ResNet18/CIFAR-10 and Tensorized Transformer/WikiText-103. On ResNet18 (a), Alice achieves substantially lower validation loss and higher validation accuracy than SGD with momentum, Adam, and AdaHessian, with the improvement over AdaHessian being particularly pronounced. On the Tensorized Transformer (b), we

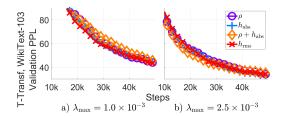


Figure 5: Effect of learning-rate bounds on stability and early loss reduction.

Settin	Validation PPL			
$\lambda_{\max}$	Terms	Min	Median	Max
$1.0 \times 10^{-3}$	ρ	40.4	43.2	49.1
$1.0 \times 10^{-3}$	$h_{ m abs}$	38.6	43.7	47.1
$1.0 \times 10^{-3}$	$\rho + h_{\mathrm{abs}}$	39.8	45.1	48.2
$1.0 \times 10^{-3}$	$h_{ m rms}$	39.4	42.9	51.2
$2.5 \times 10^{-3}$	$\rho$	28.8	33.6	35.8
$2.5 \times 10^{-3}$	$h_{ m abs}$	30.0	32.5	36.8
$2.5 \times 10^{-3}$	$\rho + h_{ m abs}$	31.0	34.8	39.1
$2.5 \times 10^{-3}$	$h_{ m rms}$	28.9	33.7	39.9

Table 3: Tensorized Transformer / WikiText-103: diagnostic comparison of curvature terms under step-length bounds.

compare Adam, AdaHessian, and Alice with glass-only and Hessian-only curvature. Here, Alice with Hessian-based curvature slightly improves over Adam, and both outperform AdaHessian.

The purpose is not to claim optimization superiority, but to demonstrate that curvature-of-expectation can be extracted consistently across architectures and may yield measurable training benefits. This diagnostic role makes the influence of glass curvature empirically visible, and establishes a baseline for future work in pruning, quantization, and curvature-aware regularization.

### 5 Broader Implications of Curvature-of-Expectation

Curvature-of-expectation extends the reach of curvature methods beyond the Hessian. In smooth regimes, the Hessian remains valid as a local quadratic surrogate. In nonsmooth regimes, glass curvature arises from cumulative gradient discontinuities. Together these form a unified picture: Hessian and glass curvature are not competing notions, but limiting cases of one framework. This perspective clarifies why Hessian-based methods succeed in some settings but fail in others, and explains the variability long observed in their empirical performance.

Compression and Pruning. Diagonal Hessian approximations are widely used to estimate the effect of weight removal. Our results suggest that glass curvature contributes an additional diagonal term that can dominate when many activation boundaries intersect a parameter's influence. This explains mismatches between pruning theory and practice, and motivates new criteria that explicitly incorporate glass terms. Quantization also introduces discontinuities, thus curvature-of-expectation may inform when quantization schemes destabilize training and how stability can be recovered.

Generalization and Flatness. Hessian eigenvalues are often used as proxies for sharpness, yet their relation to generalization remains debated. Expectation curvature offers an alternative: glass terms bound expected loss increases with a 3/2 power law, distinct from quadratic Hessian growth. This provides a sharper diagnostic of flatness in nonsmooth regions and could serve as a complementary generalization metric.

**Interpretation with Alice.** Because Alice isolates curvature contributions, it enables practitioners to test directly when glass curvature is negligible, comparable, or dominant. If glass terms vanish, Hessian-based methods suffice. If they dominate, one may prefer smoother activations or rely on the stabilizing influence of the modified quasi-Newton update. Either way, Alice provides a diagnostic baseline that translates theoretical distinctions into actionable practice.

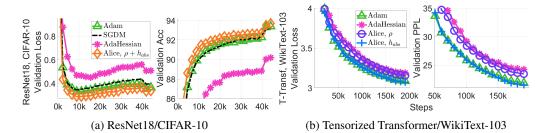


Figure 6: Comparisons with standard optimizers provide context for curvature terms. Alice is used diagnostically to test the role of Hessian vs. glass curvature.

Method	Test Accuracy		у	Method	V	alidation PI	PL
	Min	Median	Max		Min	Median	Max
Adam	93.23%	93.41%	93.73%	Adam	20.84	21.83	21.86
SGDM	92.65%	93.19%	93.75%	AdaHessian	22.12	24.23	25.79
AdaHessian	78.41%	90.25%	91.50%	Alice, $\rho$	21.07	23.43	24.95
Alice, $\rho + h_{\rm abs}$	93.30%	93.69%	94.09%	Alice, $h_{ m abs}$	18.66	21.50	23.74

Table 4: ResNet18 / CIFAR-10: diagnostic compari- Table 5: Tensorized Transformer / WikiTextson with standard optimizers.

103: curvature diagnostic comparisons.

# CONCLUSION

We developed a theoretical framework for curvature-of-expectation in nonsmooth loss landscapes, where activation-induced gradient discontinuities accumulate into a dense field, i.e. a gradient glass. Our analysis introduced: (i) a model of gradient glass and its density, (ii) optimal kernels and perturbation distributions for diagonal estimation, (iii) bounds on expected loss change with a 3/2 power law, (iv) quasi-Newton steps that unify Hessian and glass curvature, and (v) exactness conditions for Nesterov acceleration. To translate these contributions into practice, we built ALICE, a streamlined optimizer designed as an investigative probe. Alice isolates Hessian and glass curvature effects by inserting their estimates into controlled updates, thereby showing when and how curvature extraction alters training dynamics. Our experiments confirm that curvature-of-expectation is measurable, relevant, and capable of influencing stability and effective step size. In this way, Alice is not a generalpurpose optimizer but a scientific instrument: it validates that curvature-of-expectation is measurable, relevant, and capable of guiding training. By making these effects visible, Alice lays a reproducible foundation for a new research direction in optimization — one that builds on a consistent theoretical picture to enable curvature-aware methods across pruning, quantization, and regularization.

#### LIMITATIONS

Our framework and probe introduce several constraints. First, extracting curvature requires three gradient evaluations per batch. This adds overhead, though far less than full second-order backpropagation, and curvature terms evolve slowly enough to amortize cost across steps. Second, our analysis assumes gradient discontinuities from ReLU-like units are approximately uniform at the chosen perturbation scale. This captures large networks well, but smoother activations may diminish the glass effect. Third, Alice includes hyperparameters for perturbation radius and step-size bounds. These are interpretable extensions of Adam defaults and need only modest tuning, but they add knobs that must be set. These constraints reflect design tradeoffs, not conceptual limits. Alice demonstrates that glass curvature is empirically accessible, while leaving open how future optimizers may exploit it more efficiently. Together, these results show that curvature-of-expectation offers a principled and practical complement to the Hessian, revealing structure in loss landscapes that was previously hidden.

Use of LLMs. We used ChatGPT (GPT-5, OpenAI) to assist with improving the clarity and consistency of exposition, formatting LaTeX, and checking grammar. All ideas, experiments, and analyses were conceived, designed, and executed by the authors.

# REFERENCES

- Sue Becker, Yann Le Cun, et al. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, pages 29–37, 1988.
- Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.
- Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204. PMLR, 2015.
  - Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv* preprint arXiv:1707.08819, 2017.
  - Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems*, 27, 2014.
  - Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.
  - Jed A Duersch. Projective integral updates for high-dimensional variational inference. *SIAM/ASA Journal on Uncertainty Quantification*, 12(1):69–100, 2024.
  - Natalia Frumkin, Dibakar Gope, and Diana Marculescu. Jumping through local minima: Quantization in the loss landscape of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16978–16988, 2023.
  - Sobhan Hemati, Guojun Zhang, Amir Estiri, and Xi Chen. Understanding hessian alignment for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19004–19014, 2023.
  - S Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. A stochastic quasinewton method with nesterov's accelerated gradient. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*, pages 743–760. Springer, 2020.
  - Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. On the maximum hessian eigenvalue and generalization. In *Proceedings of Machine Learning Research*, pages 51–65. PMLR, 2023.
  - Yajing Kong, Liu Liu, Huanhuan Chen, Janusz Kacprzyk, and Dacheng Tao. Overcoming catastrophic forgetting in continual learning by exploring eigenvalues of hessian matrix. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
  - Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.
  - JunHoo Lee, Jayeon Yoo, and Nojun Kwak. Shot: suppressing the hessian along the optimization trajectory for gradient-based meta-learning. *Advances in Neural Information Processing Systems*, 36:61450–61465, 2023a.
- Sunwoo Lee, Chaoyang He, and Salman Avestimehr. Achieving small-batch accuracy with large-batch scalability via hessian-aware learning rate adjustment. *Neural Networks*, 158:1–14, 2023b.
- Lin Li and Michael Spratling. Understanding and combating robust overfitting via input loss landscape analysis and regularization. *Pattern Recognition*, 136:109229, 2023.
- Xin-Chun Li, Lan Li, and De-Chuan Zhan. Visualizing, rethinking, and mining the loss landscape of deep neural networks. *arXiv preprint arXiv:2405.12493*, 2024.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pages 23190–23211. PMLR, 2023.

- Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. A tensorized transformer for language modeling. *Advances in neural information processing systems*, 32, 2019.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.
- Hiroshi Ninomiya. A novel quasi-newton-based optimization for neural network training incorporating nesterov's accelerated gradient. *Nonlinear Theory and Its Applications, IEICE*, 8(4):289–301, 2017.
- Giorgio Parisi. Course 3 mean field theory of spin glasses: Statics and dynamics. *Les Houches*, 85: 131–178, 2007.
- Mrinmay Sen, C Krishna Mohan, and A Kai Qin. Federated optimization with linear-time approximated hessian diagonal. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 106–113. Springer, 2023.
- Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021.
- Stefano Spigler, Mario Geiger, Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Matthieu Wyart. A jamming transition from under-to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, 2019.
- Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In 2020 IEEE international conference on big data (Big data), pages 581–590. IEEE, 2020.
- Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10665–10673, 2021.

The authors used an AI-assisted writing tool (OpenAI's ChatGPT) for language editing and clarity improvements. All conceptual contributions, experiments, and programming were carried out by the authors.

# A PROOFS

**Proof of Theorem 1 (Glass from ReLUs).** For a unit  $k \in \mathcal{S}_{\psi}$ , write  $\Delta_k = \boldsymbol{\delta}^{\top} \hat{\boldsymbol{\gamma}}^{(k)}$ . With uniform density on  $[-\psi, \psi]$ , the probability of a flip is

$$p(\text{flip } k) = \frac{|\Delta_k|}{2\psi}.\tag{17}$$

Given  $z^{(k)} = \max(0, y^{(k)})$ , a flip changes  $\frac{\partial z^{(k)}}{\partial y^{(k)}}$  by  $\pm 1$ . By the chain rule, the gradient changes by  $\gamma^{(k)} = \pm \frac{d\mathcal{L}}{dz^{(k)}} \hat{\gamma}^{(k)}$ . By the independence assumption,

$$oldsymbol{v}_i(oldsymbol{\delta}) = \sum_{k \in \mathcal{S}_{vb}} p( ext{flip } k) \left(rac{d\mathcal{L}}{dz^{(k)}}
ight)^2 \hat{oldsymbol{\gamma}}_i^{(k)2}.$$

Finally,  $|\Delta_k| \leq \sum_j |\hat{\gamma}_j^{(k)}| |\delta_j|$  gives the stated bound in Equation (3).

**Proof of Theorem 2.** Write  $y_i = \sum_j M_{ij} \delta_j$ . Since  $\mathbb{E}[\delta_j] = 0$ ,  $\mathbb{E}[\delta_j^2] = 1$ , and  $\mathbb{E}[\delta_i \delta_j] = 0$  for  $i \neq j$ , integrating over all  $\{\delta_j\}_{j \neq i}$  gives

$$\int \{\boldsymbol{y}_i,\,\boldsymbol{y}_i^2\} \prod_{j\neq i} dp(\boldsymbol{\delta}_j) = \{\boldsymbol{m}_i\,\boldsymbol{\delta}_i,\,\boldsymbol{m}_i^2(\boldsymbol{\delta}_i^2 + \boldsymbol{\omega}_i^2)\}.$$

Consider estimators  $Z_i = \kappa_i(\boldsymbol{\delta}_i) \boldsymbol{y}_i$ . The zero-bias condition Equation (4) is equivalent to

$$\int \kappa_i(\boldsymbol{\delta}_i) \, \boldsymbol{\delta}_i \, dp(\boldsymbol{\delta}_i) = 1. \tag{18}$$

The variance is then

$$\operatorname{Var}[Z_i] = \mathbb{E}[Z_i^2] - \boldsymbol{m}_i^2 = \boldsymbol{m}_i^2 \left( \int \kappa_i(\boldsymbol{\delta}_i)^2 \left(\boldsymbol{\delta}_i^2 + \boldsymbol{\omega}_i^2\right) dp(\boldsymbol{\delta}_i) - 1 \right).$$

To minimize this subject to Equation (18), apply the variational principle. Let  $\kappa_i = \kappa_i^* + \varepsilon \eta$  where  $\eta$  satisfies  $\int \eta(\boldsymbol{\delta}_i) \, \boldsymbol{\delta}_i \, dp(\boldsymbol{\delta}_i) = 0$ . Differentiating at  $\varepsilon = 0$  yields

$$\left.\frac{\partial\operatorname{Var}[Z_i]}{\partial\varepsilon}\right|_{\varepsilon=0}=2\boldsymbol{m}_i^2\int\eta(\boldsymbol{\delta}_i)\,\kappa_i^*(\boldsymbol{\delta}_i)\,(\boldsymbol{\delta}_i^2+\boldsymbol{\omega}_i^2)\,dp(\boldsymbol{\delta}_i)=0.$$

Thus  $\kappa_i^*(\delta_i)$  ( $\delta_i^2 + \omega_i^2$ )  $\propto \delta_i$ , i.e.  $\kappa_i^*(\delta_i) \propto \delta_i/(\delta_i^2 + \omega_i^2)$ . The normalization is fixed by Equation (18). Uniqueness holds in the weighted  $L^2$  class since the objective is strictly convex under weight ( $\delta_i^2 + \omega_i^2$ ) and the constraint space is affine.

**Proof of Theorem 3.** Let  $p^*$  denote the optimal factor distribution and write a perturbation  $p(\boldsymbol{\delta}_i) = p^*(\boldsymbol{\delta}_i) + \varepsilon \, \eta(\boldsymbol{\delta}_i)$ , restricted so that  $\eta(\boldsymbol{\delta}_i) = 0$  when  $p^*(\boldsymbol{\delta}_i) = 0$ . Normalization requires

$$\int \eta(\boldsymbol{\delta}_i) \, d\boldsymbol{\delta}_i = 0. \tag{19}$$

With the optimal kernel Equation (5), the variance takes the form

$$\boldsymbol{m}_i^{-2} \operatorname{Var}[Z_i] = \int \kappa_i^*(\boldsymbol{\delta}_i)^2 (\boldsymbol{\delta}_i^2 + \boldsymbol{\omega}_i^2) dp(\boldsymbol{\delta}_i) - 1 = c^{-1} - 1,$$

where  $c=\int \frac{\boldsymbol{\delta}_i^2}{\boldsymbol{\delta}_i^2+\boldsymbol{\omega}_i^2}\,dp(\boldsymbol{\delta}_i)$ . Differentiating w.r.t.  $\varepsilon$  gives

$$\frac{\partial}{\partial \varepsilon} \operatorname{Var}[Z_i] \Big|_{\varepsilon=0} = -\boldsymbol{m}_i^2 c^{-2} \int \frac{\boldsymbol{\delta}_i^2}{\boldsymbol{\delta}_i^2 + \boldsymbol{\omega}_i^2} \, \eta(\boldsymbol{\delta}_i) \, d\boldsymbol{\delta}_i = 0.$$

Thus, on the support of  $p^*$  the function  $\frac{\delta_i^2}{\delta_i^2 + \omega_i^2}$  must be constant, since Equation (19) provides the only restriction on  $\eta$  there. This is possible only if the support consists of two symmetric points  $\{\pm x\}$ . Imposing zero mean forces symmetry, and unit variance requires x=1, giving the Rademacher distribution.

**Proof of Theorem 4.** Fix a coordinate k. Divide the path  $\mu_k \to \mu_k + \delta_k$  into n segments. Let  $\gamma_k^{(j)}$  be i.i.d. zero-mean increments with  $\mathbb{E}[(\gamma_k^{(j)})^2] = \rho_k |\delta_k|/n$  so that density  $\rho_k$  is preserved. The loss change before applying the floor is

$$\Delta_n(\boldsymbol{\delta}_k) = \sum_{j=1}^n rac{n-j}{n} \, oldsymbol{\gamma}_k^{(j)} oldsymbol{\delta}_k.$$

Each term has mean zero, so variances add:

$$\mathbb{E}[\Delta_n(\boldsymbol{\delta}_k)^2] = \sum_{i=1}^n \left(\frac{n-j}{n}\right)^2 \frac{\rho_k|\boldsymbol{\delta}_k|}{n} \boldsymbol{\delta}_k^2.$$

Taking  $n \to \infty$  gives

$$\mathbb{E}[\Delta(\boldsymbol{\delta}_k)^2] = \frac{1}{3} \, \boldsymbol{\rho}_k |\boldsymbol{\delta}_k|^3.$$

By the Lyapunov CLT,  $\Delta(\boldsymbol{\delta}_k) \sim \mathcal{N}\left(0, \; \boldsymbol{\rho}_k | \boldsymbol{\delta}_k|^3/3\right)$ . Imposing the floor  $\Delta \mathcal{L}(\boldsymbol{\delta}_k) = |\Delta(\boldsymbol{\delta}_k)|$  makes this a folded normal, whose mean is  $\sqrt{2/\pi} \, \sigma$  with variance parameter  $\sigma^2$ . Summing over coordinates yields the bound in Equation (6).

**Proof of Theorem 5.** The bound separates by coordinate k. Stationarity requires

$$0 = \boldsymbol{g}_k + \boldsymbol{h}_k \boldsymbol{\delta}_k + \sqrt{\frac{3\boldsymbol{\rho}_k}{2\pi}} \operatorname{sign}(\boldsymbol{\delta}_k) |\boldsymbol{\delta}_k|^{1/2}.$$

To enforce descent, set  $\delta_k = -\operatorname{sign}(\boldsymbol{g}_k)x^2$  with x > 0. Substituting  $\hat{\boldsymbol{h}}_k = 3\boldsymbol{\rho}_k/(4\pi|\boldsymbol{g}_k| + \varepsilon)$  gives

$$0 = -\sqrt{\frac{|\boldsymbol{g}_k|}{2}} + \sqrt{\hat{\boldsymbol{h}}_k} \, x + \frac{\boldsymbol{h}_k}{\sqrt{2|\boldsymbol{g}_k|}} \, x^2.$$

Solving the quadratic (positive root) yields

$$x = \frac{\sqrt{2|\boldsymbol{g}_k|}}{\sqrt{\hat{\boldsymbol{h}}_k} + \sqrt{\hat{\boldsymbol{h}}_k + 2\boldsymbol{h}_k}}.$$

Thus  $\delta_k = -g_k/(\hat{h}_k + h_k + \sqrt{\hat{h}_k(\hat{h}_k + 2h_k)} + \varepsilon)$ , which matches Equation (7).

**Proof of Theorem 6.** From Equation (8) and Equation (10),

$$g(\boldsymbol{\mu}^{(s+1)}) = \boldsymbol{g}^{(s)} + \varphi \bar{\boldsymbol{h}} * \boldsymbol{\delta}^{(s)} = \boldsymbol{g}^{(s)} - (\varphi) \boldsymbol{g}^{(s)} = (1 - \varphi) \boldsymbol{g}^{(s)}.$$

Thus setting  $\varphi = 1 - \beta_1$  makes  $g(\boldsymbol{\mu}^{(s+1)}) = \beta_1 \boldsymbol{g}^{(s)}$ , consistent with Equation (9). Next, substitute the evaluation point  $\omega = 1$  into Equation (11), combine with the error decomposition Equation (13), and place the result in Equation (9):

$$\mathbf{g}^{(s+1)} = \beta_1 \mathbf{g}^{(s)} + (1 - \beta_1) (\mathbf{g}^{(s)} - \boldsymbol{\gamma}^{(s)} + \boldsymbol{H} \boldsymbol{\delta}^{(s)})$$
$$= \mathbf{g}^* (\boldsymbol{\mu}^{(s+1)}) + \beta_1 \boldsymbol{\gamma}^{(s)}.$$

This shows that the update both tracks the hidden linear structure and contracts errors by  $\beta_1$ .

# B METHODOLOGICAL DETAILS

**Algorithms.** Algorithm 2 shows the full topography update in a memory-efficient form using one temporary variable. Line 6 additionally enables an RMS Hessian approximation: taking the square root of the running second moment yields a non-negative diagonal suitable for quasi-Newton steps. This is similar to AdaHessian (Yao et al., 2021), except that we maintain per-parameter statistics instead of block averages. Algorithm 3 then applies quasi-Newton updates combining glass and Hessian diagonals, with bounds for stability.

# Algorithm 2 Alice Diagnostic Topography Update

**Input**: evaluation center  $\nu$ ; gradient function  $g(\theta)$ .

**Input/Output**: running averages g,  $\rho$ ,  $h_{\rm abs}$ ,  $h_{\rm rms}^2$ , and s.

**Hyperparameters**:  $\lambda, \beta_1, \beta_2$ .

Implements Eqs. 14–16 to adapt Theorems 1–3, requiring only one parameter-length temporary variable t.

- 1: Draw Rademacher t. Set  $\theta \leftarrow \nu + \lambda t$ .
- 2: Evaluate  $q(\boldsymbol{\theta}) = \boldsymbol{q}^{(+)}$ .

702

703

704

705

706

708

709

710

711

716

717

718

719 720 721

722

725

726 727

728

729

730

731

732

733

735

736 737 738

739 740

741

742 743 744

745

746 747 748

749

750 751 752

753

754

755

- 3: Set  $\theta \leftarrow \nu \lambda t$ . Store  $t \leftarrow g^{(+)}$ .
- 4: Evaluate  $g(\boldsymbol{\theta}) = \boldsymbol{g}^{(-)}$ .

  - 5:  $h_{\mathrm{abs}} \leftarrow \beta_2 h_{\mathrm{abs}} + (1 \beta_2) \frac{1}{2\lambda} |t g^{(-)}|.$ 6:  $h_{\mathrm{rms}}^2 \leftarrow \beta_2 h_{\mathrm{rms}}^2 + (1 \beta_2) \frac{1}{4\lambda^2} (t g^{(-)})^2.$ 7: Store  $t \leftarrow \frac{1}{2} (t + g^{(-)}).$  Set  $\theta \leftarrow \mu.$

  - 8: Evaluate  $q(\boldsymbol{\theta}) = \boldsymbol{q}^{(0)}$
  - 9:  $\mathbf{g} \leftarrow \beta_1 \mathbf{g} + (1 \beta_1) \mathbf{g}^{(0)}$ .
  - 10:  $\rho \leftarrow \beta_2 \rho + (1 \beta_2) \frac{2}{\lambda} (t g^{(0)})^2$ .
  - 11:  $\mathbf{s} \leftarrow \beta_2 \mathbf{s} + (1 \beta_2) \mathbf{g}^{(0)2}$ .

# Algorithm 3 Alice Quasi-Newton Step with Diagnostic Curvature Terms

**Input**: running averages g,  $\rho$ , h, s.

723 **Input/Output**: parameters  $\mu$ . 724

**Output**: evaluation center  $\nu$ .

Hyperparameters: limit-method,  $\lambda, \lambda_{\min}, \lambda_{\max}, \varepsilon$ .

Implements Theorems 4, 5, and 6, with stability enforced by Adam-style limits.

- 1: Compute glass term  $\hat{h} = \frac{3}{4\pi} \rho * (|g| + \varepsilon)^{-1}$ .
- 2: Compute modified Hessian  $\bar{h} = \hat{h} + h + \sqrt{\hat{h} * (\hat{h} + 2h)} + \varepsilon$ .
- 3: Quasi-Newton scale  $\delta = |\mathbf{g}| * \bar{\mathbf{h}}^{-1}$ .
- Fixed scale:  $\delta_{\min/\max} = \lambda_{\min/\max}$ ; SGD-M:  $\delta_{\min/\max} = \lambda_{\min/\max} |g|$ ; 4: Apply limits: Adam:  $\delta_{\min/\max} = \lambda_{\min/\max} |g| * (\sqrt{s} + \varepsilon)^{-1}$ .
- 5: Enforce bounds:  $\delta \leftarrow \max(\delta_{\min}, \min(\delta_{\max}, \delta))$ .
- 6: Correct sign:  $\delta \leftarrow -\operatorname{sign}(g)\delta$ .
- 7: Update evaluation center  $\nu \leftarrow \mu + \omega \delta$ .
- 8: Update parameters  $\mu \leftarrow \mu + \varphi \delta$ .

# EXPERIMENTAL SETUP

We evaluate Alice as a diagnostic probe on standard benchmarks. Results are reported as minimum, median, and maximum values over multiple seeds, showing both variability and best-case outcomes.

**Hardware and Software.** Experiments were run on compute nodes with AMD 32-core CPUs and one NVIDIA A100 GPU (40GB memory). Code is implemented in PyTorch 2.2.1 (BSD 3-Clause License) with CUDA 11.8.

Licenses. PyTorch: BSD 3-Clause. ResNet: BSD 3-Clause. Tensorized Transformer and CIFAR-10: MIT. WikiText-103 (and Wikipedia): CC BY-SA 3.0. ImageNet: non-commercial research/educational use only.

**ResNet18, Power Law.** Figure 2 uses a specialized Alice variant with additional memory and evaluations to store gradient variations at  $\lambda$  and  $2\lambda$ . We train a modified ResNet18 with multiplicative channel masks, improving prediction quality for SGDM, Adam, and Alice. Training: 40 epochs,  $\lambda = 0.002, \, \beta_1 = 0.9, \, \beta_2 = 0.999, \, \varepsilon = 10^{-8}, \, \varphi = 0.1, \, \omega = 1.0, \, \lambda_{\min} = 0, \, \lambda_{\max} = 0.002,$ quick\_steps = 0, Adam-based limiting. Curvature terms:  $\rho$  and  $h_{abs}$ .

**ResNet18, NAQ.** Figure 4: ResNet18 with tuned  $\lambda=0.005$  and fixed-scale limiting ( $\lambda_{\rm max}=5\times10^{-3}$ ). Grid search over  $\{0.001,0.002,0.005,0.01,0.02,0.05\}$  and limiting strategies. 10 seeds per setting.

 **ViT, NAQ.** Vision Transformer with 8 layers, linear-complexity attention (Shen et al., 2021), Imagenet-64x64. Training: 4 epochs,  $\lambda = 0.005$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\varepsilon = 10^{-8}$ ,  $\lambda_{\min} = 0$ ,  $\lambda_{\max} = 0.01$ , quick\_steps = 3, Adam-based limiting. 5 seeds per setting.

**Tensorized Transformer, Stability Limit.** Tensorized Transformer (Ma et al., 2019), dropout 0.1, batch size 60, 50k steps. Settings:  $\lambda = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\varepsilon = 10^{-8}$ ,  $\varphi = 0.1$ ,  $\omega = 1.0$ , quick\_steps = 3, Adam-based limiting.

**ResNet18, Training Methods.** For Figure 6: Adam (λ = 0.001, β<sub>1</sub> = 0.9, β<sub>2</sub> = 0.999, ε =  $10^{-8}$ ); SGD-M (λ = 0.001, β = 0.9); AdaHessian (λ = 0.15, tuned from {0.1, 0.15, 0.2, 0.25, 0.3}); Alice (λ = 0.005, β<sub>1</sub> = 0.9, β<sub>2</sub> = 0.999, ε =  $10^{-8}$ ,  $\varphi$  = 0.1,  $\omega$  = 1.0,  $\lambda_{\min}$  = 0,  $\lambda_{\max}$  = 0.01, quick\_steps = 3, Adam-based limiting). Curvature:  $\rho$  and  $h_{abs}$ .

**Tensorized Transformer, Training Methods.** For Figure 6: 200k steps. Adam and Alice:  $\lambda = 0.00025, \ \beta_1 = 0.9, \ \beta_2 = 0.999, \ \varepsilon = 10^{-8}$ . Alice also:  $\varphi = 1.0, \ \omega = 1.0, \ \lambda_{\max} = 0.000375,$  quick\_steps = 1, Adam-based limiting. AdaHessian:  $\lambda = 0.00025$ .