A Simple and Efficient Measure of Loss Landscape Curvature

Hee-Sung Kim MUWONIJR@HANYANG.AC.KR
Department of Computer Science, Hanyang University, Seoul, Korea

Sungyoon Lee SUNGYOONLEE@HANYANG.AC.KR
Department of Computer Science, Hanyang University, Seoul, Korea

Abstract

The Edge of Stability characterizes training dynamics through a sharpness measure — the Hessian eigenvalue for GD, a preconditioned variant for adaptive methods — reaching an optimizer-dependent threshold, yet measuring this quantity is prohibitive at scale and incompatible with modern training kernels. We revisit directional curvature along the optimizer's update direction as a scalable alternative, and show that its one-step descent boundary stays at $2/\eta$ across optimizers — including momentum and adaptive methods — by absorbing the optimizer state into the update direction rather than the threshold. To make this measure practical, we introduce two forward-pass estimators that avoid Hessian-vector products: a symmetric finite-difference estimator and a KL-divergence estimator requiring only one or two extra forward passes per step. On full-batch CIFAR-10 training under GD and AdamW, both estimators reproduce progressive sharpening and oscillate around the predicted boundary, with the KL-based estimator tracking the reference curvature cleanly at lower cost.

1. Introduction

The curvature of the loss landscape governs the stability of neural network training. Under full-batch gradient descent, the largest Hessian eigenvalue $\lambda_{\max}^{H}$ rises during training until it reaches the boundary $2/\eta$, after which the loss enters the *Edge of Stability* (EoS) regime — sharpness hovers near $2/\eta$ while the loss decreases non-monotonically [1, 3, 16]. Tracking how curvature evolves relative to this boundary has become a standard lens for understanding training dynamics.

In practice, however, $\lambda_{\max}^{H}$ is increasingly difficult to measure. Power iteration and Lanczos require many Hessian-vector products, which are costly at LLM scale and incompatible with high-throughput training kernels (e.g., FlashAttention) that do not expose the second derivatives needed for double backpropagation. This makes Hessian sharpness an unattractive signal for routine curvature monitoring during training.

A scalable alternative is to measure curvature only along the direction $\Delta\theta$ that the optimizer actually takes. Directional sharpness along $\Delta\theta$ requires only the scalar $\Delta\theta^\top H \Delta\theta$ rather than the full Hessian spectrum [2, 9, 14], and a natural derivation from the *one-step descent condition* yields the curvature

$$\lambda_{\mathrm{dir}} = \frac{\Delta\theta^\top H \Delta\theta}{g^\top \Delta\theta}, \tag{1}$$

whose vanishing-descent boundary is $2/\eta$. The remaining practical question is whether λ_{dir} itself can be estimated without second-order autodiff.

© H.-S. Kim & S. Lee.

This work. We focus on making λ_{dir} a cheap, training-time signal. We introduce two forward-pass estimators that avoid Hessian-vector products entirely: a symmetric finite-difference estimator (two extra forward passes) and a KL-divergence estimator (one extra forward pass when base logits are cached). Both estimators are compatible with modern training kernels and require no double-backpropagation. Empirically, on full-batch CIFAR-10 training with SGD and Adam, both estimators rise during training and oscillate around $2/\eta$, recovering the progressive-sharpening / EoS picture through a forward-only measurement.

The $2/\eta$ boundary attached to λ_{dir} should be read as a one-step descent threshold, complementary to — and not in conflict with — the long-horizon iterate-stability thresholds of momentum and adaptive EoS analyses [1, 2].

Contributions.

- **Optimizer-independent $2/\eta$ boundary for λ_{dir}.** Derived from the one-step descent condition, $\lambda_{\text{dir}} = \Delta\theta^\top H \Delta\theta / (g^\top \Delta\theta)$ retains the $2/\eta$ boundary across optimizers (SGD, momentum, Adam), since momentum and adaptive preconditioning are absorbed into the update direction $\Delta\theta$. This distinguishes λ_{dir} from classical Hessian sharpness and preconditioned-Hessian sharpness, whose stability thresholds shift with optimizer hyperparameters (e.g., $(2 + 2\beta)/\eta$ for heavy-ball momentum).

- **Forward-only estimators of λ_{dir}.** Two estimators that avoid Hessian-vector products — a symmetric finite-difference estimator (λ_{FD}, two extra forward passes) and a KL-divergence estimator (λ_{KL}, one extra forward pass with cached logits) — both compatible with modern training kernels.

A stochastic generalization to mini-batch training, which subsumes the Interaction-Aware Sharpness of Lee and Jang [7], and a finer comparison with optimizer-specific dynamical thresholds are deferred to the appendix.

2. Related Work

Hessian sharpness and EoS. The top Hessian eigenvalue $\lambda_{\max}^H$ is the canonical curvature measure for analyzing training dynamics. Cohen et al. [1] showed that under full-batch GD, $\lambda_{\max}^H$ rises until it reaches $2/\eta$ and then hovers around this boundary, a regime later explained analytically via self-stabilization [3]. Direct measurement, however, requires Hessian-vector products and is prohibitive at scale, both due to iterative eigensolvers and the incompatibility of double backpropagation with high-throughput training kernels.

Directional sharpness. A scalable alternative measures curvature only along the optimizer's update direction $\Delta\theta$: the directional curvature $\lambda_{\text{dir}} = \Delta\theta^\top H \Delta\theta / (\Delta\theta^\top g)$, studied for GD [9] and extended to preconditioned Hessians for adaptive optimizers [2]. Roulet et al. [14] adopt a more general form of this measure as a learning-rate tuning signal, carefully accounting for the possible misalignment between g and $\Delta\theta$ (so that the quantity need not be positive), and compute its second-order term $\Delta\theta^\top H \Delta\theta$ without Hessian-vector products through forward-over-forward mode autodiff. The closest *autodiff-free* approach, relying on forward passes alone, instead estimates a critical learning rate η_c via a two-phase line search costing up to $\sim$10 forward passes per step [4]; we probe the second-order Taylor coefficient along $\Delta\theta$ directly, recovering λ_{dir} in 1–2 forward passes.

Optimizer-specific stability thresholds. A separate line of work treats the optimizer state (parameters together with momentum buffers) as a dynamical system and asks when the iteration itself is stable. This yields momentum-dependent thresholds such as $(2 + 2\beta)/\eta$ for heavy-ball momentum [1] and preconditioner-dependent boundaries for Adam [2]. These describe long-horizon iterate stability and are complementary to, not in conflict with, the one-step descent condition we analyze.

3. Directional Curvature in Full-Batch Training

We now make precise the curvature notion underlying our forward-pass estimators. Two questions guide the derivation: (i) what is the natural scalar curvature to associate with an arbitrary optimizer update $\Delta\theta$, and (ii) what threshold does this curvature meet at the boundary of stability? Answering both from a single Taylor expansion recovers the *directional curvature* (following [9, 14]), whose definition adapts to the optimizer while the threshold stays fixed at $2/\eta$.

3.1. One-Step Descent Analysis

Let $L(\theta)$ denote the objective, $g = \nabla L(\theta)$ the gradient, and $H = \nabla^2 L(\theta)$ the Hessian. Consider any optimizer producing a parameter update $\theta_{\text{new}} = \theta - \eta\,\Delta\theta$, where $\Delta\theta$ is the effective update direction — incorporating momentum, adaptive preconditioning, or both — and η is the global learning rate. A second-order expansion of the loss along $\Delta\theta$ gives

$$\Delta L \;=\; L(\theta_{\text{new}}) - L(\theta) \;=\; -\eta\, g^\top \Delta\theta \;+\; \tfrac{1}{2}\,\eta^2\, \Delta\theta^\top H \Delta\theta \;+\; O(\eta^3). \tag{2}$$

For descent directions ($g^\top \Delta\theta > 0$), the boundary $\Delta L = 0$ marks the smallest η at which one-step descent ceases. Solving for the corresponding curvature yields the *directional curvature*:

$$\boxed{\lambda_{\text{dir}} \;=\; \frac{\Delta\theta^\top H \Delta\theta}{g^\top \Delta\theta} \;\approx\; \frac{2}{\eta}.} \tag{3}$$

The numerator captures the second-order curvature cost along the chosen direction; the denominator captures the first-order descent gain — the projection of the gradient onto the optimizer update.

3.2. Threshold Behavior Across Optimizers

Because momentum and preconditioning are absorbed into $\Delta\theta$ rather than the threshold, (3) yields the same boundary $2/\eta$ regardless of optimizer. For full-batch SGD, $\Delta\theta = g$ and λ_{dir} reduces to the standard directional sharpness

$$\lambda_{\text{dir}}^{\text{SGD}} \;=\; \frac{g^\top H g}{g^\top g}.$$

For momentum and adaptive optimizers, $\Delta\theta$ is no longer aligned with g, but the same Taylor expansion applies and the boundary stays at $2/\eta$.

Contrast with Hessian and preconditioned-Hessian sharpness. Under classical Hessian sharpness $\lambda_{\max}^{H}$, the stability threshold itself moves with the optimizer: $2/\eta$ for plain GD, $(2 + 2\beta)/\eta$ for heavy-ball momentum [1], and a state-dependent boundary on a preconditioned Hessian for Adam [2]. The directional measure shifts the optimizer dependence from the threshold to the measured quantity: λ_{dir} changes with the optimizer through $\Delta\theta$, while $2/\eta$ is held fixed. The two pictures answer different questions — long-horizon iterate stability versus one-step loss descent — and are complementary rather than in conflict.

Extension to stochastic updates. When $\Delta\theta$ is random, applying the same Taylor argument in expectation yields a stochastic generalization with an analogous $2/\eta$ boundary, connecting to the Interaction-Aware Sharpness of Lee and Jang [7]. As we focus on full-batch setup, we defer the derivation to Appendix C.

4. Forward-Pass Estimators

We estimate $\Delta\theta^\top H \Delta\theta$ from forward evaluations along $\Delta\theta$, sidestepping the Hessian-vector products that are unavailable in high-throughput training kernels.

4.1. Symmetric Finite-Difference Estimator

A symmetric perturbation around θ cancels the odd-order terms of the Taylor expansion:

$$L(\theta + \epsilon\,\Delta\theta) + L(\theta - \epsilon\,\Delta\theta) \;=\; 2L(\theta) + \epsilon^2\,\Delta\theta^\top H \Delta\theta \;+\; O(\epsilon^4), \tag{4}$$

yielding a second-order accurate estimator

$$\lambda_{\mathrm{FD}} \;\triangleq\; \frac{L(\theta + \epsilon\,\Delta\theta) - 2L(\theta) + L(\theta - \epsilon\,\Delta\theta)}{\epsilon^2\, g^\top \Delta\theta} \;=\; \lambda_{\mathrm{dir}} + O(\epsilon^2) \tag{5}$$

at the cost of two extra forward evaluations per step. The probe ϵ trades higher-order Taylor error against floating-point cancellation in the numerator.

4.2. KL-Divergence Estimator

Under cross-entropy loss, the local curvature of the predictive distribution $p_\theta(\cdot|x)$ is captured by the Fisher information matrix (FIM)

$$F = \mathbb{E}_x\left[\mathbb{E}_{y\sim p_\theta(\cdot|x)}\left[-\nabla_\theta^2 \log p_\theta(y|x)\right]\right], \tag{6}$$

which coincides with the Gauss-Newton component of the Hessian and dominates near interpolating optima [5, 6, 8, 10–13, 15]; see Appendix D for the precise relation between Hessian and FIM. The expected KL divergence under a small parameter perturbation expands as

$$\mathbb{E}_x[D_{\mathrm{KL}}(p_\theta(\cdot|x)\,\|\,p_{\theta+\epsilon\Delta\theta}(\cdot|x))] \;=\; \tfrac{1}{2}\,\epsilon^2\,\Delta\theta^\top F \Delta\theta \;+\; O(\epsilon^3), \tag{7}$$

giving

$$\lambda_{\mathrm{KL}} \;\triangleq\; \frac{2\,\mathbb{E}_x[D_{\mathrm{KL}}(p_\theta(\cdot|x)\,\|\,p_{\theta+\epsilon\Delta\theta}(\cdot|x))]}{\epsilon^2\, g^\top \Delta\theta} \;=\; \frac{\Delta\theta^\top F \Delta\theta}{g^\top \Delta\theta} \;+\; O(\epsilon). \tag{8}$$

With the base logits $p_\theta(\cdot|x)$ already cached from the training step, this requires only one extra forward pass — a Fisher analogue of λ_{dir} at half the cost.

5. Experiments and Discussion

We validate λ_{FD} and λ_{KL} on ResNet-9 (without batch normalization) trained on CIFAR-10 with *full-batch* updates and a constant learning rate, comparing GD and AdamW (β_1=0.9, β_2=0.999). At each step we record three measurements along the optimizer's update direction $\Delta\theta$: the exact λ_{dir} (computed with one HVP, as reference), λ_{FD}, and λ_{KL}. The probe scale ϵ is fixed across runs; full hyperparameters and additional results on VGG are deferred to Appendix A.

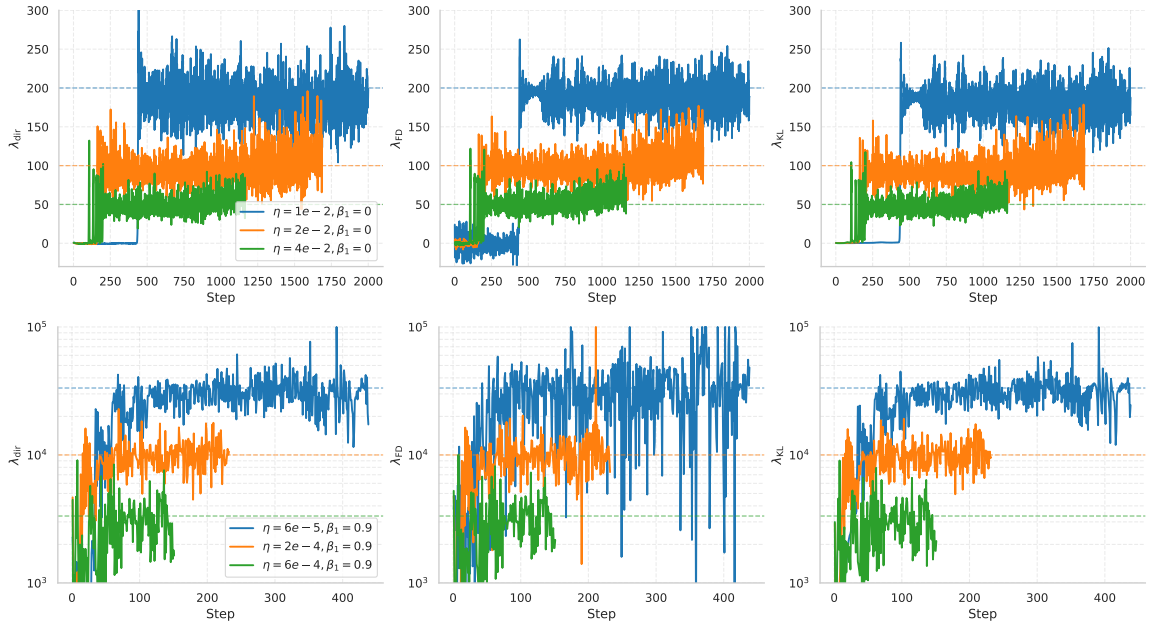

Figure 1: Directional curvature on full-batch CIFAR-10 with ResNet-9 (no BN). **Top:** GD. **Bottom:** AdamW (β_1=0.9, β_2=0.999). **Left:** exact λ_{dir} (HVP). **Middle:** λ_{FD}. **Right:** λ_{KL}. All three measurements rise during training and oscillate around $2/\eta$ for both optimizers.

Both estimators track $2/\eta$ across optimizers. Under both GD and AdamW, all three measurements exhibit the progressive-sharpening signature: the curvature rises during training and then oscillates around $2/\eta$, consistent with the Edge-of-Stability picture. Notably, the same $2/\eta$ comparison holds for AdamW *without* any optimizer-specific threshold adjustment, empirically supporting the optimizer-independent reading of λ_{dir}.

λ_{FD} vs. λ_{KL}. λ_{FD} captures the qualitative trend of λ_{dir} but is visibly noisier — its symmetric loss-difference numerator amplifies floating-point cancellation near $2L(\theta)$. λ_{KL} is markedly cleaner, tracks λ_{dir} more faithfully throughout training, and costs half as many forward passes, making it the preferable default for cross-entropy objectives.

6. Conclusion

We studied directional curvature $\lambda_{\text{dir}} = \Delta\theta^\top H \Delta\theta / (g^\top \Delta\theta)$ as a scalable replacement for Hessian sharpness. Derived from a one-step descent condition, it keeps the $2/\eta$ boundary across GD, momentum, and AdamW: the optimizer state enters through $\Delta\theta$, not the threshold. We proposed two forward-pass estimators, λ_{FD} and λ_{KL}, both avoiding Hessian-vector products. On full-batch CIFAR-10, both reproduce progressive sharpening and oscillate around $2/\eta$ under GD and AdamW, with λ_{KL} matching the reference signal at half the cost.

The remaining questions are largely scale and precision. Validating λ_{KL} on mid- and LLM-scale training is the clearest next step, since that is where HVP-based monitoring breaks down. A stochastic extension with the same $2/\eta$ boundary is straightforward (Appendix C) and links to

Lee and Jang [7], but waits on mini-batch experiments. At lower precision the ϵ window for λ_{FD} tightens, inviting adaptive-ϵ or mixed-precision probing; λ_{KL} avoids this but assumes cached logits. A concrete downstream use is curvature-aware learning-rate tuning [14], where swapping the HVP-based sharpness for λ_{KL} would make the tuner deployable at scales it currently cannot reach.

References

[1] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jh-rTtvkGeM.

[2] Jeremy M. Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E. Dahl, and Justin Gilmer. Adaptive gradient methods at the edge of stability, 2022. URL https://arxiv.org/abs/2207.14484.

[3] Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nhKHA59gXz.

[4] Dayal Singh Kalra, Jean-Christophe Gagnon-Audet, Andrey Gromov, Ishita Mediratta, Kelvin Niu, Alexander H Miller, and Michael Shvartsman. A scalable measure of loss landscape curvature for analyzing the training dynamics of llms, 2026. URL https://arxiv.org/abs/2601.16979.

[5] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1032–1041. PMLR, 2019.

[6] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Pathological spectra of the fisher information metric and its variants in deep neural networks. *Neural Computation*, 33(8):2274–2307, 07 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01411. URL https://doi.org/10.1162/neco_a_01411.

[7] Sungyoon Lee and Cheongjae Jang. A new characterization of the edge of stability based on a sharpness measure aware of batch gradient distribution. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=bH-kCY6LdKg.

[8] James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL http://jmlr.org/papers/v21/17-678.html.

[9] Yan Pan and Yuanzhi Li. Toward understanding why adam converges faster than sgd for transformers, 2023. URL https://arxiv.org/abs/2306.00204.

[10] Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size, 2018. URL https://arxiv.org/abs/1811.07062.

[11] Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5012–5021. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/papyan19a.html.

[12] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020. URL http://jmlr.org/papers/v21/20-933.html.

[13] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks, 2014. URL https://arxiv.org/abs/1301.3584.

[14] Vincent Roulet, Atish Agarwala, Jean-Bastien Grill, Grzegorz Michal Swirszcz, Mathieu Blondel, and Fabian Pedregosa. Stepping on the edge: Curvature aware learning rate tuners. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=SEflLHIhhJ.

[15] Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks, 2018. URL https://arxiv.org/abs/1706.04454.

[16] Lei Wu, Chao Ma, and Weinan E. How SGD Selects the Global Minima in Over-parameterized Learning: A Dynamical Stability Perspective. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Appendix A. Experimental Setup

Data. All experiments use a 5k subset of CIFAR-10 (the first $5{,}000$ examples). Each channel is normalized by the dataset-wide mean and standard deviation. No data augmentation is applied. The same fixed subset is used as the full batch in every step.

Models. We use two CNN architectures: **ResNet-9 without batch normalization** (main text) and **VGG-11** (this appendix). Both networks are trained from scratch with the default PyTorch initialization.

Optimizers. We compare two optimizers in the full-batch regime:

- **GD**: vanilla gradient descent with no momentum.
- **AdamW**: β_1=0.9, β_2=0.999.

Each optimizer is run with a constant learning rate over three values: $\eta \in \{2/50,\ 2/100,\ 2/200\}$ for GD, and $\eta \in \{6\times10^{-5},\ 2\times10^{-4},\ 6\times10^{-4}\}$ for AdamW. For GD these are chosen so that the predicted boundary $2/\eta \in \{50, 100, 200\}$ falls within the curvature range observed during training; for AdamW the values are selected to elicit the analogous Edge-of-Stability behavior under the adaptive update direction $\Delta\theta_{\mathrm{AdamW}}$. Each run is terminated when the training loss first reaches a fixed target threshold, rather than at a fixed step count, so that curves at different learning rates are compared at matched loss levels.

Curvature measurements. At every training step we record three quantities along the optimizer's update direction $\Delta\theta$:

- λ_{dir} — the exact directional curvature, computed with a single Hessian-vector product.
- λ_{FD} — symmetric finite-difference estimator, two extra forward passes per step.
- λ_{KL} — KL-divergence estimator, one extra forward pass per step (base logits cached from the training forward pass).

The probe scale ϵ is fixed across all runs and both estimators.

Appendix B. Additional Experiments: VGG-11

To check that the behavior reported in the main text is not specific to ResNet-9, we repeat the same protocol on VGG-11 under the identical full-batch CIFAR-10 setup.

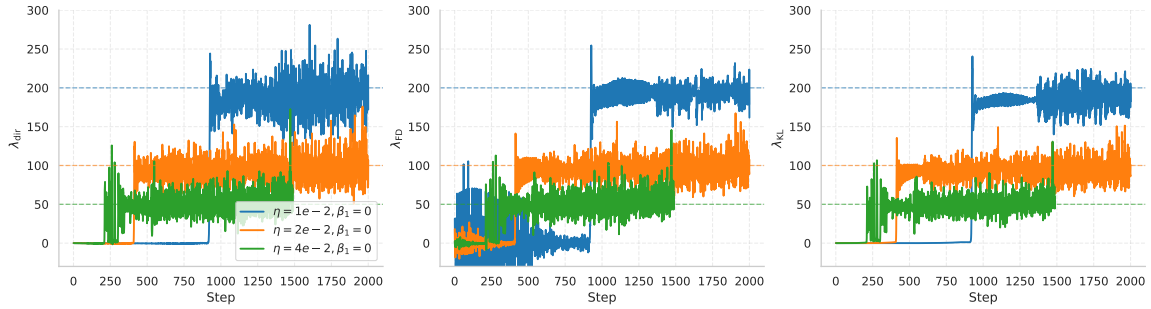

Figure 2: Directional curvature on VGG-11, full-batch CIFAR-10, GD with constant learning rate. Both estimators rise during training and oscillate around the predicted boundary $2/\eta$ (dashed horizontal lines), matching the ResNet-9 results in the main text.

Appendix C. Stochastic Generalization to Mini-Batch Training

When $\Delta\theta$ is random — owing to mini-batch sampling, momentum state initialised from past stochastic gradients, or sampled preconditioning — the one-step descent argument of Section 3 carries through in expectation. Taking the expectation of the second-order Taylor expansion over the stochasticity of $\Delta\theta$ yields

$$\mathbb{E}[\Delta L] \;\approx\; -\eta\, g^\top \mathbb{E}[\Delta\theta] \;+\; \tfrac{1}{2}\,\eta^2\, \mathrm{tr}\Big(H\, \mathbb{E}[\Delta\theta\, \Delta\theta^\top]\Big)\,. \tag{9}$$

Setting the expected one-step progress $\mathbb{E}[\Delta L] = 0$ gives the *Generalized Directional Curvature*

$$\lambda_{\text{gen}} \;\triangleq\; \frac{\mathrm{tr}\big(H\, \mathbb{E}[\Delta\theta\, \Delta\theta^\top]\big)}{g^\top \mathbb{E}[\Delta\theta]} \;\approx\; \frac{2}{\eta}. \tag{10}$$

The denominator captures the expected first-order descent (expected gain), while the numerator encapsulates the expected second-order curvature cost, explicitly accounting for the interaction between the Hessian geometry and the non-isotropic variance of the update vector.

Recovery of the deterministic case. Under full-batch GD the stochasticity vanishes, $\mathbb{E}[\Delta\theta\, \Delta\theta^\top] = g\, g^\top$, and (10) reduces to $\lambda_{\text{dir}} = g^\top H g/(g^\top g)$, the classical directional sharpness used in Section 3.

Recovery of Interaction-Aware Sharpness. For mini-batch SGD with batch gradient $\hat{g}$, (10) specialises to $\mathrm{tr}(H\, \mathbb{E}[\hat{g}\, \hat{g}^\top])/(g^\top \mathbb{E}[\hat{g}])$, which is precisely the Interaction-Aware Sharpness of Lee and Jang [7]. Our derivation extends this to arbitrary stochastic update rules, including momentum and adaptively-preconditioned variants.

Appendix D. Hessian–Fisher Decomposition for Cross-Entropy Loss

This appendix gives the precise relationship between the Hessian $H = \nabla_\theta^2 L$ and the Fisher information matrix F used in the KL-divergence estimator of Section 4, and identifies the regime in which λ_{KL} tracks λ_{dir}. The decomposition below is standard [8, 13]; we restate it here in the notation of the main text.

D.1. Setup

Consider a classification model with logits $z(x;\theta) \in \mathbb{R}^C$ and predictive distribution $f(x;\theta) = \mathrm{softmax}(z(x;\theta))$. Let $J(x;\theta) := \nabla_\theta z(x;\theta) \in \mathbb{R}^{C\times m}$ denote the logit Jacobian. The per-sample cross-entropy loss is $\ell_i(\theta) = -\log p_\theta(y_i \mid x_i)$, and the empirical loss is $L(\theta) = \frac{1}{n}\sum_{i=1}^n \ell_i(\theta)$.

D.2. Per-Sample Hessian–Gauss-Newton Identity

Differentiating ℓ_i twice through the softmax–CE composition yields, for each sample i,

$$\nabla_\theta^2 \ell_i \;=\; \underbrace{J_i^\top\big(\mathrm{diag}(f_i) - f_i f_i^\top\big) J_i}_{G_i(\theta) \,:=\, \text{Gauss-Newton term}} \;+\; \underbrace{\sum_{k=1}^{C} (f_i - y_i)_k \, \nabla_\theta^2 z_k(x_i;\theta)}_{R_i(\theta) \,:=\, \text{residual term}}, \tag{11}$$

where $f_i := f(x_i;\theta)$ and y_i is the one-hot label. The first term G_i depends only on the model's predictions; the residual R_i is a sum weighted by the prediction errors $(f_i - y_i)$.

Averaging over the dataset,

$$H(\theta) \;=\; G(\theta) \;+\; R(\theta), \qquad G(\theta) := \frac{1}{n}\sum_{i=1}^n G_i(\theta), \quad R(\theta) := \frac{1}{n}\sum_{i=1}^n R_i(\theta). \tag{12}$$

D.3. Equivalence of Gauss-Newton and Empirical Fisher

For softmax–CE, the Gauss-Newton matrix $G(\theta)$ is exactly the *empirical* Fisher information matrix:

$$G(\theta) \;=\; \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{y\sim p_\theta(\cdot\mid x_i)}\Big[\nabla_\theta \log p_\theta(y \mid x_i) \nabla_\theta \log p_\theta(y \mid x_i)^\top\Big] \;=\; F(\theta). \tag{13}$$

The identity follows because the per-sample logit Hessian of CE, $\nabla_z^2 \ell_i = \mathrm{diag}(f_i) - f_i f_i^\top$, is exactly the covariance of $\nabla_z \log p_\theta(y \mid x_i)$ under $y \sim p_\theta(\cdot \mid x_i)$. Equation (13) is the reason F — rather than the full Hessian — naturally arises in the second-order expansion of the KL divergence (Section 4).

Combining (12) and (13),

$$\boxed{H(\theta) \;=\; F(\theta) \;+\; R(\theta).} \tag{14}$$

The Fisher term is positive semi-definite; the residual $R(\theta)$ carries the sign of the prediction errors and may be indefinite.

D.4. Regime Where FIM Dominates Hessian

The residual $R(\theta)$ is weighted by the per-sample errors $(f_i - y_i)$. Under *near-interpolation* — i.e., when the model fits the training labels well so that $\|f_i - y_i\|$ is small for most i — the residual contracts and $H(\theta) \approx F(\theta)$. Concretely, if $\|f_i - y_i\|_1 \leq \varepsilon_R$ and the logit Hessians $\|\nabla_\theta^2 z_k\|_2$ are bounded, then by Weyl's inequality,

$$\left|\lambda_{\max}(H) - \lambda_{\max}(F)\right| \;\leq\; \|R\|_2 \;=\; O(\varepsilon_R). \tag{15}$$

For directional quantities, the same bound applied to the Rayleigh quotient along $\Delta\theta$ gives

$$\left|\frac{\Delta\theta^\top H \Delta\theta}{\|\Delta\theta\|^2} \;-\; \frac{\Delta\theta^\top F \Delta\theta}{\|\Delta\theta\|^2}\right| \;\leq\; \|R\|_2. \tag{16}$$

Dividing both Rayleigh quotients by $g^\top \Delta\theta / \|\Delta\theta\|^2$ yields

$$\lambda_{\mathrm{KL}} \;\xrightarrow[\varepsilon_R \to 0]{}\; \lambda_{\mathrm{dir}}, \tag{17}$$

i.e., the Fisher-based estimator converges to the Hessian-based directional curvature as training approaches interpolation.

D.5. Empirical Spectrum and Why This Matters in Practice

A complementary observation justifies using F as a proxy for H even away from exact interpolation. The Fisher (and, by (13), the Gauss-Newton) spectrum of overparameterized networks is well known to be dominated by a small number of outlier eigenvalues, with the remaining bulk concentrated near zero [5, 6, 10–12, 15]. For classification, the number of large outliers scales with the number of classes $C \ll m$. Because λ_{dir} is a Rayleigh quotient along $\Delta\theta$, and $\Delta\theta$ — being either the gradient or an adaptive transform thereof — tends to align with the top eigenspace of F, the Fisher-side Rayleigh quotient $\Delta\theta^\top F \Delta\theta / (g^\top \Delta\theta)$ captures the same dominant curvature mode as $\Delta\theta^\top H \Delta\theta / (g^\top \Delta\theta)$ throughout training, not only at interpolation.

This is consistent with the empirical observation in Section 5 that λ_{KL} tracks the exact λ_{dir} cleanly from initialization through the Edge-of-Stability regime, and explains why a Fisher-based estimator is a viable drop-in for full-Hessian directional curvature in cross-entropy training.