FROM CURVATURE FIDELITY TO STEP RELIABILITY: WHAT GGN PROXIES CAPTURE AND MISS

Anonymous authors
Paper under double-blind review

ABSTRACT

Curvature-aware optimizers train neural networks by using a local loss model to guide the direction and size of parameter updates. They often replace the Hessian with the generalized Gauss–Newton (GGN) matrix, a positive-semidefinite surrogate that supports practical second-order computation. But when does a more faithful curvature model lead to a more reliable update, especially when the update is evaluated on a different batch? We propose a diagnostic framework linking curvature comparisons to predicted and realized loss changes. Within a common measured subspace, operator-pair geometry distinguishes differences in curvature directions from differences in normalized spectral shape. An exact error accounting then separates the effect of replacing the resolved GGN from the remaining Hessian contribution, finite-step nonlinearity, and evaluation-batch transfer. Automatic-differentiation products and dense reference checks support these measurements. Across independent ResNet and vision-transformer image-classification cohorts, transfer between batches dominates the measured absolute-error budget, largely through gradient differences. The GGN and resolved positive-Hessian models select identical updates from the original candidates; matching their predicted first-order decrease makes curvature differences affect selection. The primary loss comparison remains inconclusive, while secondary evaluation targets favor the GGN selector. These results explain why curvature fidelity, influence on step selection, and realized loss reduction must be assessed together, providing a basis for diagnosing which part of a local model matters for an update.

1 INTRODUCTION

Training a neural network requires choosing update directions and sizes from limited local information. Curvature-aware methods construct a local loss model to rescale gradients, suggest step sizes, or rank candidates. Hessian-free optimization and K-FAC use this information to shape updates (Martens, 2010; Martens & Grosse, 2015). What makes a curvature approximation informative for the loss reduction of the step actually taken?

The gradient predicts first-order loss change; the Hessian describes how that slope changes, completing a local quadratic model. The full Hessian is costly to form, and negative curvature can make an unconstrained model unbounded below. The generalized Gauss–Newton (GGN) matrix instead combines output-loss curvature with the network's first-order parameter sensitivity. For losses convex in the outputs, it is positive semidefinite, with nonnegative quadratic contributions in every direction. Matrix-free products support damped updates without storing the full matrix (Schraudolph, 2002), but how well does this model describe a finite step?

Figure 1 illustrates a concrete setting: an optimizer estimates the gradient and curvature on a small training batch, uses the resulting model to compare candidate updates, and evaluates those candidates on another set of training examples. We call the first set the *construction batch* and the second the *evaluation batch*. If predicted and observed loss changes disagree, several explanations are possible. The curvature proxy may be inaccurate along the update, the step may extend beyond the region where a quadratic model is useful, or the two batches may give different local loss landscapes. These explanations suggest different follow-up checks: refine the curvature model, examine a smaller step,

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.

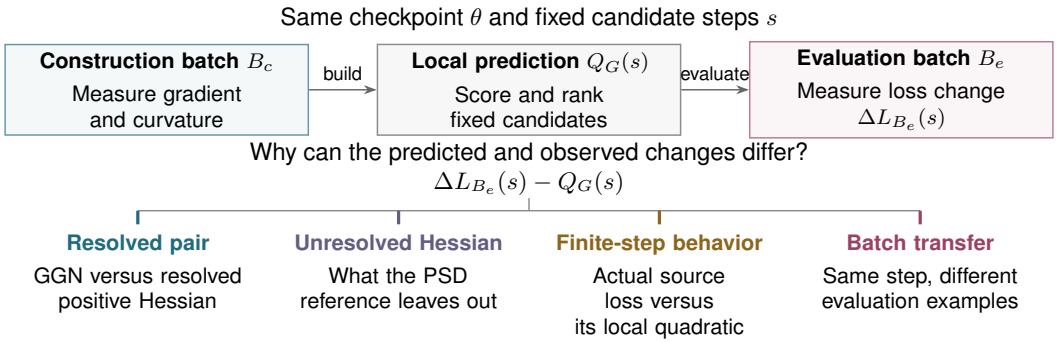


Figure 1: From a local prediction to an observed loss change. The same checkpoint and candidate steps are used throughout. Q_G denotes the construction-batch quadratic using the resolved GGN; ΔL_{B_e} is the loss change on separate evaluation examples. The lower row identifies the four contributions measured in Section 4.

or measure the discrepancy between batches. Distinguishing them is therefore part of assessing the value of curvature information. Work on debiasing mini-batch quadratics already establishes that using the same batch to select and evaluate directions can distort local predictions (Tatzel et al., 2025). The question here is how this batch effect interacts with curvature approximation and finite-step error, and when those errors matter for choosing an update.

Eigenvectors describe curvature directions and eigenvalues their strength, but a finite update uses only curvature along its own direction. Normalization also removes absolute scale, and construction-batch agreement does not account for a different evaluation gradient. Thus operator agreement must be connected to the prediction error of a specific step.

There is a further distinction between improving a prediction and changing a decision. Suppose two candidates already differ substantially in the loss decrease predicted by the gradient alone. Replacing the curvature term may change both scores without changing their ranking. When the first-order decreases are matched, however, curvature can become decisive. This motivates comparing the curvature correction with the margin between candidate scores, then measuring whether the resulting choice has lower loss on the intended target.

We propose a diagnostic framework connecting these matrix-, step-, and decision-level questions. We compare the GGN with the positive part of a Hessian projected into the same measured subspace, calling the components represented there *resolved*. Operator-pair geometry separates eigenbasis misalignment from normalized spectral-shape mismatch. An exact four-term accounting then attributes the discrepancy between predicted and observed loss changes to the resolved pair, the remaining signed Hessian contribution, finite-step nonquadratic behavior, and batch transfer. This identifies what replacing the resolved curvature changes and what it leaves unchanged. Automatic-differentiation products and dense reference checks support the measurements; a gradient-only scoring baseline and matched-first-order-decrease intervention test when the correction affects selection.

Across independent ResNet and vision-transformer image-classification cohorts, batch transfer dominates the measured absolute-error budget under the fixed equal-radius protocol, largely through source–target gradient differences. The GGN and resolved positive-Hessian models initially select identical actions, but matching the candidates' first-order decrease reveals different curvature-based choices. Their primary construction-batch regret comparison remains inconclusive, whereas secondary utility targets favor the GGN selector. These results connect operator comparison to the error components and decision margins that determine its relevance for an update.

2 Related Work

Curvature proxies and second-order optimization. Matrix-free Hessian and GGN products enable Hessian-free optimization (Pearlmutter, 1994; Schraudolph, 2002; Martens, 2010), while K-FAC uses structured Fisher approximations (Martens & Grosse, 2015). EK-FAC refines scaling

in that eigenbasis (George et al., 2018). Recent GN work studies structured exact solves (Korbit et al., 2025) and language-model iteration efficiency (Abreu et al., 2026). Under suitable likelihood parameterizations, the model Fisher and GGN coincide; the empirical Fisher is a distinct object in general (Kunstner et al., 2019). Natural gradient has an information-geometric motivation independent of Hessian approximation (Amari, 1998). Sophia combines inexpensive curvature estimates with clipping (Liu et al., 2024); Muon uses matrix orthogonalization (Liu et al., 2025). SOAP applies Adam in a Shampoo eigenbasis (Vyas et al., 2025); a KL-minimization perspective further motivates variants of both (Lin et al., 2026). Rather than comparing optimizer trajectories, we isolate quadratic scoring on fixed candidates, using the projected positive Hessian as a diagnostic reference.

Stochastic quadratics and step decisions. Tatzel et al. (2025) analyze bias from selecting and evaluating directions on the same mini-batch, using cross-batch eigenspace overlap and a second batch to debias directional estimates. Building on this batch perspective, we separate resolved-pair, unresolved-curvature, and nonquadratic errors and test their decision relevance. Song et al. (2025) show that alignment with leading Hessian eigenspaces need not preserve SGD's progress when updates are restricted to those spaces. Meterez et al. (2026) find that quadratic models can track finite training windows; we examine component-level error on transferred steps.

Hessian studies characterize bulk–outlier structure, negative curvature, and training stability (Sagun et al., 2018; Ghorbani et al., 2019; Cohen et al., 2021). Critical sharpness examines loss behavior along an update, including comparisons between optimized and evaluated objectives (Kalra et al., 2026). Probabilistic line searches address uncertainty in stochastic step-size decisions (Mahsereci & Hennig, 2017). Our linear baseline and equal-linear intervention test when a curvature correction changes a decision.

Operator geometry and task-specific fidelity. The trace functional $\mathrm{Tr}(\rho_A^{\alpha}\rho_B^{\beta})$ is rooted in the theory of matrix concavity and quantum quasi-entropies (Lieb, 1973; Petz, 1986). Eigenvector-overlap matrices also appear in random-matrix analyses of spectral estimation (Allez & Bouchaud, 2014; Bun et al., 2017). We use these established objects to separate normalized spectral shape from eigenbasis coupling, then restore scale and direction for step-level analysis. Fidelity's value depends on its use: Hong et al. (2025) study how curvature approximations affect influence-function data attribution, whereas McKay et al. (2026) analyze GN through function-space projections and error whitening. We measure utility by the selected action's loss, rather than infer it from operator proximity.

3 Operator-Pair Geometry in a Common Subspace

We first isolate the curvature comparison in Figure 1. Here an operator is a curvature matrix, accessed through its action on vectors. The analysis has three steps: compare both operators in the same measured coordinates, distinguish their relative curvature strengths from their directional alignment, and restore the scale and update direction needed to compare loss predictions. Section 4 then adds the remaining sources of error.

A common resolved reference. Let $L_c(\theta)$ be the empirical loss on a construction batch B_c, with gradient g_c, Hessian H_c, and GGN $G_c \succeq 0$. For a network with p parameters, repeated matrix–vector products generate Krylov directions on which curvature can be measured without forming the full matrices. At depth m, let $V_m \in \mathbb{R}^{p\times r_m}$ be an orthonormal basis of their common measured subspace S_m. Shared coordinates ensure that measured differences reflect the operators rather than different measurement subspaces. Here m is the requested depth *per operator*, whereas r_m is the common-basis rank; Section 5 specifies the construction. With $\mathrm{sym}(M) = (M + M^{\top})/2$, define

$$\widetilde{H}_{m,c} = \mathrm{sym}(V_m^{\top} H_c V_m), \qquad \widetilde{G}_{m,c} = \mathrm{sym}(V_m^{\top} G_c V_m). \tag{1}$$

For the full-m track, we compare

$$G_{m,\mathrm{full},c} = V_m \widetilde{G}_{m,c} V_m^{\top}, \qquad H_{m,\mathrm{full},+,c} = V_m [\widetilde{H}_{m,c}]_+ V_m^{\top}, \tag{2}$$

where $[\cdot]_+$ replaces negative eigenvalues of the projected symmetric matrix by zero. At the largest registered depth $m_\star$, a second track independently retains the top k eigenmodes of $\widetilde{G}_{m_\star,c}$ and

$[\widetilde{H}_{m_\star,c}]_+$; we denote their lifts by $G_{m_\star,k,c}$ and $H_{m_\star,k,+,c}$. Thus k is an operator-specific spectral truncation inside a fixed common coordinate system, whose common-subspace rank remains $r_{m_\star}$. Varying m tests how the comparison changes as more directions are measured; varying k tests how much it depends on the retained spectral modes within those coordinates.

The resolved positive Hessian is deliberately defined as the positive part *after* projection. In general,

$$V_m[\widetilde{H}_{m,c}]_+V_m^\top \neq P_m H_{+,c} P_m, \qquad P_m = V_m V_m^\top. \tag{3}$$

Positive-part extraction retains the locally convex modes of the measured Hessian, putting the reference and GGN in the same PSD class. The remaining signed Hessian contribution is accounted for separately in Section 4. Equation (3) specifies which reference is being tested: the empirical comparison uses the projected-then-positive operator. When the distinction between the two registered tracks is immaterial, (m, k) denotes either a full-m configuration with $k = \text{full}$ or an endpoint-k configuration.

Separating spectral shape from alignment. Two quadratic models can have the same curvature strengths but assign them to different directions, or share directions with different relative strengths. To distinguish these cases, we first remove overall scale. For nonzero PSD $A, B \in \mathbb{R}^{d\times d}$ in a common coordinate space, define $\rho_A = A/\operatorname{Tr}(A)$ and $\rho_B = B/\operatorname{Tr}(B)$. Write their eigenvalues in decreasing order as a_i, b_i, their eigenvectors as u_i, v_i, and $K_{ij} = |u_i^\top v_j|^2$. For $\alpha, \beta > 0$ and positive overlap trace, let

$$Z_{\alpha,\beta}(A, B) = \operatorname{Tr}(\rho_A^\alpha \rho_B^\beta) = \sum_{i,j} a_i^\alpha b_j^\beta K_{ij}, \tag{4}$$

$$Z^\star_{\alpha,\beta}(A, B) = \sum_i a_i^\alpha b_i^\beta. \tag{5}$$

Trace normalization makes each set of eigenvalues sum to one. The overlap weights K_{ij} record which curvature directions coincide. The aligned reference $Z^\star$ keeps the two spectra fixed and pairs their eigenmodes in decreasing order. Comparing Z with this reference therefore isolates the effect of their alignment. Define the normalized coupling and its signed self-gap by

$$C_{\alpha,\beta}(A, B) = \log \frac{d\, Z_{\alpha,\beta}(A, B)}{(\sum_i a_i^\alpha)(\sum_j b_j^\beta)}, \tag{6}$$

$$D_{\alpha,\beta}(A, B) = C_{\alpha,\beta}(A, A) - C_{\alpha,\beta}(A, B). \tag{7}$$

Adding and subtracting the aligned coupling gives

$$D_{\alpha,\beta} = \underbrace{\log \frac{Z^\star_{\alpha,\beta}}{Z_{\alpha,\beta}}}_{D^{\mathrm{geo}}_{\alpha,\beta}} + \underbrace{\log \frac{(\sum_i a_i^{\alpha+\beta})(\sum_i b_i^\beta)}{(\sum_i a_i^\beta) Z^\star_{\alpha,\beta}}}_{D^{\mathrm{spec}}_{\alpha,\beta}}. \tag{8}$$

The geometric term is nonnegative by the trace rearrangement inequality; the spectral term is signed. This decomposition distinguishes an alignment discrepancy from a difference in normalized spectral shape, even when a single aggregate score would combine them. We use $(A, B) = (G_{m,k,c}, H_{m,k,+,c})$ and $\alpha = \beta = 1/2$. Zero-mass modes remain in the common coordinate system, and all scores are normalized by the retained positive traces. Consequently, the estimand of these scores is the registered resolved pair. Estimating full-space geometry additionally requires tail information.

Connecting geometry to a tested direction. We now return to a candidate update: its quadratic prediction depends on absolute curvature along that step. Both pair scores are invariant under independent positive rescaling of A and B. Let $a = \operatorname{Tr}(A)$ and $b = \operatorname{Tr}(B)$. For a fixed step s, the resolved directional error is

$$E_{\mathrm{pair}}(s) = \tfrac{1}{2} s^\top (B - A) s \tag{9}$$

$$= \tfrac{b}{2} s^\top (\rho_B - \rho_A) s + \tfrac{b-a}{2} s^\top \rho_A s. \tag{10}$$

Thus normalized shape, absolute scale, and directional exposure are distinct inputs. In particular, $A = 2B \neq 0$ gives zero geometric and spectral deficits but generally nonzero $E_{\mathrm{pair}}(s)$. Operator-

pair geometry describes the resolved normalized pair, while this directional calculation asks how much their quadratic predictions differ for the actual update. This distinction supplies the resolved pair term in the loss-error accounting below.

Proofs, ordering qualifications, and the treatment of degenerate blocks are collected in Appendix A.

4 A Measurable Four-Term Reliability Accounting

The preceding section determines how replacing the resolved curvature changes a prediction along a step. To reach the observed outcome in Figure 1, we hold the checkpoint and candidate step s fixed and successively replace the resolved GGN by the resolved positive Hessian, restore the full Hessian quadratic, measure the actual construction-batch loss, and switch to the evaluation batch. The difference at each transition gives one error term. Let B_c and B_e denote the construction and evaluation batches, respectively.

For a batch B_b, write $\Delta L_{B_b}(s) = L_{B_b}(\theta + s) - L_{B_b}(\theta)$. For any registered (m, k) configuration, the resolved GGN quadratic constructed on B_c is

$$Q_{G_{m,k,c},B_c}(s) = g_c^\top s + \tfrac{1}{2} s^\top G_{m,k,c} s. \tag{11}$$

To separate operator replacement, unresolved curvature, finite-step behavior, and the evaluation target, define

$$E_{\mathrm{pair}}(s) = \tfrac{1}{2} s^\top (H_{m,k,+,c} - G_{m,k,c}) s, \tag{12}$$

$$E_{\mathrm{unresolved,signed}}(s) = \tfrac{1}{2} s^\top (H_c - H_{m,k,+,c}) s, \tag{13}$$

$$R_{3,c}(s) = \Delta L_{B_c}(s) - g_c^\top s - \tfrac{1}{2} s^\top H_c s, \tag{14}$$

$$E_{\mathrm{batch}}(s) = \Delta L_{B_e}(s) - \Delta L_{B_c}(s). \tag{15}$$

Adding the definitions gives the exact finite-step identity

$$\boxed{\Delta L_{B_e}(s) - Q_{G_{m,k,c},B_c}(s) = E_{\mathrm{pair}}(s) + E_{\mathrm{unresolved,signed}}(s) + R_{3,c}(s) + E_{\mathrm{batch}}(s).} \tag{16}$$

The first term measures the change from replacing the resolved GGN by the resolved positive Hessian along s. The signed unresolved term restores the rest of the Hessian quadratic; $R_{3,c}$ then accounts for the difference between that quadratic and the construction-batch loss. Finally, E_{batch} measures how the same step behaves when its evaluation target changes. Thus a large error can be located in the operator comparison, the remaining local model, or target transfer. This is an ex post attribution, with signs retained because the terms can reinforce or cancel one another.

The resolved positive-Hessian reference

$$Q_{H_{m,k,+,c},B_c}(s) = g_c^\top s + \tfrac{1}{2} s^\top H_{m,k,+,c} s \tag{17}$$

obeys the companion identity

$$\Delta L_{B_e}(s) - Q_{H_{m,k,+,c},B_c}(s) = E_{\mathrm{unresolved,signed}}(s) + R_{3,c}(s) + E_{\mathrm{batch}}(s). \tag{18}$$

It removes exactly the resolved pair term while leaving the other terms unchanged. This controlled replacement lets us ask whether improving the resolved comparison changes a prediction or a selected action. The signed unresolved term is measured directly with Hessian actions, without requiring separate estimates of the full positive and negative spectral parts. Appendix A gives that finer spectral decomposition and the conditions for interpreting a positive gap as a nonnegative tail.

Interpreting the finite-step residual. The notation $R_{3,c}$ records what remains after the Hessian quadratic. An $O(\|s\|^3)$ interpretation requires suitable local smoothness. ReLU activation-boundary crossings can violate a uniform smooth-region reading, so the empirical term is called the finite-step nonquadratic residual. Fixed-mask controls distinguish smooth within-cell behavior from realized-path switching, helping interpret why a local quadratic can change in accuracy as the radius grows.

Separating gradient transfer from higher-order effects. For a fixed direction p and $s = tp$, twice differentiable source and target losses satisfy

$$E_{\text{batch}}(tp) = t(g_e - g_c)^\top p + \tfrac{t^2}{2} p^\top (H_e - H_c)p + o(t^2). \tag{19}$$

When the directional gradient discrepancy is nonzero, it is the leading transfer term as the radius tends to zero under this expansion. Measuring it tests whether an observed transfer error arises mainly before curvature enters the prediction. We measure the target gradient on the frozen evaluation batch and define

$$E^{(1)}_{\text{batch}}(s) = (g_e - g_c)^\top s, \qquad E^{(\geq 2)}_{\text{batch}}(s) = E_{\text{batch}}(s) - E^{(1)}_{\text{batch}}(s). \tag{20}$$

The resulting measurable refinement is

$$\Delta L_{B_e} - Q_G = E_{\text{pair}} + E_{\text{unresolved,signed}} + R_{3,c} + E^{(1)}_{\text{batch}} + E^{(\geq 2)}_{\text{batch}}. \tag{21}$$

This measured five-term form refines batch transfer, rather than splitting the positive and negative Hessian spectra. It quantifies how much of the fixed-batch discrepancy is explained by the gradient and what remains after that contribution is removed.

5 Measurement and Protocol

The protocol follows the same progression as the analysis: validate the operator actions, construct a shared comparison, and evaluate fixed candidate steps on separate targets. Holding the candidates fixed isolates the effect of changing the quadratic score from changing the proposed directions themselves.

Operator construction and numerical checks. ViViT exploits low-rank GGN structure for scalable directional quantities (Dangel et al., 2022), while linear-operator interfaces provide a unified way to access curvature without materializing matrices (Dangel et al., 2025). We use this matrix-free viewpoint to measure both operators on a shared subspace. Hessian and GGN matrix–vector products are computed by automatic differentiation. Small dense models provide operator-level truth for the actions, projected matrices, eigenvalues, and quadratic forms. Separate convolutional and attention backends are checked before any trained checkpoint is interpreted.

Separate depth-192 Hessian and GGN Krylov chains start from $g_c/\|g_c\|$. Their interleaved, reorthogonalized common prefix V_m has dimension $r_m \leq 2m - 1$, where m is per-operator depth. Fresh actions recompute both projected operators; Appendix B specifies ordering, rejection tolerances, and orthogonalization. The full-m track uses $m \in \{64, 96, 128, 160, 192\}$; the endpoint-$k$ track fixes $m = 192$ and independently truncates the two PSD coordinate operators at $k \in \{16, 32, 64, 96, 128\}$.

Residuals, orthogonality, symmetry, linearity, repeatability, invariance leakage, and microbatch replay validate the actions. Dense comparisons check component accuracy separately from closure, which could otherwise hide cancellation between inaccurate terms.

Directions and finite-action decisions. The direction bases are separate from the common pair basis. If U_G is the endpoint GGN Krylov basis and $T_G = \text{sym}(U_G^\top G_c U_G)$, the unnormalized restricted step is

$$p_G = -U_G(T_G + \gamma I)^{-1} U_G^\top g_c, \qquad d_G = p_G/\|p_G\|, \qquad \gamma = 0.01. \tag{22}$$

We also set $d_{\text{ng}} = -g_c/\|g_c\|$. A descriptive control replaces U_G, T_G by an independent Hessian Krylov basis U_H and $[\text{sym}(U_H^\top H_c U_H)]_+$. The damping defines these directions, while both evaluated quadratics retain the undamped loss model. Finite steps are $s = rd$ at radii $r \in \{0.001, 0.01, 0.05\}$ fixed in advance.

The external selector excludes the positive-Hessian control and uses exactly seven actions: no-op and the two operational directions d_{ng}, d_G at the three radii. Each Q_G or Q_{H+} selector minimizes its score; values within 10^{-12} of the minimum are tied and resolved by smaller radius, then no-op, negative gradient, and restricted GGN. The empirical oracle minimizes the primary external-batch loss change. An action is oracle-compatible when its change is within the largest registered dual-microbatch tolerance of that minimum, where each action's tolerance is

$$\max\{10^{-10},\, 10|\Delta L^{(32)} - \Delta L^{(64)}|\}. \tag{23}$$

The oracle-compatible fraction measures how often a selector chooses an action whose observed loss is indistinguishable from the best candidate under this tolerance. We label a rule reliable when the run-clustered 95% confidence-interval lower endpoint of that fraction exceeds 0.5. Regret is the selected action's loss change minus the minimum observed loss change in the same candidate set. These complementary quantities record both the frequency and the cost of a suboptimal choice.

The linear baseline is $Q_{\mathrm{linear}}(s) = g_c^\top s$. To assess decision sensitivity, we compare the range of E_{pair} over the candidates with the gap between the two lowest Q_G scores. Their ratio, using the score-tie tolerance as a denominator floor, is the reported pair-correction leverage. It measures the correction's size relative to the ordering it would need to change. In the equal-linear challenge, one action on each operational direction is scaled to the same construction-gradient decrease. Specifically, $\ell_\eta = \eta \min_d[-g_c^\top d\, 0.05]$ and $r_d = \ell_\eta/(-g_c^\top d)$ for $\eta \in \{0.25, 0.5, 0.75, 1\}$. The candidate set is no-op plus those two matched actions. This intervention removes the linear-score difference so that the quadratic terms determine their ordering. The registered primary regret comparison uses the construction batch; the disjoint utility comparisons are secondary.

Batch roles and statistical units. Construction, utility, and line-search batches are disjoint in every protocol. The new core cohorts use held-out training targets; the official test set is reserved for a separate generalization study. Independent training runs are the inferential units; checkpoints, batches, directions, radii, and m/k configurations are repeated measurements. We first average repeated measurements within each run and bootstrap the six run-level means (20,000 registered resamples). For an error vector $(x_1, \ldots, x_J)$, the reported absolute-error share is computed per row as

$$w_j = \frac{|x_j|}{\sum_{\ell=1}^{J} |x_\ell|}. \tag{24}$$

If the denominator is zero, all shares on that row are set to zero; no reported core row has a zero denominator. We average these row-level ratios within each run and then bootstrap the six run means. Thus the estimand is the mean of per-row shares, not a ratio of separately averaged magnitudes, and it does not allocate the signed net error when components cancel. For component dominance, the lower 95% bound must exceed 0.5 and at least five of six run shares must exceed 0.5. The CIFAR-100 setting changes both dataset and label task, while the vision-transformer setting uses independent, unpaired seeds. We therefore compare the pattern across settings, with each cohort retaining its own run-level uncertainty estimate.

Appendix B gives the implementation and numerical validation details.

6 Experiments

We organize the experiments around three questions: whether the geometric measurements are numerically reliable, which terms explain external loss-model error, and when the resolved pair correction changes a decision. The central cohorts are ResNet20 on CIFAR-10 and CIFAR-100, and a compact vision transformer on CIFAR-10. Each setting has six independent training runs with repeated checkpoint measurements. Component shares are evaluated on utility batches, while the original action selectors are evaluated on disjoint line-search batches. All confidence intervals below use the run-level aggregation in Section 5; full settings and cohort counts appear in Appendix C.

6.1 Establishing the measurement

Before interpreting geometry, we compare the matrix-free actions and decompositions with explicit dense matrices on small models. All five prespecified dense-accuracy gates pass for the operator checks, and the four-term accounting closes against dense and tiny-network truth. The attention backend is checked separately before the transformer cohort is analyzed. Appendix B reports the action errors and Ritz residuals alongside closure checks. On the larger same-batch cohort, the resolved pair term does not meet the strict-dominance criterion for the core directions. Fixed-mask ReLU controls further locate the smooth cubic interpretation of the residual within activation regions.

The matched finite-difference comparison illustrates why these checks matter. On the same archived checkpoint trajectories, the float32 finite-difference pipeline produces a final mean D^{geo} about twenty times the automatic-differentiation value. Thus the apparent eigenspace discrepancy is sensitive to the

Table 1: Cross-setting six-run synthesis. Entries are percentages with 95% run-cluster bootstrap intervals. E_{batch} is its absolute-error-budget share on utility targets; compatibility is measured on disjoint line-search targets. Q_{H+} has the same compatibility mean and selected action as Q_G in every setting. Their intervals also agree, except that the C10 ViT Q_{H+} upper endpoint is 77.778% rather than 76.389%. Reliability requires the lower endpoint to exceed 50%.

Setting	E_{batch} share	Q_G compatible	Same action
C10 ResNet	93.562% [92.143%, 94.930%]	40.278% [34.722%, 45.833%]	720/720
C100 ResNet	98.541% [98.264%, 98.831%]	66.667% [62.500%, 70.833%]	720/720
C10 ViT	93.719% [93.017%, 94.429%]	69.444% [61.111%, 76.389%]	720/720

Table 2: First-order attribution of target transfer on utility batches. The within-transfer share is $|E^{(1)}_{\mathrm{batch}}|/(|E^{(1)}_{\mathrm{batch}}| + |E^{(\geq 2)}_{\mathrm{batch}}|)$; intervals are run-clustered 95% CIs. The five-term column includes the other four error components. The largest post-removal component is $E^{(\geq 2)}_{\mathrm{batch}}$ in all three settings.

Setting	Within transfer	Five-term share	Post-removal share
C10 ResNet	94.292% [93.534%, 94.994%]	90.082%	47.280% [45.723%, 48.650%]
C100 ResNet	98.230% [97.849%, 98.594%]	96.967%	43.153% [41.418%, 44.474%]
C10 ViT	97.946% [97.781%, 98.100%]	92.091%	35.030% [32.278%, 38.127%]

measurement procedure, even with the checkpoints held fixed. The separate budget study preserves all 18/18 primary labels as the common subspace is expanded, but only 12/18 along the retained-rank track. The measured conclusions are therefore more sensitive to discarding spectral modes than to the tested increases in common-subspace depth.

Earlier seed-held-out studies do not meet registered gates for incremental direction or one-step utility prediction over direct operator baselines. They retain their own folds and inferential units. These results motivate evaluating pair geometry relative to the remaining error components.

6.2 Locating the external loss-model error

Table 1 compares the four-term accounting across the independent cohorts. Target transfer dominates the mean absolute-error budget in all three settings under the fixed equal-radius protocol. Replacing the resolved GGN quadratic with the resolved positive-Hessian quadratic removes the pair term, but leaves this transfer unchanged. Consequently, improving the resolved comparison alone addresses only a small share of the measured component magnitudes in these experiments.

Transfer dominance does not determine action quality: GGN meets the oracle-compatibility criterion on CIFAR-100 ResNet and CIFAR-10 ViT, but not CIFAR-10 ResNet. Prediction errors need not change the best candidate. The resolved positive-Hessian selector always agrees with GGN and thus has identical compatibility means.

To understand the source of transfer, we measure the gradient on each frozen evaluation target and apply Eq. (20). Table 2 shows that the first-order discrepancy accounts for most of the transfer budget and remains the largest contribution in the expanded five-term accounting. After it is removed, no remaining component passes the dominance rule. This directly connects the observed transfer to the linear part of the loss model: for the tested steps, changing the batch mainly changes the gradient contribution, rather than only changing the curvature prediction.

A paired sensitivity study increases the target batch from 64 to 512 examples while fixing construction operators and directions. The first-order share changes by 2.223 percentage points (95% CI [1.417, 3.000]), below the registered material-change threshold, supporting persistence of first-order dominance across these sizes.

Table 3: Decision sensitivity. Compatibility entries are six-run means; original settings have 720 tasks each. Equal-linear rows use epoch 0, full $m = 192$, and $\eta = .25$. Regret is $\mathrm{regret}_{H+} - \mathrm{regret}_{G}$ in units of 10^{-6}; construction is the primary comparison and utility is secondary.

Design	Linear	Q_G	Q_{H+}	Split	Regret [95% CI]
Original C10 ResNet	54.2%	40.3%	40.3%	0/720	0
Original C100 ResNet	66.7%	66.7%	66.7%	0/720	0
Original C10 ViT	69.4%	69.4%	69.4%	0/720	0
Equal-linear B_c	50.0%	100.0%	66.7%	5/6	0.790 [−0.373, 2.513]
Equal-linear utility	20.8%	79.2%	29.2%	5/6	99.656 [44.699, 153.748]

6.3 Testing when curvature changes the choice

In Table 3, linear scores reproduce every selection on CIFAR-100 ResNet and CIFAR-10 ViT. Curvature changes some CIFAR-10 ResNet choices, but GGN and resolved positive-Hessian scores still agree: the pair correction is smaller than the GGN decision margin.

Matching the two directions' construction-gradient decrease makes the quadratic selectors disagree in 5/6 runs in the registered primary stratum. Thus resolved curvature can change a decision when linear scores are controlled.

Whether the changed choice is better is a separate empirical question. The primary construction-batch regret difference has a confidence interval spanning zero, so it does not establish an advantage for either quadratic selector. Secondary utility comparisons favor GGN, with the paired regret interval retaining that direction across the registered linear budgets. Restoring the signed unresolved contribution to the resolved positive-Hessian model yields the full signed Hessian quadratic, which selects the same action as GGN throughout the challenge. This control links the observed selection difference to what the resolved PSD reference leaves out; it does not isolate positive-tail and negative-curvature effects.

7 Discussion

To diagnose a curvature proxy, first verify operator actions and the measured subspace, then compare directional pair corrections with candidate score margins. Corrections too small to change the ordering have little immediate decision leverage; changed selections require evaluation on the intended target.

The error terms guide follow-up tests: measure target gradients for large batch transfer, expand directional coverage for large unresolved terms, and sweep radii or control ReLU activation regions for large nonquadratic residuals. Online acceptance additionally requires estimating these quantities before committing to an update.

The six-run cohorts establish recurring patterns for finite actions on held-out training targets, not separate causal effects of data and architecture: CIFAR-100 changes dataset and task together, and architecture seeds are unpaired. Full-space positive-Hessian comparisons, separate spectral-tail attribution, and long-horizon optimization extend the present projected reference and require additional measurements.

8 Conclusion

We connected resolved GGN–Hessian geometry to directional loss-model error and finite-action decisions. Geometry separates alignment from spectral shape; four-term accounting locates departures from the proxy prediction. Across the studied cohorts, batch transfer dominates external error, mostly through gradient differences. Matching linear decreases makes curvature affect selection, but primary regret remains inconclusive and secondary utility favors GGN. Curvature fidelity therefore needs to be assessed together with directional exposure, decision margins, and the evaluation target.

Reproducibility Statement

The manuscript distinguishes independent training runs from repeated checkpoint measurements and separates construction, utility, and line-search batches. Operator conventions, numerical checks, cohort-specific protocols, and the generation of reported tables and figures are documented in the appendices and accompanying code.

AI Use Statement

Generative AI tools assisted with the conceptual and mathematical framing, experimental design, code development, result analysis and interpretation, and drafting, translation, and editing of the manuscript. The authors reviewed the AI-assisted work and take responsibility for the final claims, code, citations, and text.

References

Natalie Abreu, Nikhil Vyas, Sham Kakade, and Depen Morwani. The potential of second-order optimization for LLMs: A study with full Gauss–Newton. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c8e9a2beb84ab1a616edb89581c4b32a-Abstract-Conference.html.

Romain Allez and Jean-Philippe Bouchaud. Eigenvector dynamics under free addition. *Random Matrices: Theory and Applications*, 3(3):1450010, 2014.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Joël Bun, Jean-Philippe Bouchaud, and Marc Potters. Cleaning large correlation matrices: Tools from Random Matrix Theory. *Physics Reports*, 666:1–109, 2017. doi: 10.1016/j.physrep.2016.10.005.

Jeremy M. 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.

Felix Dangel, Lukas Tatzel, and Philipp Hennig. ViViT: Curvature access through the generalized Gauss–Newton's low-rank structure. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=DzJ7JfPXkE.

Felix Dangel, Runa Eschenhagen, Weronika Ormaniec, Andres Fernandez, Lukas Tatzel, and Agustinus Kristiadi. Position: Curvature matrices should be democratized via linear operators. *arXiv preprint arXiv:2501.19183*, 2025. URL https://arxiv.org/abs/2501.19183.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast approximate natural gradient descent in a Kronecker factored eigenbasis. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.neurips.cc/paper_files/paper/2018/hash/48000647b315f6f00f913caa757a70b3-Abstract.html.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via Hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241, 2019.

Steve Hong, Runa Eschenhagen, Bruno Mlodozeniec, and Richard E. Turner. Better Hessians matter: Studying the impact of curvature approximations in influence functions. *arXiv preprint arXiv:2509.23437*, 2025. URL https://arxiv.org/abs/2509.23437.

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. *arXiv preprint arXiv:2601.16979*, 2026.

Mikalai Korbit, Adeyemi D. Adeoye, Alberto Bemporad, and Mario Zanon. Exact Gauss–Newton optimization for training deep neural networks. *Neurocomputing*, 658:131738, 2025. doi: 10.1016/j.neucom.2025.131738. URL https://arxiv.org/abs/2405.14402.

Frederik Kunstner, Lukas Balles, and Philipp Hennig. Limitations of the empirical Fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Elliott H. Lieb. Convex trace functions and the Wigner-Yanase-Dyson conjecture. *Advances in Mathematics*, 11(3):267–288, 1973.

Wu Lin, Scott C. Lowe, Felix Dangel, Runa Eschenhagen, Zikun Xu, and Roger B. Grosse. Understanding and improving Shampoo and SOAP via Kullback–Leibler minimization. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.03378v10. Extended version: arXiv:2509.03378v10.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *International Conference on Learning Representations*, pp. 1621–1650, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/06960915ba8674c7a898ec0b472b80ff-Abstract-Conference.html.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training, 2025. URL https://arxiv.org/abs/2502.16982. arXiv:2502.16982.

Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. *Journal of Machine Learning Research*, 18(119):1–59, 2017. URL https://jmlr.org/papers/v18/17-049.html.

James Martens. Deep learning via Hessian-free optimization. In *International Conference on Machine Learning*, pp. 735–742, 2010.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pp. 2408–2417, 2015.

Maricela Best McKay, Nathan P. Lawrence, Brian Wetton, and R. Bhushan Gopaluni. Error whitening: Why Gauss–Newton outperforms Newton. *arXiv preprint arXiv:2605.11316*, 2026. URL https://arxiv.org/abs/2605.11316.

Alexandru Meterez, Pranav Ajit Nair, Depen Morwani, Cengiz Pehlevan, Sham Kakade, and Alex Damian. A defense of the quadratic model. *arXiv preprint arXiv:2607.21716*, 2026. URL https://arxiv.org/abs/2607.21716.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994.

Dénes Petz. Quasi-entropies for finite quantum systems. *Reports on Mathematical Physics*, 23(1):57–65, 1986.

Levent Sagun, Utku Evci, V. Uğur Güney, Yann Dauphin, and Léon Bottou. Empirical analysis of the Hessian of over-parametrized neural networks. In *International Conference on Learning Representations (Workshop Track)*, 2018. URL https://openreview.net/forum?id=lfsOJhlH2P2.

Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002.

Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does SGD really happen in tiny subspaces? In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1757af1fe1429801bdf3abf5600f8bba-Abstract-Conference.html.

Lukas Tatzel, Bálint Mucsányi, Osane Hackel, and Philipp Hennig. Debiasing mini-batch quadratics for applications in deep learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Q0TEVKV2cp.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e988664070e9591f93fdcf605f7dc623-Abstract-Conference.html.

A Theory Details

Let $a_1 \geq \cdots \geq a_d \geq 0$ and $b_1 \geq \cdots \geq b_d \geq 0$ be the eigenvalues of the trace-normalized pair. Direct cancellation in the coupling definition gives

$$D_{\alpha,\beta} = \log \frac{(\sum_i a_i^{\alpha+\beta})(\sum_j b_j^{\beta})}{(\sum_i a_i^{\beta}) Z_{\alpha,\beta}}. \tag{25}$$

Multiplying and dividing by $Z^{\star}_{\alpha,\beta} = \sum_i a_i^{\alpha} b_i^{\beta}$ yields exactly $D_{\alpha,\beta} = D^{\mathrm{geo}}_{\alpha,\beta} + D^{\mathrm{spec}}_{\alpha,\beta}$. Von Neumann's trace inequality applied to ρ_A^{α} and ρ_B^{β} gives

$$Z_{\alpha,\beta} = \mathrm{Tr}(\rho_A^{\alpha}\rho_B^{\beta}) \leq \sum_i a_i^{\alpha} b_i^{\beta} = Z^{\star}_{\alpha,\beta}, \tag{26}$$

so $D^{\mathrm{geo}}_{\alpha,\beta} \geq 0$ whenever the observed overlap trace is positive. The spectral term D^{spec} is signed.

The score is invariant to the choice of eigenvectors within a repeated- eigenvalue block. Rotations inside a degenerate eigenspace leave the matrix powers and the trace Z invariant; in the overlap-sum representation their weights are constant within the corresponding block. Zero eigenvalues contribute zero mass. The implementation retains the full common coordinate dimension and rejects a pair only when either positive trace vanishes.

Equation (3) records a structural distinction. For $H = \mathrm{diag}(1, -1)$ and $V = 2^{-1/2}(1, 1)^{\top}$, one has $V^{\top} H V = 0$, and hence $V[V^{\top} H V]_+ V^{\top} = 0$. In contrast, $P H_+ P \neq 0$ for $P = VV^{\top}$. The experimental estimand is therefore the positive part of the projected Hessian; the projection of the full-space positive Hessian is a different operator.

Independent positive rescaling leaves ρ_A and ρ_B unchanged. Taking $A = 2B \neq 0$ therefore makes both normalized pair deficits vanish, whereas

$$\tfrac{1}{2} s^{\top}(B - A)s = -\tfrac{1}{2} s^{\top} B s \tag{27}$$

is nonzero whenever s has nonzero exposure to B. Thus normalized pair scores identify normalized shape, while absolute resolved quadratic error additionally requires scale and directional exposure.

More generally, writing $A = a\rho_A$ and $B = b\rho_B$ gives

$$\tfrac{1}{2} s^{\top}(B - A)s = \tfrac{b}{2} s^{\top}(\rho_B - \rho_A)s + \tfrac{b-a}{2} s^{\top}\rho_A s. \tag{28}$$

The first term still depends on the tested direction s, while the second depends on trace scale. A scalar normalized geometry score contains neither piece of directional information.

Adding the definitions in Section 4 cancels the inserted resolved positive-Hessian quadratic:

$$E_{\mathrm{pair}} + E_{\mathrm{unresolved,signed}} = \tfrac{1}{2} s^{\top}(H_c - G_{m,k,c})s, \tag{29}$$

$$E_{\mathrm{pair}} + E_{\mathrm{unresolved,signed}} + R_{3,c} = \Delta L_{B_c}(s) - Q_{G_{m,k,c},B_c}(s). \tag{30}$$

Adding E_{batch} gives Eq. (16). Dropping E_{pair} and replacing Q_G by Q_{H+} gives Eq. (18). These are algebraic identities for every tested s, independently of any Taylor remainder bound.

If exact full-space spectral parts $H_{+,c}$ and $H_{-,c}$ are available, with $H_{-,c} \succeq 0$ and $H_c = H_{+,c} - H_{-,c}$, then

$$E_{\mathrm{unresolved,signed}} = \underbrace{\tfrac{1}{2} s^{\top}(H_{+,c} - H_{m,k,+,c})s}_{E_{+,\mathrm{gap}}} \underbrace{-\tfrac{1}{2} s^{\top} H_{-,c} s}_{E_- \leq 0}. \tag{31}$$

Because projection and positive-part extraction do not commute, $E_{+,\mathrm{gap}}$ is signed for the projected-then-positive reference. A nonnegative tail follows under the stronger condition of a full-space spectral truncation $H_{+,k} \preceq H_+$. Substitution gives

$$\Delta L_{B_e} - Q_G = E_{\mathrm{pair}} + E_{+,\mathrm{gap}} + E_- + R_{3,c} + E_{\mathrm{batch}}. \tag{32}$$

The empirical large-model tables retain the four-term form because only the signed sum of the two unresolved spectral contributions is measured by direct Hessian action. In particular, a small signed sum may reflect genuinely small unresolved curvature or cancellation between the two gaps. Distinguishing these cases requires separate spectral certificates.

B Measurement and Reproducibility Details

All Hessian and GGN actions are evaluated at a fixed parameter vector, data ordering, reduction convention, and model mode. The reported quadratic omits the algorithmic damping penalty: damping defines the candidate direction, while the evaluated loss model remains undamped. All curvature and finite-loss measurements use evaluation mode. The ResNet20 has no normalization layer; the compact transformer uses LayerNorm and no dropout. Thus neither BatchNorm state updates nor stochastic dropout enter the reported actions.

The negative-gradient, restricted-GGN, and restricted-positive-Hessian control directions are rebuilt from the construction batch. Each restricted direction uses its own gradient-started Krylov chain, its endpoint projected system, and damping $\gamma = 0.01$, as in Eq. (22). The resulting step is normalized before a registered radius is applied. The reconstruction is confined to the projected system and uses neither a full-space orthogonal-complement inverse nor a $1/\lambda$ correction.

The Hessian and GGN direction chains share their first normalized-gradient vector but are otherwise built independently. To construct V_m, we add the common source once and then alternate the Hessian and GGN candidate at each depth. Every candidate receives two orthogonalization passes against the accepted basis and is rejected if its residual ratio is at most 10^{-12}. Therefore the recorded `power_per_operator` m has a theoretical common-rank ceiling $2m-1$; the realized `actual_common_rank` is the matrix dimension used downstream. Fresh exact actions on every accepted basis vector produce the symmetric coordinate matrices in Eq. (1).

The full-m configurations use all coordinate modes at paired depths $64, 96, 128, 160, 192$. The endpoint-k configurations keep the endpoint basis fixed and apply a PSD top-k eigentruncation separately to $\widetilde{G}_{192,c}$ and $[\widetilde{H}_{192,c}]_+$. Accordingly, the two top-k operators may retain different eigenvectors. Pair geometry is computed in the full common coordinate dimension, with omitted modes carrying zero spectral mass. The full projected GGN is used in the full-m quadratic; a qualification gate records its tiny negative numerical ratio while retaining the original operator.

For each checkpoint, external target batch, and (m, k) configuration, the operational action set contains no-op plus negative-gradient and restricted-GGN directions at three radii. The positive-Hessian direction is only a descriptive stability control. The two score models evaluate the same seven actions. Score ties, dual-microbatch actual tolerances, oracle compatibility, and selector ordering are fixed as stated in Section 5; the candidate set is specified before outcome inspection. The comparison covers a finite local action set; evaluating a continuous optimizer requires a separate protocol.

The qualification chain contains dense action checks, repeatability, linearity, bilinear symmetry, GGN PSD checks, weighted microbatch replay, explicit residuals, reorthogonalization diagnostics, and multiple subspace and retained-rank budgets. Exact AD removes finite-difference truncation; the remaining numerical sources are floating-point, projection, and Krylov truncation error. Five of five dense-accuracy gates pass. The tiny-network Hessian/GGN action errors are 0 and 1.465×10^{-16}; the tiny-transformer dense-action error and maximum Ritz residual are 6.929×10^{-15} and 8.440×10^{-14}. Full-m labels are stable in 18/18 primary strata, while endpoint-k labels are stable in 12/18. The archived finite-difference/exact-AD ratio of mean final D^{geo} is 20.562.

The six training runs are the independent units; the 18 checkpoints and decomposition rows are repeated measurements. A global metric is first averaged over all registered repetitions within each run; the six means are then resampled with replacement 20,000 times. The action label is `reliable` only when the lower endpoint of the resulting 95% interval exceeds 0.5. Component-dominance labels additionally require at least five of six run-level shares above 0.5. Each quantity is reported

under its respective registered rule. Absolute-error shares follow Eq. (24): ratios are formed per measurement row, averaged within run, and only then bootstrapped. This aggregation order is used for both the original four-term and measured five-term budgets.

C Cohorts, Inference, and Reporting Boundaries

The earlier checkpoint studies and new independent cohorts differ in runs, schedules, directions, ranks, and target roles. Each protocol retains its own inferential sample.

The direction study has 36 records from three seeds, two learning rates, and six epochs; the one-step study has 36 valid checkpoint records under a separate fixed protocol. Checkpoints are repeated measurements, the original runs are inferential units, and both studies remain separate from the new cohorts. All three registered 10% incremental direction-prediction gates fail against direct operator baselines. The one-step prediction, mechanism, and overall gates also fail; the largest point improvement among the four primary one-step targets is 2.931%, below the 5% threshold.

Each central cohort has six independent runs and three registered checkpoints per run. Every checkpoint combines two directions, three radii, five full-m points, and five endpoint-k points. Four utility batches support component dominance; four disjoint line-search batches support the seven-action selector. This yields 720 selector tasks per cohort and 2,160 in total, with six runs per cohort as the inferential samples.

The C10 and C100 convolutional cohorts use ResNet20 without normalization; the classifier has 10 or 100 outputs. The transformer is CIFARViT-P4-D72-L4-H3 with three attention heads, LayerNorm, 261,730 trainable parameters, and no dropout. Every model is trained for 160 epochs with batch size 128, mean cross-entropy without label smoothing, SGD with momentum 0.9 and weight decay 5×10^{-4}, no Nesterov or clipping, five warmup epochs, and cosine decay to 0.01 of the base rate. Each cohort has three runs at base rate 0.05 and three at 0.10; measurements use epochs 0, 80, and 160. Training uses random crop and horizontal flip; all evaluation roles use deterministic normalization without augmentation.

Training run is the clustering unit; checkpoints, targets, directions, radii, and subspace configurations are repeated observations. Protocol heterogeneity across the three cohorts makes their synthesis descriptive.

The new cohorts use disjoint construction, utility, and line-search batches drawn from the training split. The official test set is reserved for a separate generalization study.

The Q_{H+} selector replaces the resolved GGN quadratic by the positive part of the projected Hessian, with linear term and action set fixed. Both selectors are unreliable on CIFAR-10 and reliable on CIFAR-100 and the vision transformer. Their choices coincide in every task, yielding zero action gain. The pure linear selector reveals that all C100 and ViT actions, and two thirds of C10 actions, are already fixed before adding curvature. The equal-linear challenge removes this linear-score difference on all 18 C10 checkpoints from six runs. It produces 3,600 target rows and 720 strata over four η values; the registered primary stratum uses epoch 0, full $m = 192$, and $\eta = 0.25$.

Exact construction and evaluation gradients are measured on the four 256-example utility batches and four disjoint 256-example line-search batches. Across the three cohorts this produces 54 checkpoint units and 2,592 transfer rows. The batch-size study uses two disjoint 512-example chains and their nested 64, 128, 256, and 512 prefixes, leaving the construction operator and directions fixed.

CIFAR-100 jointly changes input distribution, label space, and task; the vision-transformer cohort changes model family with new, unpaired seeds. Their synthesis describes pattern transfer. Dataset- and architecture-only causal effects require controlled, paired experiments.

Each cohort uses full-m points $m \in \{64, 96, 128, 160, 192\}$ and endpoint $k \in \{16, 32, 64, 96, 128\}$ in the fixed $m = 192$ coordinates. Maximum same-batch closure errors for CIFAR-10 ResNet, CIFAR-100 ResNet, and CIFAR-10 ViT are 2.78×10^{-17}, 5.42×10^{-18}, and 3.90×10^{-18}. The latter two external maxima are 2.78×10^{-17} and 1.39×10^{-17}; C10 ResNet uses the same exact construction and reports a 2.78×10^{-17} maximum.

Core measurements used Linux 5.15, Python 3.12.3, PyTorch 2.8.0+cu128, CUDA 12.8, cuDNN 91002, driver 580.142, one RTX 4080 SUPER, and eight CPU threads, with deterministic execution

and AMP/TF32 disabled. C10 ResNet, C100 ResNet, and C10 ViT took 5290.075, 5234.614, and 11581.570 seconds, with peak CUDA allocations of 0.223772, 0.223922, and 0.422493 GiB. The 6.14 sequential GPU-hours exclude training and preliminary implementation checks; capacity planning also depends on overhead and concurrency.