LOCAL TARGET CONTRAST UNDER CONDITIONAL UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Local contrastive losses for regression use target differences to choose which nearby representations to separate. These differences can contain irreducible noise. We analyze a quadratic margin loss by conditioning on the input batch, including the dependence introduced by a target-gap median. Its preferred pair distance is proportional to the conditional probability of a push label, which need not measure a difference in conditional means. When the loss acts directly on scalar predictions, we give a sufficient condition under which the Bayes predictor is not a local minimum of the combined population objective. An exact example has positive excess squared risk for every positive loss weight. For nonconstant means, a finite-design analysis identifies net pair forces; incompatible edge preferences can cancel. Intermediate geometry can instead use prediction-insensitive directions, but independent activation changes need not correspond to a shared encoder update. We test this distinction using separate fitting, protection, diagnostic and evaluation inputs. The tested shared updates show a tradeoff between geometry descent and protection of predictions on independent inputs. On the related red and white Wine Quality tasks, MLP and Transformer comparisons show small, mixed effects. The results separate a condition for objective bias from a general ranking of loss locations in training.

1 INTRODUCTION

Nearby inputs can have different observed targets even when their expected targets are equal. A contrastive regression loss may ask a model to separate these observations, while squared loss asks it to predict their conditional mean. When do these two objectives conflict?

Separation encouraged by a pair loss is insufficient to establish a prediction benefit. At the prediction layer, it constrains the values used to compute regression error. At an intermediate layer, the head can ignore some representation changes. Neither observation determines the trained model's risk: pair forces may cancel, and a shared encoder may not realize all changes that preserve individual predictions.

We ask when the pair objective admits a descent direction at the Bayes predictor. For a quadratic margin loss, conditioning on the input batch accounts for its random target-gap threshold. This gives a sufficient bias condition and, for nonconstant means on a finite design, a local net-force criterion. Parameter interventions examine the separate question of shared update feasibility. Real regression comparisons check the practical ranking of loss locations; they do not identify the source of target uncertainty.

Our contributions are threefold:

- We derive the preferred pair distance under conditional target uncertainty and a sufficient condition for bias when the loss acts on scalar predictions.
- We specialize pair-force equilibrium to regression around a nonconstant conditional mean, obtaining a local prediction-shift formula and its excess squared risk.
- We test the limits of inferring prediction benefits from geometry: shared updates trade geometry descent against output protection in synthetic models, while real regression gives no consistent loss-location ranking.

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.

2 Related Work

Regression and pair geometry. Squared loss targets the conditional mean (Shalev-Shwartz & Ben-David, 2014); mixture models address distributions or specialized predictors (Bishop, 1994; Jacobs et al., 1991). Our pair loss has the DrLIM form (Hadsell et al., 2006). That work also describes equilibrium between attractive and repulsive forces (Section 2.2, Figure 2). Here we condition the pair label on batch inputs and relate the resulting force to displacement from the conditional mean and excess squared risk. The loss and force interpretation are inherited; the regression analysis is the focus. RNC orders representations by target distance (Zha et al., 2023); its ordering theorem concerns feature similarities near the RNC loss infimum, rather than displacement of the conditional mean under our joint squared-loss objective. ConR aligns label and feature neighborhoods (Keramati et al., 2024). SCOREQ uses target-ordered triplets for speech-quality regression (Ragano et al., 2024). SRL imposes enveloping and homogeneity constraints for imbalanced regression (Dong et al., 2025). Our quadratic margin analysis does not cover these full objectives. The earlier MSE–RNC comparison is an adaptation (Appendix H).

GAR combines pointwise regression with pairwise label-difference losses, motivated by gradient magnitude and direction alignment (Zhu et al., 2025). Its stated scope is clean regression without noise or distribution shift. We study a different loss under conditional target uncertainty; the resulting bias conditions are not a claim about GAR.

Uncertain pair supervision. Robust contrastive learning studies noisy views (Chuang et al., 2022), weak class supervision (Cui et al., 2023), and noisy labels (Cui et al., 2025). ConFrag selects regression samples through contrastive fragments and neighborhood agreement (Kim et al., 2024). Our targets can be correctly recorded yet conditionally uncertain. UACQR separates conditional variability from estimation uncertainty for prediction intervals (Rossellini et al., 2024); here the endpoint is pointwise squared risk. We ask when target-gap geometry conflicts with conditional-mean prediction, accounting for its coupling to the predictor.

Output protection. OGD projects gradients to preserve outputs in continual learning (Farajtabar et al., 2020). We use this existing construction to test shared update feasibility. Its covariance identities and implementation details are in Appendix F; the projection is not a contribution of the present work.

3 Prediction and Local Pair Supervision

Let (X, Y) have a finite second target moment, and let $\mu(x) = \mathbb{E}[Y \mid X = x]$. For a fixed predictor f,

$$R(f) - R^* = \mathbb{E}[(f(X) - \mu(X))^2], \qquad R^* = \mathbb{E}[\mathrm{Var}(Y \mid X)]. \tag{1}$$

The right-hand side separates prediction error from residual label noise. The population identity does not imply that a training algorithm finds its minimum. When a generator is known, we estimate it on test inputs using its conditional mean, rather than subtracting two noisy empirical MSE values.

For each observation in a training batch of size B, the local pair rule is

$$j(i) = \arg\min_{j \neq i} \|x_i - x_j\|_2, \qquad d_i = |y_i - y_{j(i)}|, \qquad A_i = \mathbf{1}[d_i > \mathrm{median}(d_1, \ldots, d_B)]. \tag{2}$$

An input-only rule resolves neighbor ties. With distance D_i and margin r, the additional loss is

$$\mathcal{L}_{\mathrm{geo}} = \frac{1}{B} \sum_i \{(1 - A_i) D_i^2 + A_i (r - D_i)_+^2\}, \qquad \mathcal{L} = \mathcal{L}_{\mathrm{MSE}} + \lambda \mathcal{L}_{\mathrm{geo}}. \tag{3}$$

The target-gap threshold adapts to the batch. It is not an estimate of which target differences are recoverable from inputs. In particular, pure target noise can create push pairs even when μ is constant.

4 When Local Contrast Changes the Regression Target

We distinguish representation distance from prediction distance. The results concern population objectives on fresh batches, not training convergence.

Conditional pair risk. Select a directed pair (i, j) using batch inputs only, and let $\mathcal{X}$ contain those inputs and the selected indices. Write $A \in \{0, 1\}$ for the push label and $p_{\mathcal{X}} = \Pr(A = 1 \mid \mathcal{X})$. This definition allows $A = \mathbf{1}\{|Y_i - Y_j| > \tau_B\}$ with a label-dependent batch median τ_B; it does not treat that median as independent of the pair. For a deterministic encoder z, let $D = \|z(X_i) - z(X_j)\|$ and margin $r > 0$. The conditional expected loss is

$$Q_p(D) = (1 - p)D^2 + p(r - D)_+^2, \qquad p = p_{\mathcal{X}}. \tag{4}$$

On $0 \le D \le r$, this equals $(D - pr)^2 + p(1 - p)r^2$. Its unique global minimizer is $D = pr$ for $0 \le p < 1$; for $p = 1$, every $D \ge r$ minimizes it. Thus noisy targets can favor distance between equal conditional means. These pairwise optima need not be jointly realizable by one embedding.

Proposition 1: a sufficient condition for prediction bias. Assume $\mathbb{E}[Y \mid X] = 0$, $\mathbb{E}[Y^2] < \infty$, and apply the pair loss directly to the scalar prediction f, so $D = |f(X_i) - f(X_j)|$. Define

$$J_\lambda(f) = \mathbb{E}[(Y - f(X))^2] + \lambda\, \mathbb{E}[Q_{p_{\mathcal{X}}}(D)], \qquad \lambda > 0. \tag{5}$$

Suppose the predictor class contains $f_t = tv$ for all sufficiently small $t \ge 0$, where v is bounded and $C_v = \mathbb{E}[p_{\mathcal{X}}|v(X_i) - v(X_j)|] > 0$. Then the Bayes predictor $f = 0$ is not a local minimum of J_λ. Indeed, writing $\Delta v = v(X_i) - v(X_j)$, for sufficiently small $t > 0$,

$$J_\lambda(tv) - J_\lambda(0) = t^2\{\mathbb{E}[v(X)^2] + \lambda\mathbb{E}[(\Delta v)^2]\} - 2\lambda r t C_v < 0. \tag{6}$$

Every global minimizer, if one exists, has positive excess squared risk. The condition $C_v > 0$ must be checked; a median rule alone does not ensure it.

An exact example and its boundary. Let X be uniform on $\{-h, h\}$, $h > 0$, and let Y independently take values $-\sigma, +\sigma$ with equal probability. Use cross-input pairs with independent labels and $A = \mathbf{1}\{|Y_i - Y_j| > \sigma\}$, so $p = 1/2$. This fixed-threshold rule differs from the training batch median. For the scalar representation and prediction $f(X) = b + aX/h$, every minimizer satisfies

$$b^* = 0, \qquad |a^*| = \frac{2\lambda pr}{1 + 4\lambda}, \qquad R(f^*) - R^* = \left(\frac{2\lambda pr}{1 + 4\lambda}\right)^2 > 0. \tag{7}$$

Now allow $z(X) = (0, prX/(2h))$ and a prediction head $H(z) = z_1$. The prediction is Bayes optimal and the pair distance is pr. Both objectives attain their separate minima: the head ignores the separated coordinate. Appendix A gives the derivations.

The proposition applies to prediction-level geometry. The synthetic token model regularizes a raw token, so it does not impose $D = |f(X_i) - f(X_j)|$. We test objective location and weight separately. Token separation alone establishes neither improved prediction nor harmful prediction bias.

Nonconstant means and shared prediction values. The preferred distance of each pair does not determine the bias of a shared predictor. To make this distinction quantitative, consider a finite input design with conditional means $\mu \in \mathbb{R}^m$, masses $\Pi = \mathrm{diag}(\pi_i) > 0$, and an oriented incidence matrix B. Edge weights w_e sum to one, and p_e is the conditional push probability given that edge, averaging over its batch contexts. Use $z_i = (f_i/c, 0)$ and the unit head $(c, \sqrt{1 - c^2})$, $0 < c \le 1$. Then prediction is f_i and pair distance is $|(Bf)_e|/c$. Changing c rotates the head while keeping a two-dimensional geometry and no input bypass. This family fixes the bottleneck explicitly.

Assume $(B\mu)_e \ne 0$ and $|(B\mu)_e| \ne cr$ for every used edge. Let $s_e = \mathrm{sign}(B\mu)_e$ and $a_e = \mathbf{1}[|(B\mu)_e| < cr]$. Define

$$L = B^\top \mathrm{diag}\{w_e(1 - p_e + p_e a_e)\}B, \quad b = B^\top(w \odot p \odot a \odot s), \quad g_c = c^{-2}L\mu - rc^{-1}b. \tag{8}$$

Proposition 2. In a region preserving these edge signs and hinge states, the stationary prediction and its excess risk are

$$\delta_\lambda = f_\lambda - \mu = -\lambda(\Pi + \lambda c^{-2}L)^{-1}g_c, \qquad R(f_\lambda) - R^* = \delta_\lambda^\top \Pi \delta_\lambda. \tag{9}$$

The formula applies only if f_λ remains in that region; it is not a global solution across changes of ordering. At μ, $g_c \ne 0$ gives a descent direction for every $\lambda > 0$, whereas $g_c = 0$ makes μ a strict local minimum. Thus *net force at each input* determines local bias. This specializes the force-equilibrium interpretation of Hadsell et al. (2006) to displacement from the conditional mean. For three equally weighted edges joining means $(-.2, 0, .2)$, take $p_e = .3$ and $c = r = 1$. Distances $.2, .2, .4$ all differ from $.3$, but their forces cancel and $g_c = 0$. Appendix B proves these claims and gives the spectral form of the risk.

Mean shifts and prediction-insensitive directions. The pair objective is unchanged by $f \mapsto f + t\mathbf{1}$. Any population local minimizer in a class admitting these shifts must therefore satisfy $\mathbb{E}[f(X) - \mu(X)] = 0$. A trained predictor can still have a mean offset; this is not the distortion required by the stationary pair objective.

For a fixed neural model and input batch, let J be the Jacobian of its predictions with respect to the token activations, and let $g_z = \nabla_z \mathcal{L}_{\text{geo}}$. With $P_\perp$ the orthogonal projection onto $\ker J$, the direction $-P_\perp g_z$ has zero first-order prediction change and geometry derivative $-\|P_\perp g_z\|^2$. We use this standard projection as an intervention diagnostic. It separates local geometry sensitivity from overall token use; it does not show that a single encoder parameter update realizes all projected activation changes.

From representation freedom to shared updates. A head-insensitive representation change need not be realizable by one encoder update across inputs. We test this separately using standard output-gradient projections. A finite protection sample constrains outputs only at its sampled inputs; independent inputs determine whether protection transfers. Appendix F gives the covariance criterion, first-order response estimate and projection construction. These checks concern local prediction changes, not regression-risk gains.

5 Models and Loss Placement

For the controlled synthetic experiments, a Transformer tokenizes each scalar input as $e_{ij} = x_{ij} w_j + v_j \in \mathbb{R}^{128}$. A shared MLP $t_i = \psi(x_i)$ with widths $16 \to 64 \to 128$ supplies one additional token. Three Transformer layers with four attention heads process $[\text{CLS}; e_{i1}; \ldots; e_{i16}; t_i]$. LayerNorm and a scalar linear head read the CLS output. The raw-token loss uses $D_i = \|t_i - t_{j(i)}\|_2$ before attention. The prediction-level control uses $D_i = |\widehat{y}_i - \widehat{y}_{j(i)}|$ with exactly the same predictor. Setting $\lambda = 0$ removes the geometry objective. An MLP with four hidden layers of width 128 provides a simpler regression reference.

This additional input token provides an explicit intermediate representation on which to place the loss. We call this the token-augmented Transformer; the code identifier ALDT refers to the same experimental model. Its role is to compare loss locations within one predictor. The real-data experiments use an MLP and a Transformer without this extra token.

6 Controlled Experiments

6.1 Nonconstant Means: Net Force and Coupling

We first isolate coupling with the finite-design model of Proposition 2. The unit head changes angle while representation dimension, metric and margin stay fixed, with no bypass. This controls a specific coupling family; it does not equate effective regularization strengths across neural models. Eight designs, two mean amplitudes, two noise scales, three coupling values and three weights give 288 conditions. Pair probabilities are estimated using the nearest-neighbor and batch-median rule with fresh noisy labels. An independent batch bank tests the calculated predictions on the same design. Full settings and solver diagnostics are in Appendix B.

The closed form stays inside its required sign/hinge region in 119 of 288 conditions. In those cases, the maximum RMS discrepancy from same-bank optimization is 2.13×10^{-9}; the median discrepancy against the independent-bank solution is 9.61×10^{-5}. The estimated descent direction lowers the independent-bank objective in all 288 conditions. Figure 1 also reports the 169 conditions outside the formula's scope. These finite-design checks test the local force calculation, not prediction on new inputs or global optimization.

6.2 Observations, Targets and Protocol

For the branch tasks, let $u \sim \text{Unif}(-2, 2)$ and $c \in \{-1, 0, 1\}$ have probabilities $(q/2, 1-q, q/2)$, independently. We use

$$Y = 1.2u + 0.5\sin(3u) + Gc + \epsilon_y, \qquad X = \phi_m(u) + \kappa c b_m + \epsilon_x, \tag{10}$$

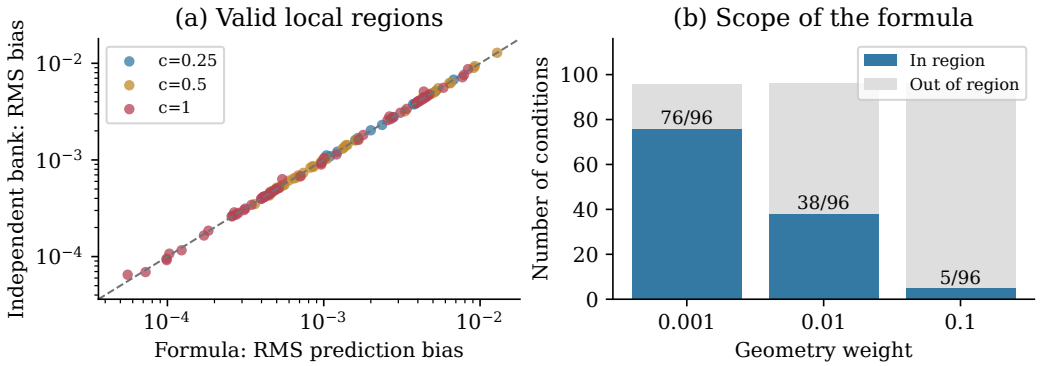

Figure 1: Nonconstant-mean finite-design probe. Left: local formula versus optimization using independently sampled pair statistics, for the 119 valid regions. Color denotes head coupling c. Right: all 288 conditions, including 169 where the local formula crosses an edge-order or hinge boundary.

with $q = 0.4$, $G = 3$, independent Gaussian observation and target noise of standard deviation 0.05, and a fixed map within each train/validation/test split triple. The exposed map contains $u, u^2, \sin(2u), \cos(3u)$ and twelve random linear combinations, with b_m the last coordinate vector. The hidden map uses sixteen random sinusoidal features and a distributed unit branch direction; it contains no explicit u coordinate. Appendix C gives the exact map and seeds. Both maps still have one continuous latent variable.

Positive visibility κ supplies partial branch information, not exact recovery in the presence of input noise. At $\kappa = 0$, the branch is absent but the base function remains predictable. A separate pure-noise control uses the same nuisance inputs at $\kappa = 0$ and independent Gaussian targets of variance $qG^2 + \sigma_y^2$, giving raw $\mu(x) = 0$. Under training normalization its conditional mean is $-\bar{y}_{\mathrm{tr}}/s_{y,\mathrm{tr}}$; Proposition 1 applies after translation. Only this latter control has its constant-mean premise.

The synthetic training grid contains 300 primary fits, 60 prediction-coupled branch controls and 30 pure-noise fits. It crosses two observation maps, $\kappa \in \{0, .08, .32\}$, training sizes 1,024 and 4,096, and five paired seeds. The primary grid compares a strong MLP with token models at $\lambda \in \{0, .005, .05, .5\}$. Checkpoints are selected by validation MSE within 120 epochs. Direct excess risk is the test average of $(\widehat{f} - \mu)^2$, using numerical branch means and exact pure-noise means. The appendix records the complete grid and seed protocol. Normalization uses training statistics only. The fixed loss-weight scan is a sensitivity study, not a claim that every model is optimally tuned. All planned cells are reported in the appendix.

6.3 Prediction Coupling under Pure Noise

With pure-noise targets and $\lambda = 0.5$, the paired changes in direct excess risk from raw-token to prediction geometry were exposed coordinates: 0.00201 (median 0.00046, seed range $[0.00037, 0.00665]$, exact $p = 0.0625$); hidden nonlinear: 0.00014 (median 0.00023, seed range $[-0.00093, 0.00129]$, exact $p = 0.8125$). The raw conditional mean is zero; evaluation accounts for the training-sample centering and scaling of targets.

All five exposed-map differences are positive; hidden-map differences have mixed signs. Thus the finite-network comparison does not establish a stable increase across both maps. Validation-selected networks need not minimize the population objective in Proposition 1.

The decomposition $(\bar{f} - \mu_0)^2 + N^{-1} \sum_i (f_i - \bar{f})^2$ helps interpret this result. At weight .5, mean-offset squared accounts for 76.3% of the average exposed-map change. The pair loss is invariant to constant shifts, so this offset cannot be attributed to its stationary pair forces. Nearest-input prediction distances also change with mixed signs (Table 3). Optimization and checkpoint selection therefore remain plausible explanations for part of the effect.

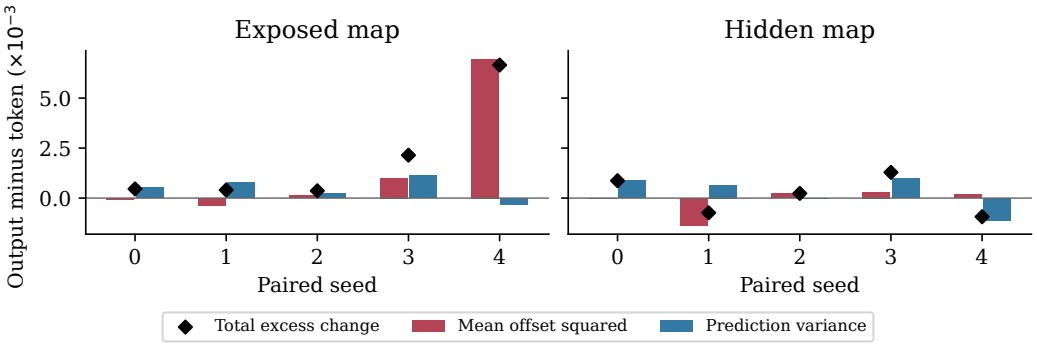

Figure 2: Pure-noise error decomposition. Each seed compares output geometry with raw-token geometry at weight .5. Bars separate the change in squared mean offset from the change in prediction variance; diamonds show their sum. This finite-test identity is not a decomposition over training samples.

6.4 Representation Changes at Fixed Parameters

We intervene on token activations from 40 checkpoints, using matched displacements at three scales and four batches per checkpoint. The projected negative gradient decreases geometry loss in all 480 tested batch–scale cases. Its mean log–log output-response slope is 1.98, versus .94 for aligned and 1.00 for random directions (Figure 8). Token permutation changes branch-task predictions by mean RMS .97–1.14, so the token is used despite these locally insensitive directions. The pure-noise response is smaller, .013–.015. These activation changes motivate the next test: can one shared parameter update preserve predictions on independent inputs? Full intervention and training comparisons are in Appendices B and D.

6.5 Shared Updates on Independent Inputs

We replace samplewise activation changes with updates of the shared 9,408-parameter encoder, holding the backbone fixed. Geometry fitting, protection, diagnostic and evaluation inputs are separate. At matched parameter step norms, ridge updates meet a joint geometry-descent and reduced-prediction-disturbance criterion in 29/32 exposed, 32/32 hidden and 28/32 four-latent models. Standard 128-input projection meets it in 32/32 models in each condition. The criterion requires both updates to descend, at least 25% of ordinary descent retained and at most 50% of ordinary prediction disturbance (Appendix F).

Finite protection does not imply population invariance: projected updates have approximately first-order responses on independent inputs. The following experiment varies protection coverage and allowed update coordinates, using these 96 models for development and 32 new models for confirmation.

6.6 Coverage and Available Geometry Descent

Protection can suppress output response while leaving too little geometry descent. We test a fixed prediction: at 512 update coordinates and 128 protection inputs, full coverage exceeds protection restricted to the leftmost eighth of a latent coordinate in joint success. The required point difference is at least .15, with a positive paired interval.

We fit rank, cell-identity, scale, directional and finite-step predictors using 96 development models, with seed-group cross-validation. Directional checks estimate geometry descent and output response on a separate diagnostic set; forward checks measure actual finite-step changes there. Appendix G specifies all features, thresholds and frozen prediction rules.

Frozen predictors are evaluated on 32 new models: 16 hidden-branch models at visibility .16 and 16 four-latent models at noise SD .75. Each has 32, 128 or 512 nested random update coordinates, 16 or 128 protection inputs, and full or restricted coverage. This is a new-model test on the same intervention grid used in development, not prediction of unseen capacities. Predictions are saved before evaluation inputs are drawn. Appendix G gives the protocol.

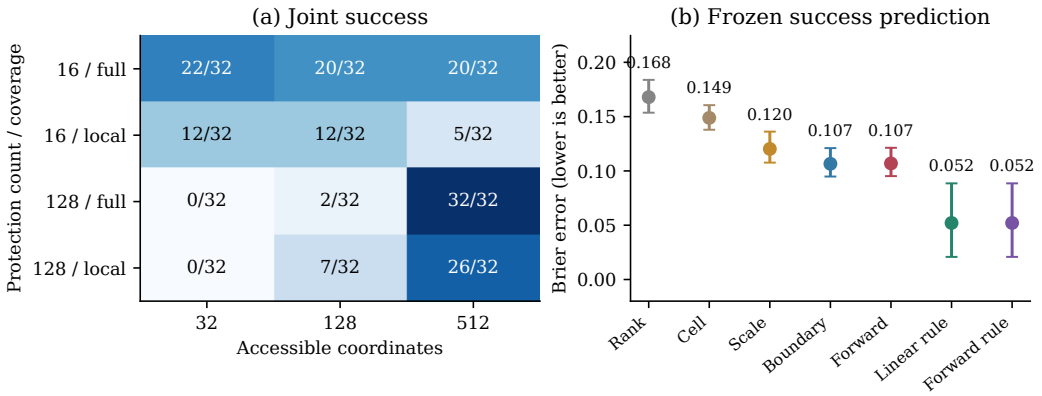

Figure 3: Coverage and update freedom. (a) Local-update success in 12 cells using the same 32 models: both updates descend, protected descent is at least 25% of ordinary descent, and output RMS change is at most 50% of ordinary change (Appendix G). Local protection covers the leftmost eighth of one latent coordinate. (b) All frozen predictors, including both raw rules; bars are 97.5% model-block bootstrap intervals for Brier error. Forward uses finite-step diagnostic changes. No evaluation outcome selects an update.

At 512 accessible coordinates and 128 protection inputs, joint success was 32/32 under full coverage and 26/32 under restricted coverage. The paired success-rate difference was 0.188 [0.062,0.344]. The prespecified coverage criterion was met. Only six pairs differ. A post-hoc exact paired test gives two-sided $p = .03125$, above .025; the bootstrap conclusion is sensitive to the inference method. It does not establish an effect of at least .15.

Scale-minus-boundary Brier error was 0.014 [0.004,0.023]. The prespecified predictive criterion was met. There were 158 successful and 226 unsuccessful cells; uncertainty is computed over 32 model blocks, not 384 independent trials. The raw linear and forward rules give identical classifications, with 20/384 errors each; both outperform calibration. This supports independent diagnostic transfer, not superiority over direct forward checks.

Appendix G.1 examines fitting-gradient retention, numerical cutoffs and exploratory screens without diagnostic labels.

6.7 Loss Placement on Real Regression Data

The red and white Wine Quality tasks (Cortez et al., 2009) each have 11 inputs and a sensory score. Standard MLP and Transformer models use no geometry or representation/prediction geometry at weights .05 and .5. Representation distances are divided by the square root of their dimension; equal weights do not match penalty strengths. Eight grouped 60/20/20 splits keep identical inputs together. Training data determine normalization; validation MSE selects checkpoints. The grid has 160 fits (Appendix I).

Table 1: Wine Quality test MSE in score units. Means over eight paired grouped splits; Rep. and Pred. use weight .5. Δ is Pred.$-$Rep.; brackets give descriptive 95% seed-bootstrap intervals. These splits overlap in observations.

Data	Model	No geometry	Rep.	Pred.	Δ [interval]
Red	MLP	0.401	0.395	0.395	+0.0003 [-0.0020, +0.0038]
Red	Transformer	0.405	0.410	0.405	-0.0048 [-0.0181, +0.0036]
White	MLP	0.515	0.516	0.509	-0.0065 [-0.0128, -0.0007]
White	Transformer	0.518	0.523	0.521	-0.0022 [-0.0107, +0.0047]

At weight .5, prediction-minus-representation MSE is .0003, $-.0048$, $-.0065$ and $-.0022$ for red MLP, red Transformer, white MLP and white Transformer, respectively. All four absolute mean differences are smaller than 2% of baseline MSE. Three descriptive intervals include zero; the white

MLP interval favors prediction geometry. Selecting weights by validation, including zero, also fails to show a consistent benefit for representation geometry (Table 29).

The differences reflect the chosen loss scales and optimization protocol as well as location. Their small means do not establish equivalence. Wine scores do not identify the conditional mean or individual rating variability, so this comparison checks training performance rather than the bias condition or the shared-update protection mechanism.

7 Conclusion and Limitations

Conditional push probabilities determine preferred pair distances. A descent direction at the conditional mean implies objective bias; for nonconstant means, net forces matter because edgewise mismatches can cancel. Intermediate representations allow further changes, subject to the encoder and head.

Synthetic interventions show a coverage–descent tradeoff that depends on the allowed update coordinates. Wine Quality training gives small, mixed loss-location effects. The experiments do not establish when conditional uncertainty materially changes trained prediction error.

The formulas are local, numerical cutoffs affect success counts, and output disturbance is not excess risk. Mean offsets explain part of the synthetic training effects. Wine scores aggregate ratings; equal weights do not match penalty strengths; repeated splits reuse observations. Neither loss location has a general recommendation here. A correction would need to distinguish predictable target differences from label variability.

Reproducibility Statement

Artifacts retain configurations, source hashes, splits, loss logs, checkpoints and predictions. Appendices document the earlier training and intervention studies. Coverage development uses 96 earlier models in 32 seed blocks; confirmation uses 32 new models and 384 cells. Predictors and endpoints are frozen before confirmation. Later audits and label comparisons are marked exploratory. The real-data grid fixes 160 fits before evaluation and retains all runs, grouped splits, losses and predictions.

AI Use Statement

Generative AI tools assisted literature checking, code development and debugging, experiment orchestration, derivation checks, figure generation, and manuscript revision. Numerical findings are computed by the recorded programs and retained in their artifacts. The authors are responsible for verifying the arguments, implementation, citations and submitted materials.

Ethics Statement

We use synthetic data and public Wine Quality data under CC BY 4.0. The released wine records contain physicochemical measurements and aggregate sensory scores; we collect no new human ratings.

REFERENCES

Mehdi Abbana Bennani, Thang Doan, and Masashi Sugiyama. Generalisation guarantees for continual learning with orthogonal gradient descent. *arXiv preprint arXiv:2006.11942*, 2020. URL https://arxiv.org/abs/2006.11942.

Christopher M. Bishop. Mixture density networks. Technical Report NCRG/94/004, Aston University, 1994.

Ching-Yao Chuang, R Devon Hjelm, Xin Wang, Vibhav Vineet, Neel Joshi, Antonio Torralba, Stefanie Jegelka, and Yale Song. Robust contrastive learning against noisy views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16670–16681, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chuang_Robust_Contrastive_Learning_Against_Noisy_Views_CVPR_2022_paper.html.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547–553, 2009. doi: 10.1016/j.dss.2009.05.016.

Jingyi Cui, Weiran Huang, Yifei Wang, and Yisen Wang. Rethinking weak supervision in helping contrastive learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6448–6467, 2023. URL https://proceedings.mlr.press/v202/cui23a.html.

Jingyi Cui, Yi-Ge Zhang, Hengyu Liu, and Yisen Wang. An inclusive theoretical framework of robust supervised contrastive loss against label noise. *arXiv preprint arXiv:2501.01130*, 2025. URL https://arxiv.org/abs/2501.01130.

Zijian Dong, Yilei Wu, Chongyao Chen, Yingtian Zou, Yichi Zhang, and Juan Helen Zhou. Improve representation for imbalanced regression through geometric constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5082–5091, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Dong_Improve_Representation_for_Imbalanced_Regression_through_Geometric_Constraints_CVPR_2025_paper.html.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3762–3773, 2020. URL https://proceedings.mlr.press/v108/farajtabar20a.html.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 1735–1742, 2006. doi: 10.1109/CVPR.2006.100.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Mahsa Keramati, Lili Meng, and R. Evans. ConR: Contrastive regularizer for deep imbalanced regression. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/704357127afabbd5a6eb979a57810767-Abstract-Conference.html.

Chris Dongjoo Kim, Sangwoo Moon, Jihwan Moon, Dongyeon Woo, and Gunhee Kim. Sample selection via contrastive fragmentation for noisy label regression. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e68c3a624d19154b28951e8690834607-Abstract-Conference.html.

Alessandro Ragano, Jan Skoglund, and Andrew Hines. SCOREQ: Speech quality assessment with contrastive regression. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bece7e02455a628b770e49fcfa791147-Abstract-Conference.html.

Raphael Rossellini, Rina Foygel Barber, and Rebecca Willett. Integrating uncertainty awareness into conformalized quantile regression. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1540–1548, 2024. URL https://proceedings.mlr.press/v238/rossellini24a.html.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: Learning continuous representations for regression. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/39e9c5913c970e3e49c2df629daff636-Abstract-Conference.html.

Dixian Zhu, Tianbao Yang, and Livnat Jerby. Gradient aligned regression via pairwise losses. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80259–80281, 2025. URL https://proceedings.mlr.press/v267/zhu25x.html.

Guide to the appendices. For the conditional-risk proofs and finite-design checks, see Appendices A–B. Synthetic training and earlier comparisons are in Appendices C–E and H. Shared updates and coverage tests are in Appendices F–G; Appendix G.1 contains the exploratory diagnostics. Real-data protocols and complete results are in Appendix I.

A Details of the Local Contrast Analysis

Batch conditioning and feasible distances. Consider an IID batch $\{(X_k, Y_k)\}_{k=1}^{B}$ and an anchor index chosen uniformly and independently of the data. Select its neighbor from the batch inputs, with a deterministic rule for ties. Conditional on $\mathcal{X}$, the encoder distance is fixed, whereas the push indicator can depend on every batch label. Taking its conditional expectation gives equation 4 without requiring independent pair indicators or a fixed threshold. Expectations over pairs in this appendix use this normalized sampling rule; multiplying the pair objective by a constant changes the effective λ.

For $D \in [0, r]$, completing the square gives $Q_p(D) = (D - pr)^2 + p(1-p)r^2$. For $D \geq r$, $Q_p(D) = (1-p)D^2$. If $0 < p < 1$, the latter is increasing and its smallest value is $(1-p)r^2$, which exceeds $Q_p(pr) = p(1-p)r^2$. For $p = 0$, $Q_0(D) = D^2$ has its unique minimum at zero. For $p = 1$, $Q_1(D) = (r - D)_+^2$ is zero exactly on $[r, \infty)$. These cases establish the global statement. In particular, $D = pr$ is always a minimizing value, but is not the unique one when $p = 1$.

Distances from one encoder must satisfy symmetry, triangle inequalities, the embedding dimension constraint, and $D(x, x) = 0$. Moreover, the same input pair may have different conditional push probabilities in different batch contexts. Minimizing every conditional pair term separately is therefore only a relaxation of the encoder problem. No realizability or optimization claim follows from this relaxation.

Proof of Proposition 1. Let $M = \operatorname{ess\,sup} |\Delta v| < \infty$. The assumption $C_v > 0$ implies $M > 0$. For $0 \leq t \leq r/M$, all hinge terms are in their quadratic region. Since $\mathbb{E}[Y \mid X] = 0$,

$$\mathbb{E}[(Y - tv(X))^2] - \mathbb{E}[Y^2] = t^2 \mathbb{E}[v(X)^2], \tag{11}$$

$$\mathbb{E}[Q_{p_{\mathcal{X}}}(t|\Delta v|) - Q_{p_{\mathcal{X}}}(0)] = t^2 \mathbb{E}[(\Delta v)^2] - 2rtC_v. \tag{12}$$

Put $B_v = \mathbb{E}[v(X)^2] + \lambda \mathbb{E}[(\Delta v)^2]$. This is finite and positive. Any sufficiently small $0 < t < \min\{r/M, 2\lambda r C_v / B_v\}$ gives $J_\lambda(tv) < J_\lambda(0)$. Since $\|tv\|_\infty$ tends to zero, the zero predictor is not a local minimum, even in the uniform norm. All predictors with zero excess squared risk equal zero P_X-almost surely and have the same pair risk under sampling from P_X. Consequently no such predictor is a global minimizer of J_λ. This proves the conditional assertion about global minimizers.

The result is about the combined population objective at fixed positive λ. It is not a lower bound on the risk of an arbitrary trained network. It gives no common positive excess-risk constant over predictor classes, no conclusion when λ vanishes with sample size, and no claim that gradient descent leaves the collapsed prediction. In particular, the norm in the pair loss is nonsmooth at collapse.

Two-input calculation. For the example in Section 4, the labels of a cross-input pair differ with probability $1/2$. Their difference is zero otherwise, so the threshold σ gives $p = 1/2$ exactly. The same calculation below applies to any cross-input rule with a common push probability $p \in [0, 1]$. For $f(X) = b + aX/h$,

$$J_\lambda(a, b) = \sigma^2 + b^2 + a^2 + \lambda\{4(1-p)a^2 + p(r - 2|a|)_+^2\}. \tag{13}$$

The minimizing intercept is zero. Set $s = |a|$. On $0 \leq s \leq r/2$, the nonconstant part of the objective is

$$(1 + 4\lambda)s^2 - 4\lambda prs + \lambda pr^2. \tag{14}$$

Its minimizer $s^* = 2\lambda pr/(1+4\lambda)$ lies strictly below $r/2$ for finite $\lambda > 0$. On $s \geq r/2$, the objective is $\sigma^2 + \{1 + 4\lambda(1-p)\}s^2$ and is increasing. Thus s^* is the global minimizing magnitude. Both signs give the same objective because the conditional mean is zero. The excess squared risk is $(s^*)^2$, which is positive exactly when $\lambda p > 0$. The arbitrary input scale h shows that the two inputs can be as close as desired without changing this calculation.

For the two-coordinate construction, $\|z(h) - z(-h)\| = pr$ and $H(z(X)) = 0$. Hence the squared risk is σ^2 and the pair risk is $p(1-p)r^2$. Each is a separate global minimum, so their sum is also globally minimal for every $\lambda \geq 0$. This is an existence result for the stated unrestricted representation and head. It does not say that a given neural architecture or training procedure finds this decomposition.

B Pair Forces and Neural Interventions

Finite-design objective and proof. The design has positive masses summing to one. Pair selection can depend on the full input batch. Once an unordered edge is selected, averaging its push indicator over labels and other batch members gives p_e. This yields the objective, up to irreducible target variance,

$$F(f) = (f-\mu)^\top \Pi (f-\mu) + \lambda \sum_e w_e Q_{p_e}(|(Bf)_e|/c). \tag{15}$$

This finite-design setting does not assume independent edge labels. Within the open region with signs s and hinge states a, expansion gives

$$F(f) = (f-\mu)^\top \Pi (f-\mu) + \lambda\{c^{-2} f^\top L f - 2rc^{-1} b^\top f + \text{constant}\}. \tag{16}$$

Its Hessian is $2(\Pi + \lambda c^{-2} L) > 0$. Setting the gradient to zero gives equation 9. If the solution remains in this region, it is its unique minimizer and a strict local minimum of the full objective. Otherwise this expression is an invalid extrapolation across an edge-order or hinge change. At μ, the gradient is $2\lambda g_c$. Nonzero g_c therefore excludes a local minimum, while zero g_c and the positive Hessian establish a strict local minimum. These are local claims, not a global optimization theorem for the nonconvex full objective.

Writing $A = \Pi^{-1/2} L \Pi^{-1/2}$ and $h = \Pi^{-1/2} g_c$, with eigenpairs (u_k, ℓ_k) of A, the exact in-region risk is

$$R(f_\lambda) - R^* = \lambda^2 \sum_k \frac{(u_k^\top h)^2}{(1 + \lambda \ell_k / c^2)^2}. \tag{17}$$

Thus both the net force and graph modes enter the risk. A push proportion alone does not specify either quantity. In the three-node example, orient the edges $(1,2), (2,3), (1,3)$. Their signed residuals are proportional to $(.1, .1, -.1)$, which is annihilated by $B^\top$. Edgewise mismatch can therefore coexist with zero net force and locally optimal Bayes prediction.

Since $B\mathbf{1} = 0$, also $\mathbf{1}^\top L = 0$ and $\mathbf{1}^\top b = 0$. Multiplying the normal equation by $\mathbf{1}^\top$ gives $\mathbf{1}^\top \Pi (f_\lambda - \mu) = 0$. More generally, for any pair objective invariant under a constant prediction shift, the derivative of population squared risk along that shift is $2\mathbb{E}[f - \mu]$. It vanishes at any local minimum in a translation-closed predictor class. This observation does not constrain a finite-sample, early-selected neural checkpoint.

Finite-design verification protocol. For each of eight seeds, we draw 24 points uniformly from $[-1, 1]^3$. The function $\sin(\pi x_1) + 0.5 x_2 x_3$ is centered over these points and scaled to RMS .2 or .8. Labels add independent Gaussian noise with standard deviation .15 or .6. We draw 4,096 batches of 12 distinct points, independently across batches, and find the nearest input neighbor and label-gap median within each batch. Frequencies estimate w_e and p_e. A separate bank of 8,192 batches provides independent estimates. Inputs use seeds $120000 + s$, estimation batches $130000 + s$, and independent batches $140000 + s$, $s = 0, \ldots, 7$. Sampling without replacement within a batch defines this finite-design probe; it is not the IID neural training distribution from the earlier study.

We cross $c \in \{.25, .5, 1\}$ and $\lambda \in \{.001, .01, .1\}$, giving 288 conditions, all retained. BFGS with analytic gradients starts at $f = \mu$, with gradient tolerance 10^{-10} and at most 1,500 iterations. Solver status and achieved gradient norms are saved. Direct optimization checks a local solution; it does not certify a global minimum. The sign and hinge conditions are checked explicitly before using the closed form. An additional test takes a step of Euclidean norm 10^{-5} against the estimated gradient at μ and evaluates the change on the independent batch bank. Both banks share the fixed design. This tests prediction of pair-objective changes, not generalization to new input points.

Of 288 conditions, 119 preserve the required sign and hinge region; the other 169 are outside the formula's scope. The valid counts are 76, 38 and 5 out of 96 at weights .001, .01 and .1. Within valid regions the maximum RMS discrepancy from same-bank optimization is 2.1×10^{-9}; the median discrepancy against independent-bank optimization is 9.6×10^{-5}. The estimated descent direction lowers the independent-bank objective in all 288 conditions. BFGS reports success in 173/288 estimation-bank and 178/288 independent-bank runs; the largest achieved gradient norms are 3.6×10^{-9} and 2.2×10^{-9}, respectively. Unsuccessful statuses are retained, and these checks do not certify global minima.

Table 2: Saved-model activation interventions. Entries average four batches within each checkpoint, then five seeds. $E_\perp$ is the fraction of geometry gradient energy orthogonal to the prediction Jacobian. Slopes fit log prediction change against log displacement over the three fixed scales. Permutation is RMS prediction change, not prediction improvement.

Task	Map	λ	$E_\perp$	Orth. slope	Aligned slope	Permutation
Branch	Exposed	0	0.886	1.97	0.93	1.037
Branch	Exposed	.5	0.932	1.98	0.85	0.969
Branch	Hidden	0	0.923	1.94	0.89	1.120
Branch	Hidden	.5	0.937	1.97	0.87	1.137
Noise	Exposed	0	0.990	2.00	1.00	0.013
Noise	Exposed	.5	0.990	2.00	0.99	0.013
Noise	Hidden	0	0.991	2.00	0.99	0.015
Noise	Hidden	.5	0.991	2.00	0.99	0.015

Saved neural checkpoint interventions. We use both observation maps, pure-noise and branch-$\kappa = .32$ tasks, $n = 1024$, raw-token weights $0, .5$, and all five original seeds: 40 checkpoints. Each uses four disjoint test batches of 256, at indices 1024–2047. Models run in evaluation mode and FP64; parameters and model selection are unchanged. Each prediction depends only on its own sample, so the activation Jacobian is block diagonal across the batch. For row J_i, the projection is

$$g_{\parallel,i} = \frac{g_{z,i}^\top J_i}{\|J_i\|^2} J_i, \qquad g_{\perp,i} = g_{z,i} - g_{\parallel,i}, \tag{18}$$

with zero aligned component for a zero row. Denominators use a numerical floor 10^{-30}. The negative projected gradients and an independent Gaussian direction are normalized to equal RMS vector displacement. Perturbation sizes are $.001, .01, .1$ times the centered token RMS norm. We record geometry loss, prediction RMS change, and direct oracle error. A within-batch permutation preserves the token marginal distribution while changing its correspondence to the input. It tests dependence on the token, not whether using that dependence improves prediction.

The projection predicts a zero linear term in output change and a negative linear term in geometry when $g_\perp \neq 0$. Under a smooth prediction map, the output remainder is second order. We check finite perturbations, since the linear approximation need not remain accurate at larger scales. Four batches are averaged within each checkpoint before summaries across checkpoints; they are not extra independent training seeds. All 40 checkpoints are retained, including failures of finite-scale descent. These sample-dependent interventions are not a new encoder, and make no claim of improved deployable predictions or realizability by shared weights.

Existing pure-noise prediction decomposition. For each saved predictor and its training-standardized constant mean μ_0, we use the exact finite-test identity

$$\widehat{E} = (\bar{f} - \mu_0)^2 + N^{-1} \sum_i (f_i - \bar{f})^2. \tag{19}$$

This separates a test-input mean offset from prediction variation; it is not the statistical bias–variance decomposition over training samples. It is a post-hoc analysis of the existing 30 pure-noise runs. Pair-loss translation invariance does not exclude indirect mean changes during optimization or validation selection. The observed mean-offset contribution therefore limits the earlier causal interpretation of total excess error. We also compute nearest-input prediction distances on 16 disjoint test batches of 256 per saved predictor. These are post-hoc checkpoint statistics, not training trajectories. Push labels use the same batch-median target-gap rule. Table 3 retains every paired seed.

C Mechanism Study Protocol

Hidden observation map. For each map seed, draw independent frequencies ω_j uniformly on $[0.7, 3]$ and phases α_j uniformly on $[0, 2\pi]$, for $j = 1, \ldots, 16$. The hidden map is

$$\phi_{\text{hidden},j}(u) = \sin(\omega_j u + \alpha_j) + 0.35 \sin\{(0.5\omega_j + 1.3)u + 0.5\alpha_j\}. \tag{20}$$

An independent standard Gaussian vector is divided by its Euclidean norm to give the branch direction b_{hidden}. The exposed map uses $[v(u), v(u)W]$, with $v(u) = [u, u^2, \sin(2u), \cos(3u)]$ and

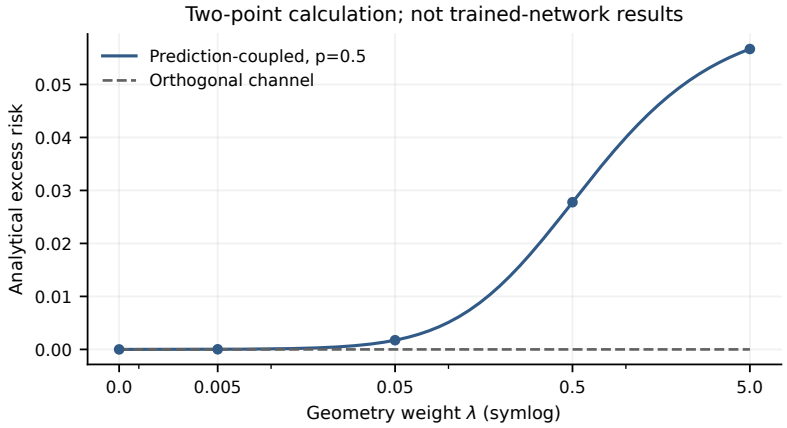

Figure 4: The earlier two-input calculation with $p = 1/2$ and margin 1. This uses a fixed cross-input pair rule, not the batch-median neural rule.

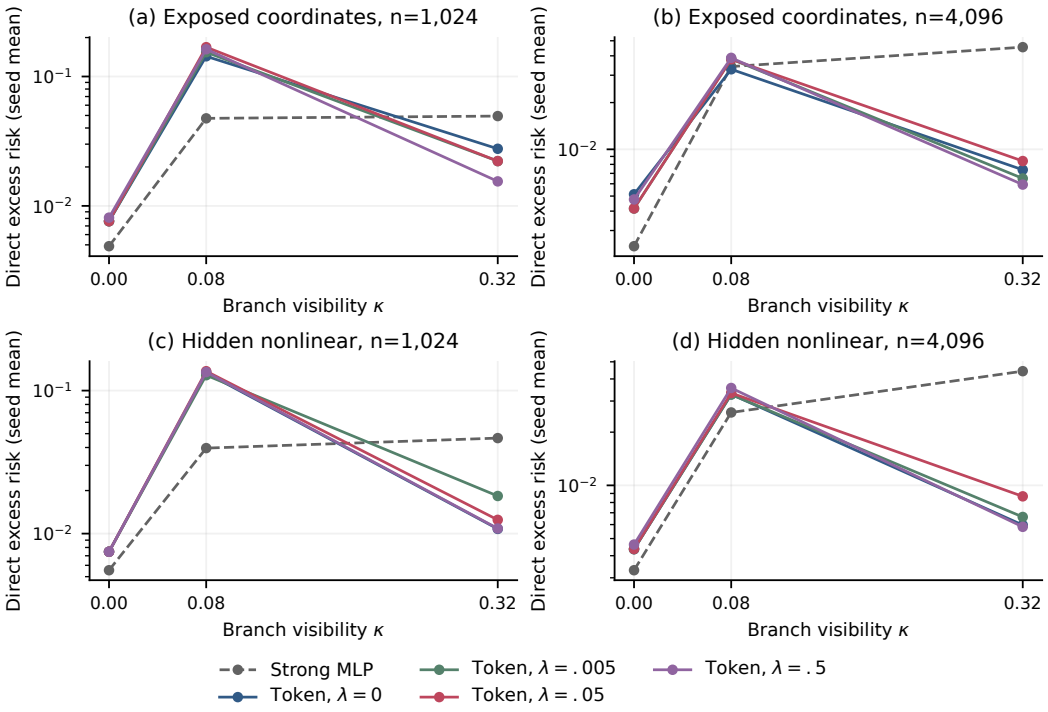

Figure 5: The earlier 390-fit study: direct excess across visibility, sample size and map. Lines show seed means; fixed weights are descriptive.

Table 3: Post-hoc nearest-input-pair analysis of saved pure-noise predictions. All entries are output geometry minus raw-token geometry at weight .5, multiplied by 10^3. Each seed averages 16 fixed disjoint test batches of 256. Distances are absolute prediction differences, using the same input pairs and batch-median push labels across variants.

Map	Seed	All distance	Push distance	Pull distance	Output geometry
Exposed	0	+0.356	+0.369	+0.343	-0.363
Exposed	1	+1.549	+1.809	+1.292	-1.652
Exposed	2	+0.142	+0.152	+0.132	-0.151
Exposed	3	+0.518	+0.546	+0.489	-0.525
Exposed	4	-1.773	-1.775	-1.770	+1.625
Hidden	0	-0.718	-0.893	-0.544	+0.389
Hidden	1	+0.573	+0.570	+0.577	-0.550
Hidden	2	+0.045	+0.041	+0.048	-0.041
Hidden	3	+0.689	+0.725	+0.654	-0.671
Hidden	4	-0.211	-0.247	-0.175	+0.242

independent $W_{jk} \sim \mathcal{N}(0, 1/4)$. Its branch direction is the last coordinate vector. Maps, directions and feature/target normalization are shared across the three data splits of each run. Each coordinate is centered by its training mean and divided by $s_{\mathrm{tr}} + 10^{-6}$, where s_{tr} is its training standard deviation using divisor n. We write $s_{y,\mathrm{tr}}$ for this stabilized target scale. The two mechanisms share the same latent and noise sampling streams when their sample seeds agree, but use different input maps. Sample-size conditions are not nested training sets.

Sampling and initialization. For repetition $s \in \{0, \ldots, 4\}$, the map seed is $50000 + s$, and the training, validation and test sample seeds are $60000{+}s$, $70000{+}s$ and $80000{+}s$. Model initialization and shuffled batches use $90000 + s$. Pure-noise labels use an independent stream obtained by adding 10^6 to the corresponding sample seed. Thus each method in a data cell sees exactly the same cached arrays, and matching token predictors start with the same parameters. The output-geometry wrapper adds no parameters or randomness. The separate seeds document the random sources; this design does not estimate separate map, sample and optimization variance components.

Budget and checkpoint selection. The optimization schedule uses AdamW, learning rate 10^{-3}, weight decay 10^{-4}, and batch size 256. Each run has at most 120 epochs and stops after 12 epochs without validation-MSE improvement, once at least 15 epochs have been trained. The checkpoint with minimum validation MSE is restored for test evaluation. No test target or oracle conditional mean enters training or selects an epoch. The token and its Transformer widths are unchanged across loss locations. The Strong MLP has four hidden layers of width 128.

The fixed weights are $0, 0.005, 0.05, 0.5$ on raw tokens. Output geometry uses $0.05, 0.5$ at training size 1,024 in the branch study. Pure-noise controls use the unregularized token and raw-token/output geometry at weight 0.5. All geometry terms have margin 1. Equal nominal weights do not equalize representation norms or gradient magnitudes. Consequently the comparisons identify these implemented objectives, not an effect of location isolated from every scale difference.

Conditional moments and direct excess. We integrate the known joint posterior over $u \in [-2, 2]$ and sum over $c \in \{-1, 0, 1\}$, using each mechanism's map and branch direction in its Gaussian input likelihood. The branch prior is $(q/2, 1-q, q/2)$. Simpson quadrature uses 2,049 nodes; 4,097 nodes check the first 128 test inputs of each cached branch cell. For pure noise, the raw conditional mean is zero and variance is exactly $qG^2 + \sigma_y^2$. After training normalization, the conditional mean is $(\mu(x) - \bar{y}_{\mathrm{tr}})/s_{y,\mathrm{tr}}$; it is not generally zero in standardized coordinates, even in the pure-noise case. We compute direct excess against this transformed conditional mean.

Statistics and selection. Within each map, visibility and training-size cell, differences pair the same five repetitions. We report all differences and their median, and use the existing exact two-sided signed-rank sign-randomization calculation. Its null assumptions include independent repetition differences and symmetry. Five nonzero differences give a minimum attainable two-sided $p = 0.0625$. Bootstrap intervals are exploratory at this replication count; neither overlapping intervals

nor a nonsignificant test establishes equivalence. Cross-cell comparisons are descriptive and are not counted as additional independent replicates of the same effect.

For the optional selected-weight summary, each repetition chooses among the four raw-token weights using its minimum validation MSE. Test metrics do not enter the selection criterion. This reuses the validation split used for checkpoint selection and spends four training runs per cell. It is reported separately from fixed-weight comparisons, with no claim of equal tuning cost against the one-configuration MLP.

D Extended Mechanism Results

Table 4: Direct oracle excess for all branch-task settings. Entries summarize the five repetitions as mean ± seed SD under fixed raw-token geometry weights.

Model / weight	$\kappa = 0.0$	$\kappa = 0.08$	$\kappa = 0.32$
Exposed coordinates, $n = 1,024$			
Strong MLP	0.00488 ± 0.00172	0.047 ± 0.022	0.049 ± 0.00408
Token, $\lambda = 0$	0.00761 ± 0.00195	0.143 ± 0.098	0.028 ± 0.025
Token, $\lambda = .005$	0.00761 ± 0.00195	0.154 ± 0.100	0.022 ± 0.021
Token, $\lambda = .05$	0.00761 ± 0.00195	0.169 ± 0.107	0.022 ± 0.019
Token, $\lambda = .5$	0.00811 ± 0.00157	0.162 ± 0.118	0.015 ± 0.010
Exposed coordinates, $n = 4,096$			
Strong MLP	0.00238 ± 0.00113	0.034 ± 0.019	0.045 ± 0.00412
Token, $\lambda = 0$	0.00514 ± 0.00216	0.033 ± 0.016	0.00738 ± 0.00291
Token, $\lambda = .005$	0.00416 ± 0.00192	0.038 ± 0.022	0.00649 ± 0.00240
Token, $\lambda = .05$	0.00416 ± 0.00192	0.038 ± 0.017	0.00841 ± 0.00395
Token, $\lambda = .5$	0.00476 ± 0.00181	0.039 ± 0.020	0.00593 ± 0.00351
Hidden nonlinear, $n = 1,024$			
Strong MLP	0.00554 ± 0.00178	0.040 ± 0.014	0.047 ± 0.00324
Token, $\lambda = 0$	0.00747 ± 0.00179	0.133 ± 0.128	0.011 ± 0.00311
Token, $\lambda = .005$	0.00747 ± 0.00179	0.128 ± 0.131	0.018 ± 0.021
Token, $\lambda = .05$	0.00747 ± 0.00179	0.137 ± 0.126	0.012 ± 0.00427
Token, $\lambda = .5$	0.00747 ± 0.00179	0.135 ± 0.127	0.011 ± 0.00337
Hidden nonlinear, $n = 4,096$			
Strong MLP	0.00331 ± 0.00143	0.026 ± 0.00822	0.044 ± 0.00296
Token, $\lambda = 0$	0.00436 ± 0.00162	0.033 ± 0.012	0.00596 ± 0.00291
Token, $\lambda = .005$	0.00436 ± 0.00162	0.033 ± 0.011	0.00663 ± 0.00333
Token, $\lambda = .05$	0.00436 ± 0.00164	0.033 ± 0.012	0.00867 ± 0.00502
Token, $\lambda = .5$	0.00462 ± 0.00209	0.036 ± 0.013	0.00583 ± 0.00366

Table 5: Paired effect of raw-token geometry at weight 0.05 versus weight zero, using 4,096 training samples. Positive differences indicate increased direct oracle excess. Exact tests have five paired repetitions.

Observation	κ	Mean	Median	Seed range	Exact p
Exposed	0.0	-0.00098	-1.2e-05	[-0.00485, 7e-07]	0.1250
Exposed	0.08	0.00528	0.00394	[0.00038, 0.014]	0.0625
Exposed	0.32	0.00102	0.00055	[-0.00316, 0.00774]	1.0000
Hidden	0.0	9.2e-06	-3.6e-06	[-8e-06, 5.9e-05]	0.8125
Hidden	0.08	0.00056	-0.00066	[-0.00457, 0.00727]	1.0000
Hidden	0.32	0.00270	0.00294	[0.00011, 0.00524]	0.0625

Table 6: Prediction-level minus raw-token geometry at weight 0.5, using 1,024 training samples. Positive differences indicate higher excess with prediction-level geometry. Both use the same token predictor.

Observation	κ	Mean	Median	Seed range	Exact p
Exposed	0.0	0.00210	0.00130	[0.00027, 0.00406]	0.0625
Exposed	0.08	-0.012	0.00107	[-0.101, 0.032]	0.6250
Exposed	0.32	0.023	0.021	[0.013, 0.044]	0.0625
Hidden	0.0	0.00437	0.00375	[0.00096, 0.00799]	0.0625
Hidden	0.08	-0.00212	-0.00362	[-0.280, 0.282]	0.6250
Hidden	0.32	0.032	0.035	[0.019, 0.043]	0.0625

Table 7: Pure-noise controls. Direct excess uses the exact conditional mean after training normalization. Paired differences and tests compare each regularized variant with the unregularized token predictor. Absolute excess entries are mean $\pm$ seed SD.

Observation	Location, weight	Excess risk	Mean Δ	Median Δ	Exact p
Exposed	Token, 0	0.00528 ± 0.00372	—	—	—
Exposed	Token, .5	0.00535 ± 0.00386	7e-05	1.1e-05	0.6250
Exposed	Output, .5	0.00735 ± 0.00471	0.00208	0.00077	0.0625
Hidden	Token, 0	0.00810 ± 0.00611	—	—	—
Hidden	Token, .5	0.00810 ± 0.00610	2.5e-06	1.7e-06	0.4375
Hidden	Output, .5	0.00825 ± 0.00585	0.00015	0.00024	0.8125

Table 8: Validation-selected raw-token geometry weights. Selection includes weight zero and uses four candidate runs per repetition and data cell. Weight counts give selection frequencies for $\lambda = 0, 0.005, 0.05, 0.5$, in that order. Excess entries are mean $\pm$ seed SD.

n	κ	Weight counts	Token, 0	Selected token	MLP
Exposed coordinates					
1024	0.0	4/0/0/1	0.00761 ± 0.00195	0.00761 ± 0.00195	0.00488 ± 0.00172
1024	0.08	2/0/1/2	0.143 ± 0.098	0.134 ± 0.107	0.047 ± 0.022
1024	0.32	1/0/1/3	0.028 ± 0.025	0.013 ± 0.00805	0.049 ± 0.00408
4096	0.0	3/0/0/2	0.00514 ± 0.00216	0.00512 ± 0.00218	0.00238 ± 0.00113
4096	0.08	2/2/0/1	0.033 ± 0.016	0.034 ± 0.017	0.034 ± 0.019
4096	0.32	0/2/1/2	0.00738 ± 0.00291	0.00558 ± 0.00198	0.045 ± 0.00412
Hidden nonlinear					
1024	0.0	3/0/0/2	0.00747 ± 0.00179	0.00746 ± 0.00179	0.00554 ± 0.00178
1024	0.08	1/3/0/1	0.133 ± 0.128	0.128 ± 0.131	0.040 ± 0.014
1024	0.32	1/2/0/2	0.011 ± 0.00311	0.00930 ± 0.00258	0.047 ± 0.00324
4096	0.0	1/1/0/3	0.00436 ± 0.00162	0.00462 ± 0.00209	0.00331 ± 0.00143
4096	0.08	2/1/1/1	0.033 ± 0.012	0.033 ± 0.011	0.026 ± 0.00822
4096	0.32	1/0/1/3	0.00596 ± 0.00291	0.00576 ± 0.00353	0.044 ± 0.00296

Table 9: Model size and descriptive runtime records. Runs share a GPU with another worker; these timings include that contention and are not isolated speed benchmarks.

Model	n	Parameters	Epochs	Fit (s)
Strong MLP	1024	51,841	64	2.2
Strong MLP	4096	51,841	44	5.0
Token, λ=0	1024	608,961	69	7.2
Token, λ=0	4096	608,961	48	14.8
Token, λ=.005	1024	608,961	69	7.8
Token, λ=.005	4096	608,961	48	18.4
Token, λ=.05	1024	608,961	69	8.0
Token, λ=.05	4096	608,961	46	17.0
Token, λ=.5	1024	608,961	70	8.0
Token, λ=.5	4096	608,961	48	17.2
Output, λ=.05	1024	608,961	71	8.6
Output, λ=.5	1024	608,961	54	6.2

Recorded hardware: NVIDIA GeForce RTX 4060 Laptop GPU. Timing and memory entries are medians across the branch-task fits in each model–sample-size group. One test pass evaluates 4096 inputs with batch size 256; each run averages 3 timed repetitions, including host transfers. Peak memory is the process-local CUDA allocator's peak allocated memory during fitting, not total device memory. Fits ran concurrently with different early-stopping lengths; these records are descriptive, not isolated speed or memory benchmarks.

Table 10: Descriptive inference and peak GPU allocation records. Inference uses the full test split in batches of 256, including host transfers. Concurrent execution prevents interpreting these as isolated speed benchmarks.

Model	n	Runs	Test pass (ms)	Peak (MiB)
Strong MLP	1024	30	12.60	68.8
Strong MLP	4096	30	11.85	68.8
Token, λ=0	1024	30	65.15	207.6
Token, λ=0	4096	30	65.24	207.6
Token, λ=.005	1024	30	65.25	207.7
Token, λ=.005	4096	30	65.17	207.7
Token, λ=.05	1024	30	66.51	207.7
Token, λ=.05	4096	30	65.02	207.7
Token, λ=.5	1024	30	66.00	207.7
Token, λ=.5	4096	30	65.90	207.7
Output, λ=.05	1024	30	64.74	207.5
Output, λ=.5	1024	30	43.95	207.5

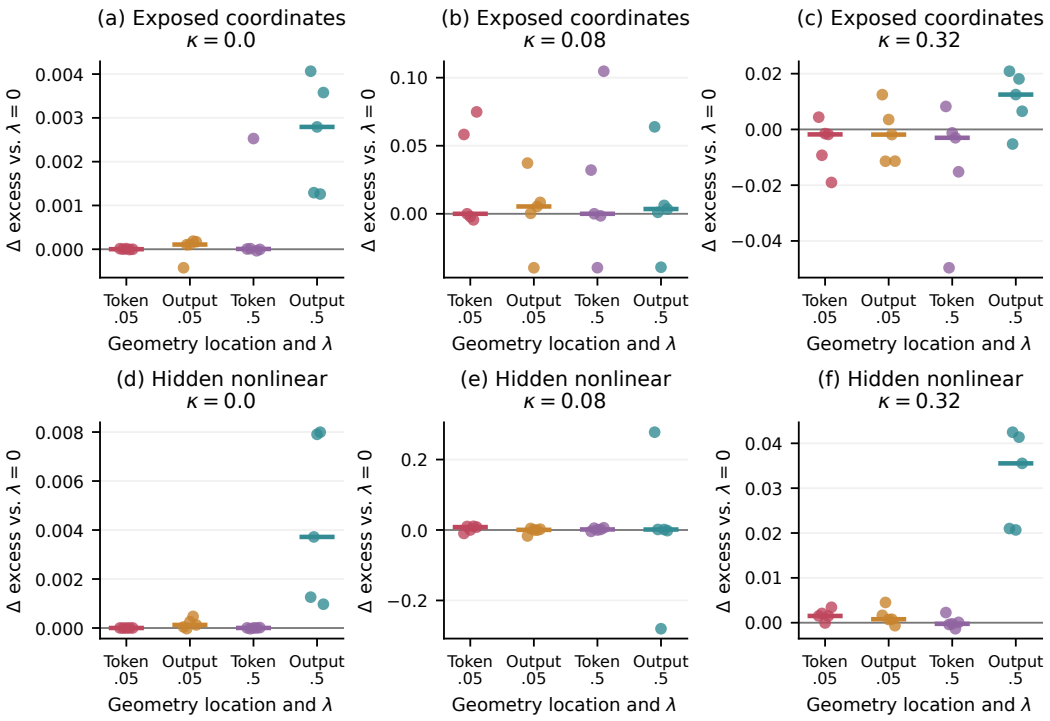

Figure 6: Geometry on raw tokens or scalar predictions relative to the same unregularized token predictor. Each point is a paired repetition; all branch tasks use nonconstant conditional means and 1,024 training samples.

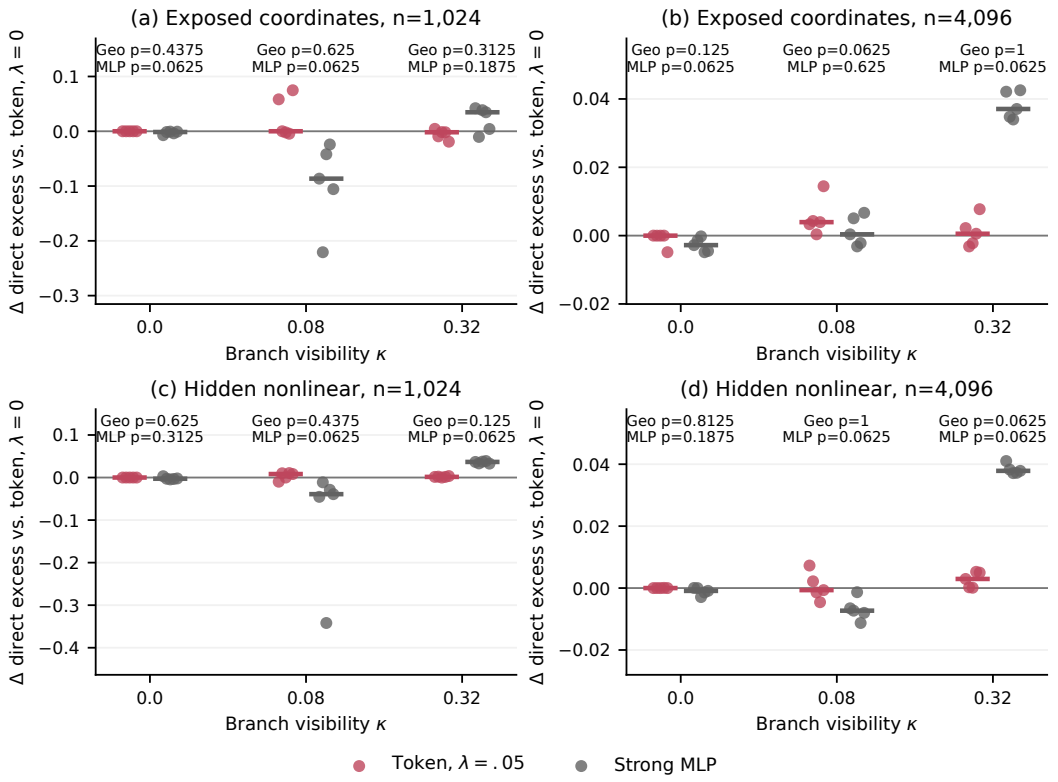

Figure 7: Individual paired changes in direct oracle excess for raw-token geometry at weight 0.05 and the Strong MLP, each relative to the unregularized token predictor. Medians and exact two-sided signed-rank tests summarize five repetitions per cell. Positive values indicate higher excess error.

E Additional Comparisons from the Earlier Mechanism Study

E.1 Weights and Partially Predictable Targets

At $n = 4096$, changing raw-token weight from 0 to 0.05 gave paired mean changes in direct excess risk from -0.00098 to 0.00528 across the 6 observation–visibility cells. Mean risk was higher in 5 cells and lower in 1. These counts describe the fixed grid; the cells are not independent replications.

The unregularized token has lower mean excess than the MLP at $\kappa = 0.32$ under both maps and sample sizes. At $\kappa = 0$, the MLP has lower mean excess in all four conditions. These changing rankings concern the base predictors; the token comparisons do not require a geometry objective. The complete curves are retained in Figure 5.

At $n = 1024$ and $\lambda = 0.5$, moving the same pair loss from the raw token to the scalar prediction gave paired mean changes in direct excess risk from -0.012 to 0.032. Mean risk was higher in 4 of 6 cells and lower in 2. This is a diagnostic change in how strongly the loss constrains prediction, not an alternative recommended training objective.

The branch tasks have nonconstant conditional means, including when the branch itself is unobserved. Their ordering is an empirical boundary test, not a consequence of the constant-mean proposition. A token and a prediction-level penalty also act on differently scaled quantities; equal numerical weights do not make their effective regularization strength equal.

E.2 Paired Evidence and Weight Selection

In Figure 7, points show all five paired seeds and bars show their median. Two-sided exact signed-rank sign randomization is exploratory and assumes independent, symmetrically distributed seed differences under the null. With five nonzero differences its smallest possible p-value is 0.0625.

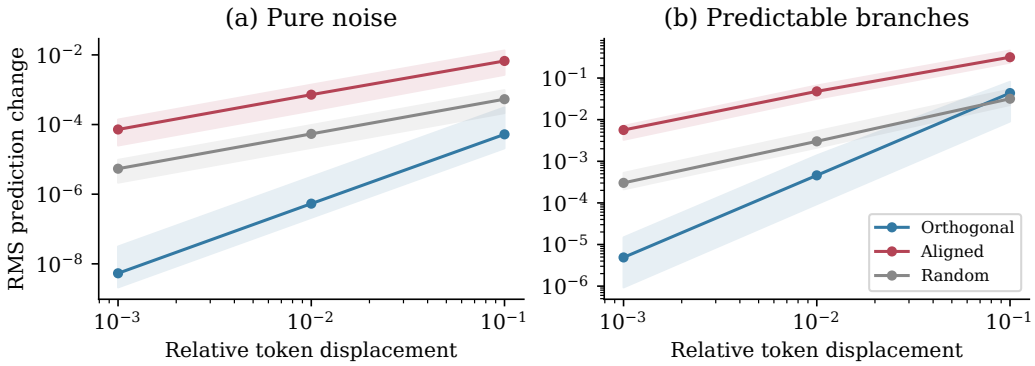

Figure 8: Norm-matched token interventions in saved predictors. Lines show checkpoint medians; shading spans the minimum to maximum checkpoint mean across 20 checkpoints per task. Orthogonal changes have a smaller, approximately second-order prediction response. Shading is descriptive, not a confidence interval; the four batches per checkpoint are averaged first.

We report no confirmatory discoveries across this grid; neither a large p-value nor a small observed difference establishes equivalence.

Validation MSE selected a nonzero token weight in 39 of 60 observation–visibility–sample-size–seed cells. Selection used neither test labels nor the test oracle. All four candidates were trained, so this descriptive result has four times the candidate budget of a single fixed-weight run and is not a budget-matched comparison with the MLP.

Appendix D reports paired effect summaries, exact tests, full fixed-weight tables and selected configurations.

The earlier shared-map study remains a separate control. ALDT, the token without geometry, an ordinary auxiliary MLP and a strong MLP had comparable mean errors; the geometry term did not lower the token's mean error in its partially informative setting. Appendix H reports these results and their five-seed uncertainty. They motivated this study, but are not new replications of its different maps and sampling seeds.

F SHARED-PARAMETER INTERVENTIONS AND FROZEN PREDICTIONS

Parameter compatibility. Condition on a trained model and the data used to construct a direction. Let $J(x) = \nabla_\theta f_\theta(x)^\top$ and assume a second-order Taylor remainder bounded in $L^2(P_X)$ in a neighborhood of θ. Then $f_{\theta+\eta v} - f_\theta = \eta J v + O_{L^2}(\eta^2)$, so

$$D(\eta, v) = \eta^2 v^\top C v + O(\eta^3), \qquad C = \mathbb{E}[J(X)^\top J(X)]. \tag{21}$$

Since C is positive semidefinite, $v^\top C v = 0$ iff $v \in \ker C$. A vector in this kernel with $g^\top v < 0$ exists iff the projection of g onto the kernel is nonzero: take $v = -P_{\ker C} g / \|P_{\ker C} g\|$. If the projection is zero, g is orthogonal to every such vector. This elementary compatibility criterion does not imply that a finite protection sample identifies the population kernel.

More precisely, if $J_A v = 0$ and $\|v\| = 1$, then

$$v^\top C v = v^\top (C - \widehat{C}_A) v \le \|C - \widehat{C}_A\|_{\mathrm{op}}, \qquad \widehat{C}_A = J_A^\top J_A / |A|. \tag{22}$$

The inequality is not an informative guarantee without covariance coverage. We therefore measure $J_B v$ on independent inputs rather than treating a small protection residual as evidence of population invariance. For a direction fixed before drawing B, $|B|^{-1}\|J_B v\|^2$ is unbiased for $v^\top C v$; its variance is $\mathrm{Var}[(J(X)v)^2]/|B|$ when the fourth moment is finite. Taking its square root does not preserve unbiasedness.

Disturbance versus prediction error. For $\Delta f = f_{\theta+\eta v} - f_\theta$, the exact squared-risk change is

$$R(f_\theta + \Delta f) - R(f_\theta) = 2\mathbb{E}[(f_\theta - \mu)\Delta f] + \mathbb{E}[(\Delta f)^2]. \tag{23}$$

Only at a Bayes-realizing predictor does the cross term vanish automatically. Our predictive endpoint is disturbance, not a prediction of improvement or harm. This avoids assigning a risk interpretation to a Jacobian norm alone.

Existing tools and the empirical question. OGD projects parameter gradients to preserve previous outputs (Farajtabar et al., 2020); related analyses study its behavior in the neural tangent regime (Bennani et al., 2020). Our projections and ridge filtering are controls using the same principle, applied to a single regression model's geometry gradient. We do not reproduce a continual-learning benchmark or claim a new version of OGD. The open empirical question here is whether shared updates retain independent-input geometry descent, how finite-sample protection transfers, and whether a separately measured directional response predicts new conditions better than scalar scale information.

Development and new models. Five previously trained exposed-map, branch-$\kappa = .32$, $n = 1024$, unregularized-token checkpoints fit the predictive baselines. Their seeds are 0–4. After these coefficients are frozen, confirmation uses seeds 100–131 and three conditions: exposed branch .32, hidden branch .08, and a four-latent task. Each condition has 32 newly fitted models; conditions share the repetition index and are not counted as independent seeds. All models use the existing token architecture, MSE-only AdamW and validation MSE selection, with 1,024 training and 1,024 validation observations, 120 epochs maximum, patience 12 and at least 15 executed epochs. Learning rate is 10^{-3}, weight decay 10^{-4} and batch size 256. Training logs and selected checkpoints are retained. Intervention inputs never select the training checkpoint.

For the new task, $u \sim \mathrm{Unif}([-1, 1]^4)$, and

$$x = [u, \sin(uW)], \quad W_{jk} \sim \mathcal{N}(0, 1/4), \qquad y = .7\sin(\pi u_1) + .5u_2u_3 + .3u_4^2 + \epsilon, \quad \epsilon \sim \mathcal{N}(0, .5^2). \tag{24}$$

The same $W \in \mathbb{R}^{4\times 12}$ is used across splits of one repetition. Because u is observed, the conditional mean is known exactly. This task changes the latent dimension and regression function, but remains synthetic. All features and labels use training means and stabilized standard deviations.

New map and initialization seeds are $200000 + s$ and $300000 + s$. Train, validation, geometry, protection, diagnostic, evaluation and trajectory sampling use seeds $210000 + s, \ldots, 270000 + s$ in that order. Development retains the original map and normalization, with these fresh post-training sampling streams. The respective post-training split sizes are 64, 128, 128, 1,024 and 512. The last split supplies eight disjoint batches of 64. Pair geometry is measured in fixed evaluation batches of 64.

Shared encoder updates. Only ϕ, the two-layer $16 \to 64 \to 128$ GELU encoder with 9,408 parameters, can change. The backbone, head and other parameters are fixed. Prediction Jacobians are computed in FP64 by the chain rule through ϕ and the backbone's token Jacobian. Every model has an autograd comparison and a central finite-difference check; numerical tolerances are 10^{-9} absolute and 10^{-3} relative, respectively.

Let g be the geometry gradient from the 64 fitting inputs. We compare $-g$, two protected directions, a ridge direction and a Gaussian direction. For protection size $n = 32$ or 128, write $J_A J_A^\top = U\Lambda U^\top$. The protected direction is

$$v_{\text{null}}^{\text{raw}} = -g + J_A^\top U_+ \Lambda_+^{-1} U_+^\top J_A g, \tag{25}$$

retaining eigenvalues larger than 10^{-8} times the largest. Ranks and residuals are recorded; protection applies to these numerical modes. The ridge control replaces the inverse with $(J_A J_A^\top + .01\lambda_{\max}(J_A J_A^\top)I)^{-1}$ and uses 128 inputs. Normalize each resulting direction to unit Euclidean norm. Each update has length $\eta = \alpha\|\theta\|$, with primary $\alpha = .001$ and secondary $\alpha = .01$. Thus a smaller parameter update cannot explain the comparison. The random direction uses seed $310000 + s$. Norm matching controls length in this fixed parameterization; it is not invariant to reparameterizing the encoder.

Eight-step paths on all 32 new exposed models compare ordinary and ridge updates at primary scale. Every step uses a fresh geometry batch, recomputes the current protection Jacobian on the fixed protection inputs, and has length $.001\|\theta_0\|$. All eight steps are retained. Separate geometry, MSE and prediction-change records and final encoder states are saved. These paths are geometry-only interventions, not a proposed replacement for joint MSE–geometry training.

Prespecified endpoints and precision. The primary feasibility endpoint is a joint success indicator for the ridge update on exposed branch .32 at $\alpha = .001$: both ordinary and ridge updates lower independent-input geometry, ridge retains at least 25% of the ordinary decrease, and its prediction RMS change is at most 50% of ordinary. A non-descending ordinary update counts as a failure, not an exclusion. The majority-evidence gate is a 97.5% Wilson lower bound above .5. With 32 repetitions the worst-case Wilson half-width is about .184, below the predeclared .20 target. This is a success-rate precision target, not a guarantee of precision for every continuous statistic. Repetition counts are not extended in response to significance.

The predictive target is log RMS prediction change on the evaluation inputs. The directional diagnostic $\eta\|J_B v\|/\sqrt{128}$ uses no fitted coefficient. The isotropic comparator is $\eta\|J_B\|_F/\sqrt{128P}$, $P = 9408$. Two log-linear baselines are fit only on the five development checkpoints: step norm alone; and step norm, geometry gradient norm, centered token RMS, parameter norm, Jacobian RMS, validation MSE and method indicators. Continuous features are logged and all features standardized using development statistics. The intercept is unpenalized; the other coefficients have ridge 10^{-6}. Development uses both preset scales and all methods. The frozen coefficient hash is recorded before any confirmation run.

For each confirmation seed, squared-log-error differences average the four nonrandom methods at primary scale. We report paired 97.5% bootstrap intervals with 10,000 seed-block resamples. Transfer requires a positive lower bound against the full scale baseline in each held-out condition. The feasibility and predictive endpoints use 97.5% intervals to allocate the two primary error budgets; bootstrap intervals remain approximate. Secondary scales, individual methods, medians and leave-one-seed-out means are reported without promoting them to replacement primary endpoints.

The prespecified joint success counts are Exposed 29/32, Hidden 32/32, Four-latent 28/32. For the primary exposed condition, the 97.5% Wilson interval is [0.730,0.972]. Its leave-one-seed-out success fraction ranges from 0.903 to 0.935.

For Exposed, the scale-minus-directional error advantage is 0.242 (median 0.260); its leave-one-seed-out mean ranges from 0.207 to 0.280, with 25/32 positive seed effects.

For Hidden, the scale-minus-directional error advantage is 0.436 (median 0.337); its leave-one-seed-out mean ranges from 0.395 to 0.449, with 32/32 positive seed effects.

For Four-latent, the scale-minus-directional error advantage is 0.461 (median 0.329); its leave-one-seed-out mean ranges from 0.397 to 0.478, with 31/32 positive seed effects.

All 96 parameter Jacobians pass the direct checks: maximum absolute autograd discrepancy 5.11×10^{-15} and maximum relative central-difference discrepancy 3.16×10^{-5}.

Despite a near-zero linear response on the 128 protection inputs, the same update has nonzero response on independent inputs. A descriptive comparison of the two preset step sizes gives median response slopes 0.994, 1.006, 1.010 for exposed, hidden and four-latent tasks (Figure 9). The reduction transfers, but first-order invariance does not; the activation experiment's second-order response cannot be assigned to these shared updates.

At trajectory step 8, the median paired ridge-to-ordinary geometry-decrease ratio is 0.952 and the prediction-response ratio is 0.161. These path summaries are secondary.

Table 11: Shared-parameter interventions at the primary step size. Prediction ratios are relative to the paired ordinary update and show median [IQR]. Geometry changes are seed means on independent evaluation batches. Joint success uses the three prespecified requirements; no seeds are omitted.

Condition	Direction	Prediction ratio	Δgeometry	Descent	Joint
Exposed	Ordinary	1.000 [1.000,1.000]	-6.56×10^{-4}	32/32	0/32
Exposed	Protect 32	0.156 [0.081,0.301]	-5.88×10^{-4}	32/32	27/32
Exposed	Protect 128	0.032 [0.017,0.052]	-5.19×10^{-4}	32/32	32/32
Exposed	Ridge 128	0.161 [0.103,0.238]	-6.24×10^{-4}	32/32	29/32
Exposed	Random	0.072 [0.057,0.124]	-1.08×10^{-6}	19/32	0/32
Hidden	Ordinary	1.000 [1.000,1.000]	-1.95×10^{-4}	32/32	0/32
Hidden	Protect 32	0.088 [0.066,0.124]	-1.92×10^{-4}	32/32	32/32
Hidden	Protect 128	0.018 [0.012,0.024]	-1.93×10^{-4}	32/32	32/32
Hidden	Ridge 128	0.104 [0.063,0.145]	-1.94×10^{-4}	32/32	32/32
Hidden	Random	0.083 [0.063,0.149]	1.02×10^{-6}	12/32	0/32
Four-latent	Ordinary	1.000 [1.000,1.000]	-4.85×10^{-3}	32/32	0/32
Four-latent	Protect 32	0.474 [0.358,0.592]	-4.82×10^{-3}	32/32	20/32
Four-latent	Protect 128	0.145 [0.111,0.183]	-4.67×10^{-3}	32/32	32/32
Four-latent	Ridge 128	0.315 [0.264,0.389]	-4.90×10^{-3}	32/32	28/32
Four-latent	Random	0.194 [0.171,0.250]	-4.61×10^{-6}	16/32	0/32

Table 12: Frozen predictors of independent-input log RMS change. Entries are mean squared log errors, first averaged over four nonrandom directions per seed. Last column is scale-baseline minus directional error with a paired 97.5% seed-bootstrap interval. All coefficients were frozen on old models.

Condition	Direction	Isotropic	Step	Full scale	Paired advantage
Exposed	0.147	2.376	1.909	0.389	0.242 [0.095,0.385]
Hidden	0.009	2.284	2.405	0.445	0.436 [0.305,0.591]
Four-latent	0.035	1.023	4.244	0.496	0.461 [0.311,0.671]

F.1 Shared Updates on Independent Inputs

We next replace samplewise activation changes with updates of the shared 9,408-parameter token encoder, holding the backbone fixed. Five old exposed checkpoints develop the prediction baselines. After freezing them, we train 32 new models for each of three conditions: exposed branch .32, hidden branch .08, and four-latent nonlinear regression. Geometry fitting, prediction protection, diagnostic inputs and evaluation use separate draws. No evaluation result selects an update or a checkpoint.

We compare ordinary geometry descent, output-Jacobian protection on 32 or 128 inputs, ridge protection on 128 inputs, and a random direction. All updates have the same parameter norm, $.001\|\theta\|$ at primary scale. The primary joint success criterion requires independent-input geometry descent, at least 25% of ordinary descent retained, and at most 50% of ordinary prediction disturbance. Sample size follows a prespecified interval precision target; full protocols are in Appendix F.

The joint success counts are Exposed 29/32, Hidden 32/32, Four-latent 28/32. The primary exposed condition has a 97.5% Wilson interval [0.730,0.972]. The primary majority-evidence gate was met.

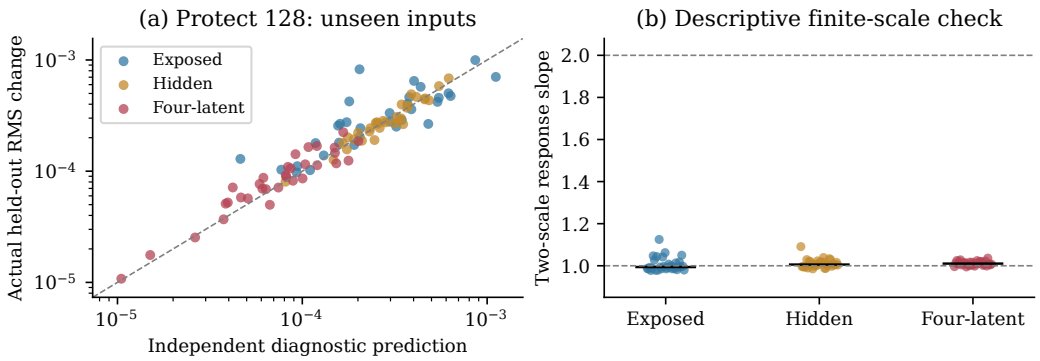

Figure 9: Shared updates protected on 128 inputs. Left: the independent-input directional diagnostic versus observed response. Right: descriptive slopes computed from the two preset scales $.001, .01$. Protection of the sampled Jacobian rows does not give second-order response on unseen inputs. Each point is a newly trained model; the slope summary is not a separate confirmatory endpoint.

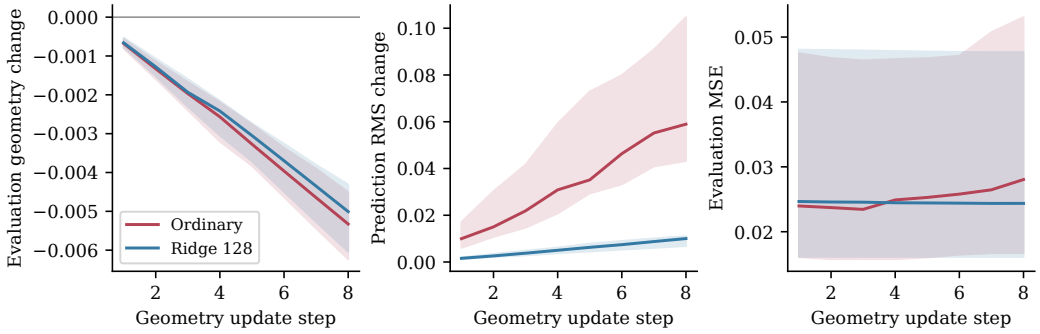

Figure 10: All eight steps of the shared-encoder intervention paths on 32 new exposed-map models. Lines show seed medians and bands the interquartile range. The same independent evaluation inputs are used throughout. These are geometry-only updates with a fixed backbone, not end-to-end training results.

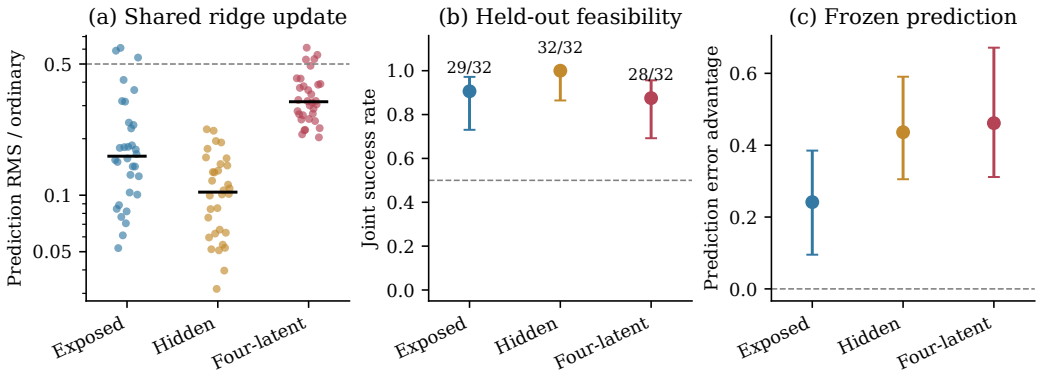

Figure 11: Frozen confirmation on new models, primary scale. (a) Ridge-update prediction disturbance relative to paired ordinary updates; every seed is shown, with median bars. (b) The joint feasibility endpoint, with 97.5% Wilson intervals. (c) Full scale-baseline error minus directional diagnostic error in predicting log RMS change, with 97.5% paired seed-bootstrap intervals. Positive values favor the directional diagnostic. Each condition has 32 seeds.

Median ridge-to-ordinary prediction response ratios were 0.161, 0.104, 0.315 for exposed, hidden and four-latent tasks, respectively. All controls are in Table 11.

The standard 128-input projection meets the joint criterion in 32/32, 32/32, 32/32 seeds across the three conditions; random directions meet it in 0/32, 0/32, 0/32. Joint success requires both reduced prediction disturbance and retained geometry descent.

Despite a near-zero linear response on the 128 protection inputs, the same update has nonzero response on independent inputs. A descriptive comparison of the two preset step sizes gives median response slopes 0.994, 1.006, 1.010 for exposed, hidden and four-latent tasks (Figure 9). The reduction transfers, but first-order invariance does not; the activation experiment's second-order response cannot be assigned to these shared updates.

F.2 Frozen Predictions and Stability

The diagnostic measures the chosen update's directional derivative on 128 independent inputs and predicts the response on 1,024 other inputs. It uses no fitted coefficient. Comparators are isotropic Jacobian scale and two development-fitted log-linear models, including a full scale model with gradient, token, parameter, Jacobian and validation-error scales plus method indicators. The target is prediction disturbance, not whether MSE improves. We average the four nonrandom directions within seed before inference.

For Exposed, full scale-baseline minus directional squared-log error was 0.242 [0.095,0.385]. For Hidden, full scale-baseline minus directional squared-log error was 0.436 [0.305,0.591]. For Four-latent, full scale-baseline minus directional squared-log error was 0.461 [0.311,0.671]. These are paired 97.5% bootstrap intervals across 32 seeds. The prespecified cross-condition prediction gate was met. All baseline scores are in Table 12.

Removing any one seed preserved a positive mean predictive advantage in all three conditions. Seed medians and all leave-one-seed-out ranges are reported in the appendix; repeated directions are not treated as extra seeds.

Eight-step geometry-only paths compare ordinary and ridge updates with fresh fitting batches and recomputed protection Jacobians. All steps are retained (Figure 10). These experiments test shared parameter realizability and transfer of a diagnostic. They do not establish that joint neural training follows the same paths or that protected geometry improves predictive accuracy.

G Protection Coverage and Accessible Update Capacity

Question and fixed criteria. The earlier parameter experiment leaves a large empirical nullspace. We therefore vary the allowed update coordinates and the support of protection inputs while keeping each trained model fixed. This tests whether available parameter directions alone explain independent-input protection. It is not a new projection method or a sweep over trained network widths.

The first hypothesis fixes 512 accessible coordinates and 128 protection inputs. Full-support protection should exceed restricted-support protection in joint success by at least .15, with a positive lower endpoint of a paired 97.5% bootstrap interval. The second hypothesis requires lower Brier error for the calibrated boundary diagnostic than the full scale/cell baseline, with a positive 97.5% interval for their paired error difference. Success and failure must each constitute at least 10% of confirmation cells to avoid a degenerate classification test. These are fixed criteria, not criteria chosen after seeing the confirmation results.

Development information and new conditions. All 96 models from the earlier parameter confirmation study now serve as development models, with fresh post-training inputs. They represent 32 seed blocks across three tasks, not 96 independent seed blocks. Their previous confirmation status does not carry over to this new analysis. Seeds 100–131 supply five deterministic CV folds through $(s - 100) \bmod 5$; all tasks and intervention cells sharing a seed remain in one fold.

New confirmation models use seeds 2000–2031. Even seeds use hidden branch visibility .16; odd seeds use the four-latent generator with target-noise standard deviation .75. Development uses visibility .08 and noise .5 for the corresponding tasks. There are 16 new models per condition; the primary estimate concerns their fixed equal mixture, with 32 independent model blocks. Condition-specific summaries are descriptive. The training architecture and MSE-only protocol are unchanged: 1,024 training and 1,024 validation observations, AdamW with learning rate .001, weight decay .0001, batch size 256, maximum 120 epochs, patience 12 and at least 15 executed epochs. Validation MSE selects the checkpoint.

For development, map seeds and normalization are retained from the saved models. New map seeds are $700000 + s$ and initialization seeds $800000 + s$. Sampling uses $600000 + 10000j + s$, with $j = 0, \ldots, 6$ for training, validation, geometry, full protection, restricted-protection candidates, diagnostic and evaluation data. Only training data determine normalization. Logs, checkpoints and all sampled arrays are retained.

Coverage and coordinates. A permutation with seed $900000 + s$ selects nested sets of 32, 128 or 512 encoder coordinates. The remaining encoder parameters, backbone and head are fixed. This is an intervention on accessible update capacity, not a claim invariant to encoder reparameterization. Full protection uses 128 independent inputs from the evaluation distribution. Restricted protection uses the first 128 qualifying samples in an independent pool of 4,096: $u < -1.5$ in the branch task, or $u_1 < -.75$ in the four-latent task. The pool must contain enough qualifying samples; otherwise the case fails rather than changing the restriction. Both restrictions cover the leftmost eighth of that latent coordinate. Within each protection set, use its first 16 or 128 samples. Evaluation always uses the full input distribution.

On the selected coordinates, ordinary descent uses the gradient of geometry on 64 fitting samples. The protected direction removes its component along the protection Jacobian rows, retaining Gram eigenvalues above 10^{-8} times the largest. If the residual norm is at most 10^{-8} times the unprojected gradient norm, the direction is unavailable; its update is zero and its joint success is false. Otherwise both directions are normalized and have parameter step length $.001\|\theta\|$. Numerical ranks, residuals, coordinate indices and directions are saved.

Boundary measurements. Let v_0 and v denote ordinary and protected unit directions, extended by zeros outside the selected coordinates. The independent diagnostic set contains 128 inputs with labels. It estimates

$$q_{\mathcal{D}} = \frac{\|J_{\mathcal{D}} v\|}{\|J_{\mathcal{D}} v_0\|}, \qquad h_{\mathcal{D}} = \frac{g_{\mathcal{D}}^{\top} v}{g_{\mathcal{D}}^{\top} v_0}, \tag{26}$$

where $g_{\mathcal{D}}$ averages geometry gradients over two batches of 64. The raw linear rule requires both directional geometry derivatives to be negative, $h_{\mathcal{D}} \geq .25$, $q_{\mathcal{D}} \leq .5$, and an available direction. A non-descending ordinary direction fails the rule. These quantities follow from standard first-order calculus. The empirical question is whether they distinguish success and failure better than counts, ranks, coverage labels and scalar scales in new conditions.

In particular, $\dim \ker J_A$ does not determine $v^\top C v$, with $C = \mathbb{E}[J(X)^\top J(X)]$. Protection inputs drawn from a restricted distribution estimate a different covariance. Increasing their count need not cover the missing response directions. This is a motivation for the controlled coverage comparison, not a theorem that more restricted samples necessarily worsen protection.

Competitors and calibration. Rank features are log coordinate count, log protection count and retained rank divided by coordinate count. Cell features are indicators for the 12 capacity/count/coverage combinations, omitting one reference cell. The full scale baseline combines these with log step length, restricted geometry gradient norm, token RMS, diagnostic Jacobian RMS, validation MSE and the ordinary direction's diagnostic response. It therefore already observes coverage, capacity and an ordinary directional scale.

The boundary model adds log $q_{\mathcal{D}}$, clipped $h_{\mathcal{D}}$ and a diagnostic descent indicator. A stronger forward comparator instead uses the actual protected and ordinary prediction changes and geometry differences on the same 128 diagnostic inputs at the prescribed finite step. Both may use diagnostic labels for geometry; neither uses evaluation labels to make predictions. This comparator is retained even if it outperforms the Jacobian diagnostic.

All five calibrated predictors use linear ridge fits to the binary success indicator, with probabilities clipped to $[0, 1]$. Unavailable directions have probability zero for every predictor. Training-fold standardization and SVD remove singular feature directions; no validation-fold values enter these transformations. Five seed-group folds select the penalty from $\{10^{-4}, 10^{-2}, 1, 100\}$ by Brier error. The penalty is scaled by the number of training rows and the intercept is unpenalized. Final coefficients are fit to all development blocks and frozen before confirmation begins. The raw linear and forward threshold rules are additional controls.

Evaluation and uncertainty. Each confirmation model first saves its frozen probabilities and diagnostic measurements. Only then is its 1,024-input evaluation split constructed and used to measure responses. All 12 cells are retained. Joint success uses the same requirements as the earlier study: independent-input geometry descent for both updates, at least 25% of ordinary descent retained, and at most 50% of ordinary RMS prediction change. The number of models is fixed at 32; no significance-based extension is allowed. The corresponding worst-case 97.5% Wilson half-width for a per-model Bernoulli endpoint is about .184, but this does not guarantee precision for the paired continuous endpoints used here.

Prediction errors are averaged over the 12 cells within a model. Paired intervals use 10,000 model-block bootstrap resamples with seed 20260924. The two primary hypotheses use 97.5% intervals; these are approximate bootstrap intervals. Balanced accuracy at probability .5 and confusion counts are descriptive. Predictions, directions and actual network outputs are saved for independent replay.

At 512 accessible coordinates and 128 protection inputs, joint success was 32/32 under full coverage and 26/32 under restricted coverage. The paired success-rate difference was 0.188 [0.062,0.344]. The prespecified coverage criterion was met.

Scale-minus-boundary Brier error was 0.014 [0.004,0.023]. The prespecified predictive criterion was met. There were 158 successful and 226 unsuccessful cells; uncertainty is computed over 32 model blocks, not 384 independent trials.

Interpretation and stability. Full coverage succeeds in all 32 models at the primary capacity/count setting; restricted coverage succeeds in 26. The paired difference is therefore carried by six models, with median zero. The interval excludes zero but does not establish a population effect of at least .15; that margin was a point-estimate requirement. Removing any one model leaves a mean coverage difference between .161 and .194; this is a descriptive sensitivity check, not a new significance test. The mean scale-minus-boundary Brier difference ranges from .012 to .016 under the same leave-one-model-out check.

Table 13: New confirmation: all coverage/capacity cells. Each row contains 32 models, shared across rows. Rank is the median retained numerical rank.

Capacity	Count	Coverage	Rank	Available	Success	RMS ratio
32	16	full	16	32/32	22/32	0.305
32	16	restricted	16	32/32	12/32	0.549
32	128	full	32	0/32	0/32	0.000
32	128	restricted	32	0/32	0/32	0.000
128	16	full	16	32/32	20/32	0.316
128	16	restricted	16	32/32	12/32	0.602
128	128	full	124	32/32	2/32	0.024
128	128	restricted	112	32/32	7/32	0.210
512	16	full	16	32/32	20/32	0.373
512	16	restricted	16	32/32	5/32	0.652
512	128	full	128	32/32	32/32	0.058
512	128	restricted	128	32/32	26/32	0.371

Table 14: Frozen prediction of joint success. Brier intervals are 97.5% model-block bootstrap intervals. BA is descriptive balanced accuracy at probability .5.

Predictor	Brier [interval]	BA	TP	FP	FN
rank	0.168 [0.154,0.184]	0.694	115	77	43
cell	0.149 [0.138,0.161]	0.791	120	40	38
scale	0.120 [0.108,0.136]	0.825	121	26	37
boundary	0.107 [0.095,0.121]	0.859	126	18	32
forward	0.107 [0.095,0.121]	0.861	126	17	32
linear rule	0.052 [0.021,0.089]	0.948	150	12	8
forward rule	0.052 [0.021,0.089]	0.948	150	12	8

Final calibration retains ranks 3, 11, 18, 21 and 21 from 3, 11, 20, 23 and 23 input features for rank, cell, scale, boundary and forward models, respectively. Selected penalties are 10^{-4} except for cell identity (10^{-2}). Boundary-minus-forward Brier error is $-.00034$ $[-.00136, .00009]$, giving no resolved advantage between those two diagnostics. Both raw rules make 20 errors (12 false positives and eight false negatives) and have Brier error $.052$ and balanced accuracy $.948$. Calibration is therefore not an improvement here. Hidden-branch cells have 104/192 successes and four-latent cells 54/192; these are repeated cells within 16 models per condition, not independent sample counts.

The empirical support is limited to this fixed mixture, small finite steps and coordinate intervention. It does not establish a universal coverage threshold, a width scaling law or better jointly trained regression risk.

Evidence chronology. The 390 training fits and 40-checkpoint activation analysis are earlier mechanism evidence. The subsequent 96-model parameter study was a separate confirmation for its original questions; all 96 models become development data for the present coverage study. Only seeds 2000–2031 constitute its 32-model confirmation. The analyses below were added after inspecting that confirmation and do not create a further confirmation set.

Post-hoc inference sensitivity. At the primary cell, six model pairs favor full protection and none favor restricted protection. Under the conditional paired null with equiprobable discordant directions, the exact two-sided binomial test gives $2(1/2)^6 = .03125$. This exceeds .025, the level corresponding to a 97.5% two-sided interval. The one-sided value is $.015625$; it does not replace the frozen bootstrap criterion. The bootstrap gate was met, but inference is sensitive to discreteness and method choice. No extra models were added after this check.

Keeping 16 models from each condition in every bootstrap resample gives 97.5% intervals $[.0625, .34375]$ for the coverage difference and $[.00402, .02284]$ for scale-minus-directional Brier error. The condition means of the latter are $.01181$ and $.01560$, with positive differences in 10/16 hidden-branch and 13/16 four-latent models. These descriptive checks do not establish a separate significant effect in each condition.

Exploratory failure analysis. Table 15 decomposes the existing outcomes using the original joint criterion. An unavailable direction is counted first; otherwise ordinary descent is required. Among those eligible cells, the geometry and output-response failures can overlap. Their union, together with the first two exclusions, reproduces every recorded failure.

All 64 cells with 32 coordinates and 128 protection inputs have unavailable directions. At 128 coordinates and 128 inputs, geometry-retention failure occurs in 30 full-coverage and 24 restricted-coverage models. Restricted coverage also has three response failures, two overlapping geometry failure. Five models succeed only under restricted coverage, and none only under full coverage. At 512 coordinates and 128 inputs, geometry retention passes in every model; the six restricted-coverage failures all concern response. Thus the observed reversal changes which criterion limits joint success. It does not prove that retained numerical rank alone causes the reversal.

Table 15: Exploratory failure decomposition on the existing confirmation set. Counts are out of 32 models per row. U: unavailable direction; O: ordinary update does not descend; G: protected geometry retains less than 25% of ordinary descent; R: output RMS exceeds 50% of ordinary response. G and R are evaluated only for available directions with ordinary descent and can overlap.

Coordinates	Count	Coverage	U	O	G	R	G∩R	Success
32	16	full	0	0	3	7	0	22
32	16	restricted	0	0	3	18	1	12
32	128	full	32	0	0	0	0	0
32	128	restricted	32	0	0	0	0	0
128	16	full	0	0	0	12	0	20
128	16	restricted	0	0	0	20	0	12
128	128	full	0	0	30	0	0	2
128	128	restricted	0	0	24	3	2	7
512	16	full	0	0	0	12	0	20
512	16	restricted	0	0	0	27	0	5
512	128	full	0	0	0	0	0	32
512	128	restricted	0	0	0	6	0	26

Exploratory diagnostic components on development data. Keeping the existing thresholds and making unavailable directions fail, the response-only rule makes 188/1,152 development errors, the geometry-only rule makes 373, and their conjunction makes 112. The latter requires negative ordinary and protected geometry derivatives and retention at least .25; the response rule requires a ratio at most .5. These counts use the original 32 development seed blocks and are not independent trials or new confirmation results. No coefficients were refit. Both diagnostic components matter for these fixed rules; this does not isolate their contribution to the fitted ridge model.

On confirmation data, the raw linear and forward rules agree on all 384 classifications. Their inputs estimate the joint endpoint on a separate sample. This is diagnostic-to-evaluation transfer, not a claim that counts and coverage alone predict the endpoint or that calibration is beneficial.

Unresolved boundary. The intervention grid is shared by development and confirmation. A law locating the geometry/response crossover at unseen coordinate counts or coverage fractions remains untested. The fixed step uses the full encoder norm; it is not a constant fraction of each selected subvector's norm. Likewise, the Gram cutoff defines a numerical projection rather than an exact kernel at every cell; the post-hoc audit below checks three cutoffs. The fixed sample count avoids optional stopping, but its Bernoulli precision calculation does not guarantee the precision of the paired primary endpoints.

G.1 Gradient Retention and Numerical Projection

Failure labels alone do not explain the tradeoff. We therefore replay the 32 saved checkpoints and measure how much fitting geometry gradient survives the projection. All analyses in this subsection were specified after seeing the coverage results. They use existing inputs and are exploratory.

Fitting loss versus gradient transfer. Restrict gradients to the selected coordinates. Let g be the fitting gradient, P an orthogonal projection onto the retained update subspace, $G = \|g\| > 0$, and

assume $Pg \neq 0$. With $v_0 = -g/G$ and $v = -Pg/\|Pg\|$, define

$$\rho = \frac{\|Pg\|}{\|g\|} = \frac{g^\top v}{g^\top v_0}. \tag{27}$$

Thus equal-length updates retain fraction ρ of fitting geometry descent. Rank alone does not determine this fraction: the orientation of g relative to the retained subspace matters. For numerical replay we measure $-g^\top v/G$ directly from the saved direction.

Let the independent diagnostic gradient be $g_{\mathcal{D}} = g + e$ and $\epsilon = \|e\|/G$. When the denominator is negative,

$$h_{\mathcal{D}} = \frac{-G\rho + e^\top v}{-G + e^\top v_0}. \tag{28}$$

This separates loss of fitting descent from transfer of the gradient. For $\epsilon < 1$, ordinary diagnostic descent is guaranteed. For any $\tau > 0$, Cauchy–Schwarz gives sufficient conditions

$$\rho - \epsilon \geq \tau(1+\epsilon) \quad \Longrightarrow \quad h_{\mathcal{D}} \geq \tau, \tag{29}$$

$$\rho + \epsilon < \tau(1-\epsilon) \quad \Longrightarrow \quad h_{\mathcal{D}} < \tau. \tag{30}$$

For the first implication, $-g_{\mathcal{D}}^\top v \geq G(\rho - \epsilon) > 0$ and $-g_{\mathcal{D}}^\top v_0 \leq G(1+\epsilon)$. For the second, use the upper numerator bound $G(\rho+\epsilon)$ and positive denominator lower bound $G(1-\epsilon)$. These are elementary deterministic bounds, not a new generalization theorem. They concern diagnostic derivatives, not finite-step evaluation loss or output protection.

Checkpoint replay. We restore each saved model, its original 64 fitting inputs, 128 diagnostic inputs and all 12 saved directions. Diagnostic gradients average two batches of 64, matching the frozen experiment. Recomputed diagnostic retention differs from the saved values by at most 2.23×10^{-16}. No evaluation loss or success label enters ρ or ϵ.

With 128 protection inputs and 128 coordinates, median ρ is .068 under full coverage and .182 under restricted coverage. The fitting retention threshold .25 is met by 0/32 and 8/32 models, respectively. At 512 coordinates the medians are .586 and .643, and all 32 models in each coverage group pass that fitting threshold. These measurements support a projection-overlap explanation for the observed small-coordinate geometry failures. They do not establish a universal crossover.

Table 16: Exploratory checkpoint audit. Medians of fitting retention ρ and diagnostic retention $h_{\mathcal{D}}$; Fit and Eval count geometry-retention passes out of 32. Unavailable directions fail both counts.

Coordinates	Count	Coverage	ρ	$h_{\mathcal{D}}$	Fit	Eval
32	16	full	0.527	0.529	31	29
32	16	restricted	0.535	0.592	31	29
32	128	full	0.000	0.000	0	0
32	128	restricted	0.000	0.000	0	0
128	16	full	0.792	0.858	32	32
128	16	restricted	0.820	0.821	32	32
128	128	full	0.068	0.058	0	2
128	128	restricted	0.182	0.146	8	8
512	16	full	0.917	0.930	32	32
512	16	restricted	0.932	0.935	32	32
512	128	full	0.586	0.644	32	32
512	128	restricted	0.643	0.703	32	32

Median ϵ is .812, .829 and .849 at 32, 128 and 512 coordinates. With $\tau = .25$, the sufficient bounds certify only four diagnostic passes and no diagnostic failures across 384 cells (320 have an available direction). They are mostly uninformative here; they cannot support a certified population boundary.

Diagnostic label requirements. A geometry gate based on fitting retention $\rho \geq .25$, combined with $q_{\mathcal{D}} \leq .5$ and availability, uses diagnostic inputs but no diagnostic labels. The original gate uses both diagnostic geometry signs and $h_{\mathcal{D}}$. Both require the same trained model, 64 labeled fitting inputs and protection inputs. The change removes the requirement for 128 diagnostic labels shared

Table 17: Exploratory screen components on the same 32 models and 384 cells. All rules require an available direction. Labels are diagnostic labels per model, shared across 12 cells; training and fitting labels are unchanged. FP denotes predicting success when the update fails.

Rule	Labels	Errors	FP	FN
Availability	0	162	162	0
Output only	0	68	63	5
Fitting only	0	118	111	7
Fitting and output	0	25	13	12
Fitting and forward output	0	25	13	12
Labeled diagnostic	128	20	12	8

across a model's 12 cells, not 128 labels per cell. It does not make training or update construction label-free. Evaluation labels remain necessary to assess either rule.

The screen without diagnostic labels makes 25/384 errors, versus 20/384 for the labeled diagnostic rule. The paired error-rate difference is 0.0130, with a descriptive 97.5% stratified model-block bootstrap interval $[0.0000, 0.0286]$. It introduces 6 errors and corrects 1; false-positive/false-negative counts are 13/12 versus 12/8.

Table 17 separates the two gates using the same availability condition and thresholds. Output response alone makes 68 errors; fitting retention alone makes 118; together they make 25. Thus fitting retention adds information beyond output response on these records. Replacing the output ratio by actual finite-step prediction changes gives the same error count and confusion counts, using no diagnostic labels. This does not show an advantage of the Jacobian check. The comparison does not include a finite-step geometry check on fitting inputs.

The aggregate difference is not uniform across cells. At 128 coordinates and 128 protection inputs with restricted coverage, errors increase from 1/32 to 5/32. Four of the six new errors occur there; all seven changed correctness outcomes involve five models. These are descriptive subgroups, not independently tested effects. All component checks reuse the original models and evaluation outcomes and were specified after confirmation.

The possible use is screening updates when fitting labels are available but a new diagnostic batch is unlabeled. Whether this is useful depends on the cost of additional labels and the consequences of accepting a failed update or rejecting a successful one. Neither was measured here. The label comparison therefore remains an exploratory limitation check, not a demonstrated deployment benefit.

The interval uses 10,000 resamples, preserving 16 model blocks from each condition, with seed 20260924. The rule and this comparison were examined after the original confirmation; no non-inferiority margin was frozen. The result is therefore a descriptive label-requirement tradeoff, not evidence of equivalent accuracy. The original combined screen retains independent output Jacobian measurements; the forward substitute uses prediction changes. Neither comparison measures runtime or total annotation cost. The screen is a direct use of the existing projection diagnostic, not a new learning method. A prospective comparison would need an application-motivated tolerance fixed before collecting new outcomes.

Numerical cutoff sensitivity. We repeat the 128-protection-input cells at 128 and 512 coordinates using relative Gram eigenvalue cutoffs $10^{-10}, 10^{-8}, 10^{-6}$. Checkpoints, coordinate sets, fitting/protection/evaluation inputs, gradient and step norms are fixed. The middle cutoff replays 128 original interventions; the other two give 256 additional interventions on the same models and evaluation inputs, not independent confirmations.

The opposite coverage orderings at 128 and 512 coordinates persist at all three tested cutoffs. This checks numerical sensitivity on existing models, not population robustness or a new confirmation. At 128 coordinates, full-coverage success counts are 0, 2 and 14 as the cutoff increases; restricted-coverage counts are 3, 7 and 25. At 512 coordinates the corresponding counts are 32 at every cutoff and 26, 26, 24. The ordering is less sensitive than the absolute success counts.

Table 18: Exploratory cutoff sensitivity with 128 protection inputs. Rank and ρ are medians; available, geometry-pass and joint-success counts are out of the same 32 models.

Cutoff	Coordinates	Coverage	Rank	ρ	Available	Geometry	Success
10^{-10}	128	full	127.0	0.012	20	0	0
10^{-10}	128	restricted	126.0	0.055	31	3	3
10^{-10}	512	full	128.0	0.586	32	32	32
10^{-10}	512	restricted	128.0	0.637	32	32	26
10^{-8}	128	full	124.0	0.068	32	2	2
10^{-8}	128	restricted	112.5	0.182	32	8	7
10^{-8}	512	full	128.0	0.586	32	32	32
10^{-8}	512	restricted	128.0	0.643	32	32	26
10^{-6}	128	full	101.0	0.228	32	14	14
10^{-6}	128	restricted	71.0	0.410	32	31	25
10^{-6}	512	full	124.0	0.599	32	32	32
10^{-6}	512	restricted	88.5	0.711	32	32	24

The combined audit separates a measured fitting-gradient overlap from success labels and checks a concrete numerical alternative explanation. It still does not identify a crossover at unseen capacities, compare other auxiliary losses, or establish a reduction in regression risk.

H EARLIER SHARED-MAP CONTROLS

This appendix preserves the earlier controlled study that motivated the experiments in the main text. It uses its original seeds, configurations and saved results. These runs are not additional replications of the new observation-map and loss-weight study. Tables report test MSE in training-standardized target units; uncertainty is the sample standard deviation across seeds unless an interval is explicitly stated.

H.1 GENERATOR AND COMPARISONS

Let $u \sim \mathrm{Unif}(-2, 2)$, $a \sim \mathrm{Bernoulli}(q)$, and b be uniform on $\{-1, +1\}$, independently. The effective branch is $c = ab$. The branch tasks use

$$f_0(u) = 1.2u + 0.5\sin(3u), \qquad Y = f_0(u) + Gc + \epsilon_y, \tag{31}$$

$$X = \phi_W(u) + \kappa c\mathbf{e}_{16} + \epsilon_x, \qquad \phi_W(u) = [v(u), v(u)W], \tag{32}$$

where $v(u) = [u, u^2, \sin(2u), \cos(3u)]$ and $W \in \mathbb{R}^{4\times 12}$ has independent $\mathcal{N}(0, 1/4)$ entries. The matrix is drawn once per configuration and shared across its splits. Independent Gaussian noises have $\sigma_x = \sigma_y = 0.05$; the other defaults are $q = 0.4$ and $G = 3$. The normal control sets $c = 0$. Partial and absent branch signal use $\kappa = 0.08$ and 0, respectively. The configuration name `resolvable` denotes the partial case; overlapping Gaussian observations do not permit exact branch recovery.

The earlier *IID-noise* task retains the predictable base function: $Y = f_0(u) + \eta + \epsilon_y$, where $\eta \sim \mathcal{N}(0, qG^2)$ is independent and the input has no branch signal. It differs from the *pure-noise diagnostic*, which uses $Y \sim \mathcal{N}(0, qG^2 + \sigma_y^2)$ independently of the nuisance input. Only the latter has a constant conditional mean.

The completed shared-map evaluation uses 5 independent seeds per main method–setting cell. Each neural run uses 4096 training, 1024 validation, and 4096 test samples with input dimension 16.

The constant baseline predicts the training target mean. Ridge regression uses all input features with its penalty selected by validation MSE; the numerical Bayes reference uses the known generator and is not a learned competitor. The token-without-geometry, direct backbone geometry, and ordinary auxiliary MLP comparisons separate pathway and objective choices. The RNC comparison uses the same Transformer with a single-view joint MSE plus weighted Rank-N-Contrast objective; it is an adaptation, not a reproduction of the original two-stage training protocol.

Training statistics normalize each split. In matched comparisons, methods share the data and feature map. Table 19 reports the main results, and Table 20 separates token, auxiliary pathway and objective choices. The strong MLP is competitive with ALDT; the token without geometry and the ordinary auxiliary pathway have slightly lower mean error than ALDT in the partial-signal setting. These differences do not establish a general architecture advantage or a benefit of the raw token objective.

Table 19: Earlier study: shared-map test MSE, mean and seed SD over five paired repetitions. The numerical Bayes predictor is a generator-aware reference. IID noise retains the base function $f_0(u)$.

Model	Normal	Partial signal	Absent signal	IID noise
Constant mean	0.990 ± 0.011	0.999 ± 0.031	0.999 ± 0.031	0.994 ± 0.019
Ridge	0.031 ± 0.00059	0.393 ± 0.032	0.694 ± 0.014	0.694 ± 0.030
Weak MLP	0.00220 ± 0.000093	0.382 ± 0.029	0.685 ± 0.016	0.685 ± 0.029
Strong MLP	0.00185 ± 0.000049	0.380 ± 0.028	0.686 ± 0.015	0.685 ± 0.031
Transformer	0.00186 ± 0.000064	0.413 ± 0.062	0.690 ± 0.013	0.688 ± 0.030
ALDT	0.00189 ± 0.00006	0.381 ± 0.022	0.689 ± 0.013	0.688 ± 0.030
Numerical Bayes	0.00162 ± 0.000054	0.347 ± 0.010	0.683 ± 0.015	0.684 ± 0.030

Mean $\pm$ seed SD; independent seeds per cell: 5.

Table 20: Earlier study: token, auxiliary-pathway and objective controls. Transformer geometry acts on CLS; ALDT geometry acts on its raw input token. RNC is a single-view joint-objective adaptation. Entries are mean and seed SD.

Model	Partial signal	Absent signal
Transformer	0.413 ± 0.062	0.690 ± 0.013
Transformer + geometry	0.404 ± 0.023	0.687 ± 0.015
Token, no geometry	0.379 ± 0.020	0.689 ± 0.013
ALDT	0.381 ± 0.022	0.689 ± 0.013
Token + CLS geometry	0.380 ± 0.019	0.688 ± 0.012
Auxiliary MLP	0.379 ± 0.022	0.689 ± 0.015
Auxiliary + CLS geometry	0.381 ± 0.022	0.689 ± 0.014
Auxiliary + raw geometry	0.381 ± 0.021	0.689 ± 0.015
Transformer + RNC	0.400 ± 0.023	0.689 ± 0.013

Mean $\pm$ seed SD; independent seeds per cell: 5.

Table 21: Earlier study: evaluated architectures and registered parameter counts. Geometry objectives add no trainable parameters.

Model	Architecture	Registered parameters
Weak MLP	GELU MLP, 1 hidden layer of width 32	577
Strong MLP	GELU MLP, 4 hidden layers of width 128	51,841
Transformer	3 layers, 4 heads, width 128	599,425
ALDT	Transformer + input token MLP $16 \to 64 \to 128$	608,961
Auxiliary MLP	Transformer + output-side MLP $16 \to 64 \to 128$	609,217

H.2 Architecture and Training

Neural models use AdamW with learning rate 0.001, weight decay 0.0001, batch size 256, and at most 120 epochs. The Transformer uses 3 layers, 4 attention heads, and hidden width 128. Checkpoint selection uses validation MSE; early-stopping patience is 12 epochs after a minimum of 15 epochs. Geometry-on configurations use weight 0.05 and separation margin 1.0; geometry-off controls use weight zero. The joint MSE–RNC configurations use RNC weight 0.05 and temperature 2.0. No test outcomes select the checkpoint. Raw artifacts retain source hashes, configuration, training logs, and selected epoch.

The baseline Transformer tokenizes the scalar features and reads its CLS output using LayerNorm and a scalar linear head. ALDT adds one input token from an MLP of widths $16 \to 64 \to 128$. Its default geometry loss acts on that raw token. The factorial controls instead regularize CLS in both architectures. The auxiliary MLP supplies an output-side input pathway of widths $16 \to 64 \to 128$. Table 21 gives the actual parameter counts; the auxiliary control is not exactly parameter matched.

H.3 Paired Evidence and Feature-Map Shift

The replication unit is a seed, which changes both data sampling and the shared feature map. For methods A, B and setting r, write $d_{s,r} = \mathrm{MSE}_{A,s,r} - \mathrm{MSE}_{B,s,r}$. Negative differences favor A. The method-by-setting interaction is

$$\delta_s = d_{s,\mathrm{partial}} - d_{s,\mathrm{absent}}. \tag{33}$$

Seed bootstrap intervals resample complete paired differences. Models and settings sharing a seed are not additional independent replications. Statistical comparisons pair identical seeds; method-by-setting contrasts subtract paired method differences before inference. Intervals resample seeds and are exploratory at this sample size; nonsignificance is not evidence of equivalence.

Sensitivity to individual seeds. For partial signal, ALDT minus Transformer has mean -0.03204 and median -0.01747. All five differences are negative, but one is -0.10352 and accounts for about 64.6% of their total magnitude. The two-sided exact signed-rank sign-randomization value is $p = 0.0625$. ALDT minus the token without geometry has mean 0.002851, median 0.002830, and $p = 0.125$. These tests assume independent, symmetrically distributed seed differences under the null. Five nonzero differences have a minimum attainable two-sided value of $2/2^5 = 0.0625$.

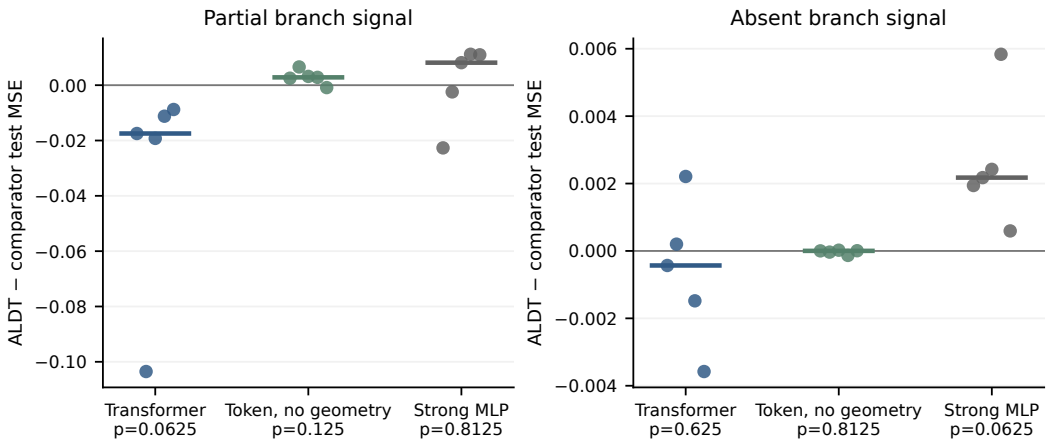

Figure 12: Earlier study: all five paired test-MSE differences for ALDT minus each comparator; bars show medians. Negative values favor ALDT. The exact paired tests describe these earlier runs, not the new direct-excess results.

Table 22: Earlier study: paired seed differences and exploratory bootstrap intervals. Differences are first model minus second model. The text and seed plot also report exact tests and sensitivity to individual seeds.

Comparison	Setting	Mean diff.	95% bootstrap CI	Seeds
ALDT - Transformer	Absent signal	-0.00062	[-0.00219, 0.00105]	5
ALDT - Transformer	Partial signal	-0.032	[-0.068, -0.011]	5
ALDT - Token, no geometry	Absent signal	-0.000027	[-0.000078, 0.000012]	5
ALDT - Token, no geometry	Partial signal	0.00285	[0.00068, 0.00509]	5
ALDT - Strong MLP	Absent signal	0.00259	[0.00123, 0.00428]	5
ALDT - Strong MLP	Partial signal	0.00104	[-0.012, 0.010]	5

A bootstrap interval excluding zero is therefore reported alongside, rather than substituted for, the individual differences and exact test. All comparisons remain exploratory.

Factorial comparison at a common loss location. Define Δ_{geo} within each architecture as MSE with CLS geometry minus MSE without it. The token-by-geometry contrast is

$$\eta_s = \Delta_{\text{geo,token,s}} - \Delta_{\text{geo,plain,s}}. \tag{34}$$

Its two terms are effects of *adding geometry*, not token main effects. Negative interaction means a more favorable change with the token. Table 23 shows intervals including zero; it provides no clear evidence of synergy. The default raw-token objective is evaluated separately.

Feature-map shift. The shift control changes only the test feature map while reusing the checkpoint, training normalization and test sample seed. In Table 24, the strong MLP's error rises sharply despite its competitive shared-map error. ALDT and the plain Transformer are less sensitive in these runs, but both exceed the constant baseline for shifted partial signal. This comparison concerns a change in the input mapping, not failure under the shared-map distribution.

Finite-test differences in Table 25 can be negative. A sampled mean conditional variance, by contrast, estimates Bayes risk without realized test-label noise. Neither is a pointwise lower-bound guarantee for empirical test MSE.

H.4 Local Disagreement and Shared-Label Dependence

For the $k = 10$ nearest neighbors excluding the query in the same split, Local Input–Target Disagreement is

$$LITD_i = \frac{\sum_{j\in\mathcal{N}_k(X_i)} w_{ij}|Y_i - Y_j|}{\sum_{j\in\mathcal{N}_k(X_i)} w_{ij}}, \qquad w_{ij} = \exp\left(-\frac{\|X_i - X_j\|^2}{2\tau^2}\right), \tag{35}$$

Table 23: Earlier study: within-architecture changes from adding CLS geometry and their interaction. All changes are geometry-on minus off.

Setting	n	$\Delta_{\mathrm{geo,token}}$	$\Delta_{\mathrm{geo,plain}}$	Interaction	95% bootstrap CI
Absent signal	5	-0.00111	-0.00204	0.00093	[-0.00268, 0.00527]
Partial signal	5	0.00173	-0.00980	0.012	[-0.012, 0.044]

Table 24: Earlier study: test feature-map shift using the same fitted checkpoints and training normalization. Entries are mean and seed SD.

	Partial signal		Absent signal	
Model	Shared map	Shifted map	Shared map	Shifted map
Constant mean	0.999 ± 0.031	0.999 ± 0.031	0.999 ± 0.031	0.999 ± 0.031
Ridge	0.393 ± 0.032	49.064 ± 21.949	0.694 ± 0.014	1.247 ± 0.553
Strong MLP	0.380 ± 0.028	37.383 ± 17.838	0.686 ± 0.015	0.954 ± 0.128
Transformer	0.413 ± 0.062	3.166 ± 2.448	0.690 ± 0.013	0.906 ± 0.168
ALDT	0.381 ± 0.022	2.551 ± 0.158	0.689 ± 0.013	0.920 ± 0.139

where τ is the median collected neighbor distance. The diagnostic uses training-normalized inputs and targets. Test labels enter only this post-evaluation analysis, not training or model selection.

In raw units, let $Y = \epsilon \sim \mathcal{N}(0, \sigma^2)$ independently of X. Bayes squared error is ϵ^2. Conditional on a query target e,

$$\mathbb{E}|e - \epsilon'| = \sigma\sqrt{\frac{2}{\pi}}e^{-e^2/(2\sigma^2)} + |e|\{2\Phi(|e|/\sigma) - 1\}, \tag{36}$$

which increases with $|e|$. Shared target noise can therefore associate LITD with squared error even for an optimal predictor without branches. Training normalization rescales these quantities and translates the Bayes mean; the corresponding shared-label dependence remains. For an input-only comparison, collect the k nearest *training* neighbors of the test query and use

$$D_{\mathrm{ref}}(x) = \frac{1}{k(k-1)}\sum_{j \neq \ell}|y_j - y_\ell|. \tag{37}$$

This score excludes the query target and has a different estimand from LITD. Its comparison with LITD measures dependence on the query label, not whether the two scores estimate exactly the same quantity.

In a branch-free pure-noise control, the correlation between LITD and the Bayes predictor's squared error is 0.853 ± 0.009, whereas a local reference score that excludes the query target has correlation -0.005 ± 0.003 (mean and seed SD over 5 seeds). This control demonstrates that shared-label coupling can produce a strong error correlation without branch structure; correlation alone cannot validate a branch-sensitive representation.

The earlier diagnostic uses five seeds with 4096 training and 4096 test samples per seed, shared maps and training-only normalization. Correlations in Table 26 use per-sample squared error of the known-generator Bayes predictor. Numerical conditional moments use a 2049-point Simpson grid; 4097-point refinement on 128 inputs per seed and visibility gives a maximum standardized mean difference of 6.3×10^{-6}. This sampled check is not a uniform error bound. Numerical conditional-variance risk decreases across the evaluated branch-visibility settings: the mean standardized risk is 0.687 at $\kappa = 0$, 0.347 at $\kappa = 0.08$, and 0.0024 at $\kappa = 0.32$. These generator-aware calculations distinguish partial information from exact branch recovery; they are oracle diagnostics, not trained-model results.

The pair-label check uses one seeded partition of training samples into batches of 256. It recomputes nearest input neighbors and the target-gap median within each batch. It is not a replay of optimizer batches. High agreement with effective-branch differences at absent signal only shows that the *targets* provide branch-related pair supervision; it does not establish that an input-only encoder can recover the branch.

Table 25: Earlier study: model test MSE minus numerical Bayes test MSE on the same samples. Mean and seed SD over five repetitions. This noisy difference estimates excess risk; it is not the direct oracle excess used in the new study or a difference from an analytic lower bound.

Model	Normal	Partial signal	Absent signal	IID noise
ALDT	0.00028 ± 0.000036	0.034 ± 0.013	0.00631 ± 0.00285	0.00391 ± 0.00198
Strong MLP	0.00023 ± 0.00004	0.033 ± 0.019	0.00372 ± 0.00192	0.00150 ± 0.00145
Transformer	0.00025 ± 0.000077	0.066 ± 0.053	0.00693 ± 0.00370	0.00449 ± 0.00215

Mean $\pm$ seed SD; independent seeds per cell: 5.
Model MSE minus numerical Bayes MSE on the same test samples.

Table 26: Earlier diagnostic: correlation with per-sample Bayes squared error, mean and seed SD. Pure noise has no predictable base function.

Setting	LITD–error correlation	Reference–error correlation
Pure noise	0.853 ± 0.009	-0.005 ± 0.003
Normal	0.809 ± 0.015	0.020 ± 0.015
Partial signal	0.523 ± 0.031	0.004 ± 0.019
Absent signal	0.862 ± 0.006	0.005 ± 0.023

H.5 Generator-Specific Lower Bound and Numerical Oracle

The squared-loss decomposition gives $R(f) \geq \mathbb{E}[\mathrm{Var}(Y \mid X)]$, with equality for $f(X) = \mathbb{E}[Y \mid X]$. At absent branch signal, $c = ab$ is independent of $X = \phi_W(u) + \epsilon_x$, has mean zero and second moment q. The branch term and target noise are conditionally uncorrelated with $f_0(u)$. Hence

$$\mathrm{Var}(Y \mid X) = \mathrm{Var}(f_0(u) \mid X) + qG^2 + \sigma_y^2, \tag{38}$$

$$R^* \geq qG^2 + \sigma_y^2 = 0.4 \times 3^2 + 0.05^2 = 3.6025. \tag{39}$$

Direct integration over $u \sim \mathrm{Unif}(-2, 2)$ gives $\mathrm{Var}(f_0(u)) \simeq 1.6479$, so $\mathrm{Var}(Y) \simeq 5.2504$. Dividing by this population variance gives the standardized lower bound $3.6025/5.2504 \simeq 0.686$. It omits the generally positive term $\mathbb{E}[\mathrm{Var}(f_0(u) \mid X)]$ and is not necessarily the exact Bayes risk. With training standard deviation $s_{y,\mathrm{tr}}$, the corresponding run-specific bound is $3.6025/s_{y,\mathrm{tr}}^2$. No optimization or attainability claim follows from this risk identity.

For partial signal, the known map and branch prior π_c give

$$p(u, c \mid x, W) \propto \pi_c \mathbf{1}[-2 \leq u \leq 2] \exp\left[-\frac{\|x - \phi_W(u) - \kappa c \mathbf{e}_{16}\|^2}{2\sigma_x^2}\right]. \tag{40}$$

Normalize over $c \in \{-1, 0, 1\}$ and $u \in [-2, 2]$, and put $h(u, c) = f_0(u) + Gc$. Then

$$\mu(x) = \mathbb{E}[h \mid x, W], \qquad v(x) = \mathbb{E}[h^2 \mid x, W] - \mu(x)^2 + \sigma_y^2. \tag{41}$$

The implementation sums the branch states and integrates u using Simpson weights. The test average of $v(x)/s_{y,\mathrm{tr}}^2$ estimates standardized Bayes risk. Normal uses $c = 0$; IID noise keeps $f_0(u)$ and adds qG^2 to the conditional target-noise variance. Pure noise has raw moments $\mu(x) = 0$ and $v(x) = qG^2 + \sigma_y^2$. The standardized mean in that control is $-\bar{y}_{\mathrm{tr}}/s_{y,\mathrm{tr}}$, not exactly zero.

Table 27: Earlier pair-label diagnostic. Balanced accuracy compares the target-gap rule with unequal effective branches. Push precision is the fraction of push pairs with unequal branches; true difference is their frequency among all pairs. Balanced accuracy is undefined with only one branch class. Entries are mean and seed SD.

Setting	Bal. acc.	Push prec.	Push frac.	True diff.
Partial signal	0.964 ± 0.00603	0.982 ± 0.011	0.499 ± 0.00032	0.519 ± 0.00997
Absent signal	0.944 ± 0.00414	0.996 ± 0.00330	0.499 ± 0.00041	0.557 ± 0.00466
Normal	N/A	0.00000 ± 0.00000	0.499 ± 0.00027	0.00000 ± 0.00000
Pure noise	N/A	0.00000 ± 0.00000	0.499 ± 0.00028	0.00000 ± 0.00000

Mean $\pm$ seed SD, $n = 5$; one fixed training partition, batches of 256.
Batchwise nearest-input pairs and target-gap median; not an optimizer replay.

I WINE QUALITY EXPERIMENTS

Data and splits. The UCI Wine Quality data contain 1,599 red and 4,898 white wine records, each with 11 physicochemical features and a sensory score (Cortez et al., 2009). The source documentation describes the score as the median of at least three expert evaluations. Individual ratings are not released. We therefore measure prediction error on observed scores, not Bayes excess risk or conditional multimodality.

Identical input vectors are assigned to the same group, regardless of their scores. For seeds 3100–3107, we shuffle the groups and allocate 60%, 20% and 20% to training, validation and testing. All rows remain in the data. Feature and target standardization use training observations only. Model initialization, splits and batch orders are paired across loss settings within each architecture. The eight splits reuse observations; they are repetitions on fixed datasets, not eight independent populations.

Models and losses. The MLP has two hidden layers of width 64 with GELU activations and a linear output. The Transformer uses scalar feature tokens of width 32, a CLS token, two pre-normalized layers with four attention heads and feedforward width 128, followed by LayerNorm and a linear scalar head. It has no additional input token. Both models use zero dropout. Representation geometry acts on the last hidden layer or CLS vector, with distance divided by the square root of its dimension. Prediction geometry acts on the standardized scalar output. This fixed distance convention does not equate the effective strengths of the two penalties. The local pair rule, target-gap median and unit margin are otherwise unchanged. Each architecture is trained with no geometry and with each location at weights .05 and .5.

Training and evaluation. We use AdamW with learning rate .001, weight decay .0001 and batch size 256 for at most 80 epochs. Validation MSE selects the checkpoint; training stops after 12 epochs without improvement, subject to a minimum of 20 epochs. The grid contains 160 fits. Its configuration and source hashes were saved before training, and test evaluation starts only after all fits finish. No model is selected using test error.

The primary comparison is prediction minus representation test MSE at weight .5, separately for each dataset and architecture. We report test MSE in the original score units and MAE as a secondary measure. The smaller weight and comparisons to zero geometry are sensitivity checks. For each loss location we also select among weights $0, .05, .5$ by validation MSE, resolving exact ties in favor of the smaller weight. The zero-weight model is shared between the two selections.

Intervals use 10,000 paired seed-bootstrap resamples with seed 20260924. They describe variation across the fixed repeated-split protocol; they are not population guarantees or multiplicity-adjusted tests. The planned eight repeats are a fixed resource budget, not a powered equivalence design. A difference of 2% of baseline MSE is a descriptive reference for effect size, not an established application tolerance. We retain all weights and splits rather than selecting conditions with larger effects.

Directions across repeated splits. At weight .5, prediction MSE exceeds representation MSE in 2/8, 3/8, 3/8, 4/8 repetitions for red MLP, red Transformer, white MLP and white Transformer, respectively. The corresponding median paired differences are -0.0009, -0.0006, -0.0023, $+0.0002$, and mean differences relative to baseline MSE are $+0.07\%$, -1.19%, -1.27%, -0.43%. Means therefore do not imply consistent rankings across splits. 12 runs reach the 80-epoch limit; 3 select the final epoch. This is a fixed-budget comparison, not a claim that every setting converges.

Interpretation. This experiment tests loss placement in real regression with two standard architectures. It does not identify the conditional noise distribution, test the constant-mean premise of Proposition 1, or isolate noise from other sources of prediction error. Red and white wine are related tasks from one data source. A common optimization schedule controls the comparison within an architecture but does not establish the best achievable accuracy of either architecture. The test does not evaluate the shared-parameter protection procedure on real data.

Table 28: Full Wine Quality grid. Test MSE mean ± seed SD; eight paired splits per dataset and architecture. Every trained setting is included.

Data	Model	Location	Weight	MSE	MAE
Red	MLP	none	0	0.401 ± 0.033	0.492
Red	MLP	prediction	0.05	0.401 ± 0.033	0.491
Red	MLP	prediction	0.5	0.395 ± 0.037	0.487
Red	MLP	representation	0.05	0.402 ± 0.033	0.492
Red	MLP	representation	0.5	0.395 ± 0.040	0.488
Red	Transformer	none	0	0.405 ± 0.033	0.499
Red	Transformer	prediction	0.05	0.404 ± 0.034	0.499
Red	Transformer	prediction	0.5	0.405 ± 0.039	0.499
Red	Transformer	representation	0.05	0.407 ± 0.034	0.501
Red	Transformer	representation	0.5	0.410 ± 0.030	0.503
White	MLP	none	0	0.515 ± 0.034	0.560
White	MLP	prediction	0.05	0.514 ± 0.034	0.560
White	MLP	prediction	0.5	0.509 ± 0.030	0.558
White	MLP	representation	0.05	0.515 ± 0.034	0.560
White	MLP	representation	0.5	0.516 ± 0.032	0.561
White	Transformer	none	0	0.518 ± 0.033	0.565
White	Transformer	prediction	0.05	0.515 ± 0.031	0.563
White	Transformer	prediction	0.5	0.521 ± 0.035	0.567
White	Transformer	representation	0.05	0.517 ± 0.029	0.564
White	Transformer	representation	0.5	0.523 ± 0.034	0.567

Table 29: Validation-selected geometry weights, including zero. Counts report selections of $0, .05, .5$ in that order. Test differences are relative to no geometry, with descriptive paired 95% intervals.

Data	Model	Location	MSE	Counts	Difference [interval]
Red	MLP	representation	0.402	6/1/1	+0.0001 [+0.0000, +0.0002]
Red	MLP	prediction	0.400	3/1/4	-0.0010 [-0.0033, +0.0007]
Red	Transformer	representation	0.405	6/0/2	-0.0003 [-0.0020, +0.0010]
Red	Transformer	prediction	0.404	3/2/3	-0.0007 [-0.0018, +0.0004]
White	MLP	representation	0.514	3/1/4	-0.0002 [-0.0008, +0.0002]
White	MLP	prediction	0.509	0/2/6	-0.0057 [-0.0113, -0.0012]
White	Transformer	representation	0.519	3/1/4	+0.0013 [-0.0073, +0.0087]
White	Transformer	prediction	0.518	0/2/6	+0.0001 [-0.0112, +0.0097]