BEYOND INPUT INTERPOLATION: ISOLATING MIXUP'S FIRST-ORDER REGULARIZATION KERNEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixup couples input interpolation with stochastic-gradient regularization. We isolate a first-order regularization kernel through *deliberate purification*: retaining a specified gradient structure while removing the input-dependent interpolation residual. For target-affine prediction gradients, target interpolation is exact at source inputs; the removed response captures input nonlinearity at the common mixed target. Sombrero Loss realizes this kernel exactly on original inputs with paired targets, then extends to independent targets. For targets of uniform mean, its expected gradient matches label smoothing at an analytically determined strength, while intrinsic noise has an explicit, data-dependent covariance. Under a prediction model, the average conditional per-example covariance and model-averaged generalized Gauss–Newton matrix share two components, specifying curvature-matching conditions and a local mechanism that can favor flatter regions. Mean-matched ablations and spectral measurements support regularization beyond average smoothing. CIFAR and ImageNet experiments show competitive generalization without input interpolation and improved stability over tested mixing strengths. Detection, translation, and class-imbalanced classification demonstrate the construction's applicability and target-prior flexibility.

1 INTRODUCTION

Mixup trains a predictor on convex combinations of examples and labels (Zhang et al., 2018), changing both its evaluation inputs and stochastic gradients. Its success has motivated input, feature, and supervision variants (Yun et al., 2019; Verma et al., 2019; Kim et al., 2020; Liu et al., 2022) and analyses of interpolation-induced regularization.

We ask: *can a first-order structure extracted from Mixup support regularization on original inputs, and what accounts for its behavior?* Input mixing can incur costs: it contracts covariance under independent pairing, frozen mixed examples lose performance relative to continued resampling, and more physically faithful mixing need not improve classification. These observations motivate separating input interpolation from its optimization signal.

Our **first-order purification** isolates a combination of gradients at the source inputs and targets, which we call a **first-order regularization kernel**. For losses whose prediction gradients are affine in the target, including softmax cross-entropy and squared error, target interpolation is preserved exactly at those inputs. We deliberately remove the nonlinear input-interpolation response at the common mixed target, studying the retained signal even when this response is appreciable.

The resulting **Sombrero Loss** combines observed-target and perturbation-target losses on unchanged inputs. With a shared coefficient and bidirectional permutation, its gradient exactly realizes the retained batch expression; independent targets make the prior a design choice (Figure 1).

For independent targets with uniform mean, the expected gradient matches label smoothing (LS), while fluctuations have a covariance shaped by the Jacobian and target distribution. Centered and decoupled variants separate these effects, enabling mean-matched comparisons. Under a prediction model, the average conditional per-example covariance and model-averaged generalized Gauss–Newton (GGN) matrix share two structural components. Coefficient matching connects them to a local curvature–noise mechanism, examined through accuracy, Hessian spectra, and landscapes. Our contributions are:

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

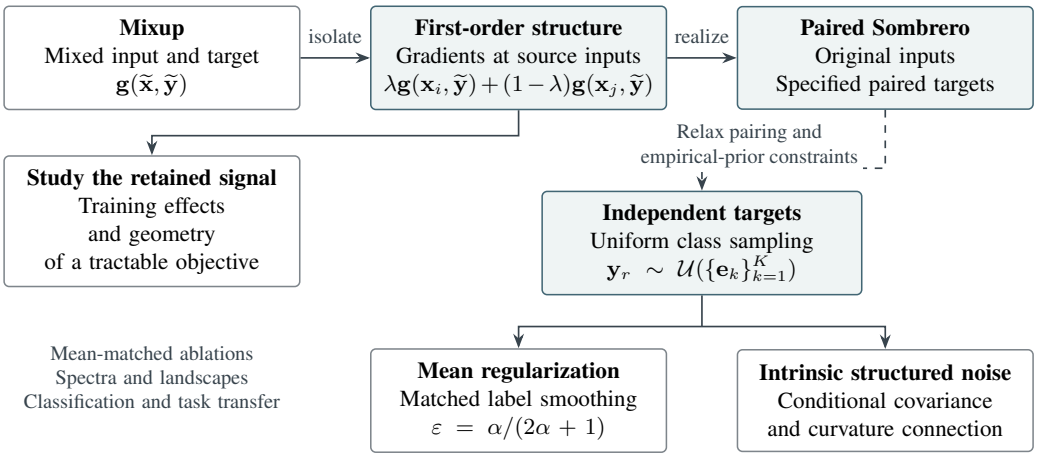


Figure 1: **Isolating a first-order regularization kernel from Mixup.** The retained gradient structure is realized by a paired objective on original inputs. Relaxing pairing and empirical class-frequency constraints permits independent targets with a chosen prior. The uniform instance combines label smoothing in expectation with structured intrinsic noise.

1. The extraction of a first-order regularization kernel from Mixup, its exact paired objective realization on original inputs, and an extension to independently sampled targets.
2. A characterization of mean regularization and intrinsic noise, a conditional connection to curvature, and mechanism tests using mean-matched interventions, spectra, and landscapes.
3. Evaluation across classification, mixing strengths, structured and discrete tasks, and class imbalance, including repeated runs and centered and decoupled variants.

2 Background and Related Work

Notation. For data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, a predictor with p parameters produces K scores $\boldsymbol{h}_i = \mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\theta})$ and loss $\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}) = \mathcal{C}(\boldsymbol{h}_i, \boldsymbol{y}_i)$. Write $\mathbf{g}_i = \nabla_{\boldsymbol{\theta}}\mathcal{L}_i$, $\mathbf{H}_i = \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_i$, and $\mathbf{J}_i = \nabla_{\boldsymbol{\theta}}\mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\theta}) \in \mathbb{R}^{K\times p}$. Subscript $\mathcal{B}$ denotes averaging over a batch of size b; $\mathbf{1}_K$, I_K, and $\{\{\cdot\}\}$ denote the all-ones vector, identity matrix, and a multiset. Mixup interpolates inputs and targets with $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$; its paired construction is developed in Sec. 3. For softmax cross-entropy (CE) and half squared error, the prediction gradients are $\mathrm{softmax}(\boldsymbol{h}) - \boldsymbol{y}$ and $\boldsymbol{h} - \boldsymbol{y}$, respectively: both are affine in the target, without requiring nearby labels.

Mixup and stochastic supervision. Loss-based analyses characterize Mixup through regularization terms, transformed data and perturbation covariances, or mixed-sample masks (Zhang et al., 2021; Carratino et al., 2022; Park et al., 2022); feature-learning analysis explains benefits for rare features (Zou et al., 2023). We instead realize a retained gradient structure on original inputs, making its mean and fluctuations separately accessible. Label smoothing (LS) modifies the target distribution (Szegedy et al., 2016; Müller et al., 2019), while randomized soft targets introduce additional stochasticity (dos Santos & Papa, 2024). Sombrero's distinction within this family is its paired realization of the retained Mixup expression and the resulting mean–covariance analysis; independent targets extend the construction through a chosen prior.

Noise geometry and transfer. Anisotropic noise and noise–curvature alignment can affect optimization (Zhu et al., 2019; Wu et al., 2022). We connect target-induced noise to modeled GGN curvature and test its contribution with mean-matched interventions, without SAM's explicit parameter-space maximization (Foret et al., 2021). LossMix interpolates supervision through losses evaluated on mixed inputs (Vu et al., 2024); Sombrero keeps inputs unchanged, enabling structured and discrete applications. Appendix E gives the expanded background, including objective modifications and further connections to label smoothing.

3 From Mixup to First-Order Purification

3.1 Motivation: what changes when inputs are mixed?

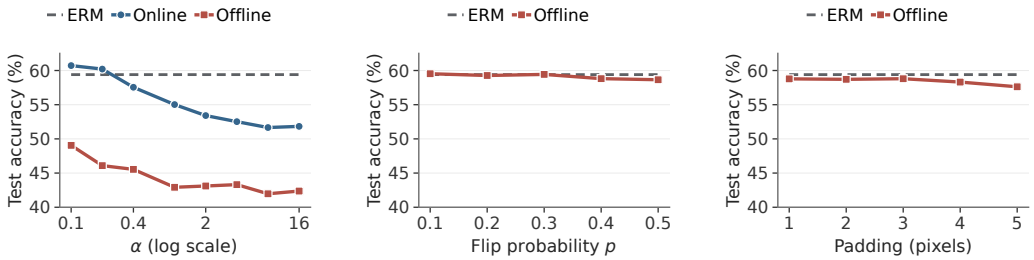
Input interpolation changes the training distribution. We examine its second moments, frozen mixed examples, and a physically motivated mixing rule to motivate which part of Mixup to isolate.

For independently drawn inputs with finite covariance Σ and an independent $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$, Mixup preserves the mean and contracts the covariance:

$$\Sigma_\alpha = c_\alpha \Sigma, \qquad c_\alpha = \frac{\alpha + 1}{2\alpha + 1} < 1 \quad (\alpha > 0). \tag{1}$$

Appendix A.1 gives the positive Wasserstein lower bound for $\mathrm{Tr}(\Sigma) > 0$ (Theorem 5; Gelbrich, 1990) and distributional illustrations. Mixup can contract covariance while introducing points outside the original support, complementing the ambiguous-supervision concern of manifold intrusion (Guo et al., 2019).

Frozen support and resampling during training. Offline augmentation reuses examples generated in the initial epoch. Under this common freeze protocol, Mixup degrades more strongly as mixing increases than random flipping and cropping (Figure 2). Continued resampling changes both the examples encountered and their gradient fluctuations, motivating an optimization-level analysis.



(a) Mixup (b) Random flip (c) Random crop

Figure 2: **Frozen support and resampling.** Online and offline Mixup on CIFAR-100 (left), compared with random flipping (middle) and cropping (right) under the same freeze protocol. Offline variants reuse examples generated during the initial epoch.

Physical fidelity as a complementary observation. Gixup varies mixing between encoded intensities and approximately linear radiometric space (rule, derivation, and study: Appendix D.1). Physically motivated mixing avoids darkening in illustrative color mixtures but does not give the best classification result: input realism and regularization need not improve together.

These observations suggest that input interpolation can incur costs alongside its regularization effects. We therefore separate the signal at original inputs from the nonlinear response along the interpolation path, deliberately retaining the former to study its mean effect, fluctuations, and training behavior.

3.2 Decomposition and Deliberate Purification

The kernel retained by **first-order purification** is the expression $\mathbf{g}_{ij}^{\mathrm{ret}}$ below. Write $\mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ at fixed parameters. For a pair, let $\widetilde{\boldsymbol{x}} = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j$ and $\widetilde{\boldsymbol{y}} = \lambda \boldsymbol{y}_i + (1-\lambda)\boldsymbol{y}_j$. Expanding the chain rule's Jacobian and criterion-gradient factors to first order around a common reference point gives the following decomposition.

Lemma 1 (Gradient Decomposition)**.** *Under the local smoothness conditions in Assumption 7, define the retained expression*

$$\mathbf{g}_{ij}^{\mathrm{ret}} = \underbrace{\lambda^2 \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_i) + (1-\lambda)^2 \mathbf{g}(\boldsymbol{x}_j, \boldsymbol{y}_j)}_{\textit{Effective Terms}} + \underbrace{\lambda(1-\lambda)\left[\mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_j) + \mathbf{g}(\boldsymbol{x}_j, \boldsymbol{y}_i)\right]}_{\textit{Perturbation Terms}},$$

$$\mathbf{g}(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) = \mathbf{g}_{ij}^{\mathrm{ret}} + \mathbf{r}_{ij}, \qquad \mathbf{r}_{ij} = \mathcal{O}(\rho^2). \tag{2}$$

Here ρ bounds the endpoint deviations in the joint input–target space from the common reference point. The quadratic order is a local statement; the retained expression is defined at any endpoint separation. The proof, including cancellation of the input-linear cross terms, appears in Appendix A.2.

The Effective Terms preserve input–target associations; the Perturbation Terms evaluate the same inputs against the other targets. For losses with target-affine parameter gradients, such as softmax CE and MSE, their sum has the exact endpoint representation

$$\mathbf{g}_{ij}^{\mathrm{ret}} = \lambda \mathbf{g}(\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}) + (1-\lambda)\mathbf{g}(\boldsymbol{x}_j, \widetilde{\boldsymbol{y}}).$$

Here $\mathbf{r}_{ij} = \mathbf{r}_{ij}^{\mathrm{inp}}$ is the Jensen gap along the input segment at the fixed common target $\widetilde{\boldsymbol{y}}$. Appendix A.2 bounds it by the squared input distance under a Lipschitz input-derivative condition, without requiring nearby labels. The endpoint identity is exact regardless of residual size; Appendix D.6 probes the batch discrepancy.

Input and target purification. We deliberately remove $\mathbf{r}_{ij}^{\mathrm{inp}}$, including curvature responses between training examples, and evaluate the retained signal through training. The preceding geometric and offline observations motivate this choice. For target-affine criteria, pure target derivatives of the criterion gradient of order two and higher vanish, so target interpolation at the source inputs is retained exactly. The removed input response can still depend on the common mixed target. Purification is thus an active choice about the input-dependent response, with no small-residual requirement for defining or training the objective.

For a batch $\mathcal{B}$ of size b, pair examples using a permutation $f \in S_b$ and a single shared coefficient λ. Reindexing the retained gradients combines the two effective weights into $\lambda_s = \lambda^2 + (1-\lambda)^2 = 1 - 2\lambda(1-\lambda)$ and pairs each input with targets from both f and f^{-1} (Corollary 8, Appendix A.3). This yields an objective on the original inputs, constructed next.

3.3 Sombrero: Paired Realization and Uniform Extension

The retained structure suggests a simple loss-level construction:

$$\mathcal{SL}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{y}_r) = \underbrace{\lambda_s \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})}_{\text{Effective Term}} + \underbrace{(1-\lambda_s)\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}_r)}_{\text{Perturbation Term}}. \tag{3}$$

We call this **Sombrero Loss**. The retention rate $\lambda_s = \lambda^2 + (1-\lambda)^2 \in [1/2, 1]$ is induced by $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$; Appendix B.2 derives its distribution, bounded variance, and dependence on α.

Exact paired realization. Using forward and inverse targets from the same permutation, define

$$\mathcal{SL}_{\mathcal{B}}^{(1)} = \frac{1}{2b} \sum_{i=1}^{b} \sum_{\pi \in \{\{f, f^{-1}\}\}} \mathcal{SL}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{y}_{\pi(i)}).$$

Its gradient is *exactly* $\mathbf{g}_{\mathcal{B}}^{\mathrm{ret}} = b^{-1} \sum_i \mathbf{g}_{i,f(i)}^{\mathrm{ret}}$ for the same batch, permutation, and shared coefficient (Proposition 10). The full Mixup gradient additionally contains the averaged input residual $\mathbf{r}_{\mathcal{B}}^{\mathrm{inp}}$. Both cross-target contributions retain their multiplicity, including when $f = f^{-1}$; a single batch shuffle does not in general realize their average. These identities fix model state and other randomness; example-specific coefficients require reindexing their weights as well. The paired identity and input residual extend to batch-dependent, permutation-equivariant predictors, including training-mode batch normalization with statistics recomputed and differentiated through each forward map (Appendix A.4). Smoothness is needed only for the local remainder estimate.

Independent targets and task transfer. We relax target sampling in two steps: under uniform random pairing independent of the coefficient, independent draws from the batch prior preserve the pairing-averaged mean gradient; choosing a different prior relaxes the empirical-frequency constraint. Our practical **Uniform** classification variant samples $\boldsymbol{y}_r$ uniformly from the K one-hot vectors, independently of λ_s. It retains the loss form and retention law, with mean and noise analyzed under its sampling rule in Sec. 4. Section 5.3 compares it with batch-shuffled targets on imbalanced data. These *data-free perturbation targets* preserve input alignment for discrete tokens, variable-length

sequences, and structured predictions independently of batch composition. Section 5.2 uses task-specific targets for detection and translation; Appendix D.3 describes the task-specific losses and settings. Appendix B.4 compares requirements; Figure 4 shows the pipelines. No explicit Hessian computation or adversarial weight perturbation is required. The mean and covariance identities below use their stated loss and sampling assumptions.

4 Mean Regularization and Structured Noise

We analyze the mean and fluctuations of Sombrero's independent-target extension, then relate classification noise to curvature through a conditional prediction model.

4.1 Mean Regularization and Matched Comparisons

Proposition 2 (Expectation Equivalence)**.** *Suppose $\nabla_{\boldsymbol{h}}\mathcal{C}(\boldsymbol{h}, \boldsymbol{y})$ is affine in $\boldsymbol{y}$. At fixed example and parameters, let λ_s and $\boldsymbol{y}_r$ be independent, with $\mathbb{E}[\boldsymbol{y}_r] = \boldsymbol{\mu}$. Then*

$$\mathbb{E}_{\lambda_s, \boldsymbol{y}_r}[\mathbf{g}_i^{(\mathcal{SL})}] = \mathbf{g}(\boldsymbol{x}_i, (1-\varepsilon)\boldsymbol{y}_i + \varepsilon\boldsymbol{\mu}), \qquad \varepsilon = 1 - \mu_{\lambda_s} = \frac{\alpha}{2\alpha+1},$$

where $\mu_{\lambda_s} = \mathbb{E}[\lambda_s]$. With $\boldsymbol{\mu} = \mathbf{1}_K/K$, this is exactly the gradient of Label Smoothing (LS). The equality also holds after averaging over examples. The proof is in Appendix C.1.

Softmax CE and MSE satisfy this condition. CE also shares the expected LS loss; other target-affine criteria may differ by target-dependent constants without changing gradients. This *mean smoothing effect* matches expected gradients at fixed parameters, without ensuring identical trajectories or lower values of every curvature statistic. To separate mean and noise, **Cen-Sombrero** restores the ERM mean by setting $\mathcal{SL}_{cen} = \mathcal{SL} - \mathbb{E}[\mathcal{SL}] + \mathcal{L}$. Centering preserves algorithmic gradient covariance conditional on the same example or batch, parameters, and sampling protocol; total covariance along training can change. Dec-Sombrero uses the mean from α_1 and centered noise from α_2 to control smoothing and noise separately (objectives and equivalent targets: Appendix B.3). In Table 1, Sombrero outperforms matched LS and Cen-Sombrero outperforms ERM on both datasets, supporting stochasticity beyond mean regularization. These comparisons, using ResNet-18 and MobileNetV2 (Sandler et al., 2018), vary the perturbation as a whole.

Table 1: **Two mean-matched comparisons.** Accuracy (%). CIFAR-100 ERM: five-run mean±SD; others: single runs. Both variants: $\alpha = 0.6$; LS: $\varepsilon = \alpha/(2\alpha+1)$. Blocks match conditional mean gradients.

Method	CIFAR-100 ResNet-18	CIFAR-10 MobileNetV2
Label Smoothing	77.71	94.64
Sombrero	**78.49**	**94.75**
Baseline (ERM)	76.23 ± 0.42	94.24
Cen-Sombrero	76.80	94.29

Table 2: **Hessian spectrum statistics on CIFAR-100 with ResNet-18.** Lanczos estimates of the mean, second moment, and maximum eigenvalue ξ; protocol in Appendix D.2.1.

Method	$\mathbb{E}[\xi]$ $(\times 10^{-6})$	$\mathbb{E}[\xi^2]$ $(\times 10^{-6})$	ξ_{max}
Label Smoothing	336.7	6933.6	125.1
Sombrero	57.2	559.0	72.2
Baseline (ERM)	47.6	501.7	47.4
Cen-Sombrero	**7.0**	**7.8**	**6.3**

4.2 Conditional Structured Noise

At fixed parameters, total covariance separates example sampling from loss randomness:

$$\mathrm{Cov}[\mathbf{g}_i^{(\mathcal{SL})}] = \underbrace{\mathrm{Cov}_i\Big[\mathbb{E}_{\lambda_s, \boldsymbol{y}_r}\mathbf{g}_i^{(\mathcal{SL})}\Big]}_{\Sigma_{sn}\text{: Sampling Noise}} + \underbrace{\mathbb{E}_i\Big[\mathrm{Cov}_{\lambda_s, \boldsymbol{y}_r}(\mathbf{g}_i^{(\mathcal{SL})} \mid i)\Big]}_{\Sigma_{in}\text{: Intrinsic Noise}}. \tag{4}$$

For Softmax CE, $\mathbf{g}_i = \mathbf{J}_i^\top(\boldsymbol{o}_i - \boldsymbol{y}_i)$ with $\boldsymbol{o}_i = \mathrm{softmax}(\mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\theta}))$. The conditional perturbation has two sources: fluctuations of $w = 1 - \lambda_s$ and sampling of the target. Both are mapped into parameter space by the Jacobian, producing structured noise even for uniform targets.

Proposition 3 (Structure of Intrinsic Noise)**.** *For one-hot source labels and uniformly sampled one-hot perturbation targets independent of λ_s, define*

$$\mathbf{P} = I_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top, \qquad \mathbf{Q}_i = K(\boldsymbol{y}_i - \mathbf{1}_K/K)(\boldsymbol{y}_i - \mathbf{1}_K/K)^\top,$$
$$\mathbf{U}_{\text{base}} = \mathbb{E}_i[\mathbf{J}_i^\top \mathbf{P}\mathbf{J}_i], \qquad \mathbf{U}_{\text{cls}} = \mathbb{E}_i[\mathbf{J}_i^\top \mathbf{Q}_i\mathbf{J}_i].$$

With $c_{in} = \mathbb{E}[w^2]$ and $v_s = \mathrm{Var}(w)$, the average conditional per-example covariance is exactly

$$\Sigma_{in} = \frac{c_{in}}{K}\left(\mathbf{U}_{\text{base}} + \gamma_{in}\mathbf{U}_{\text{cls}}\right), \quad c_{in} = \frac{\alpha(\alpha+1)}{(2\alpha+1)(2\alpha+3)}, \quad \gamma_{in} = \frac{v_s}{c_{in}} = \frac{1}{(2\alpha+1)(\alpha+1)}. \tag{5}$$

Both structural matrices are PSD, and $\gamma_{in} \in (0,1)$ for $\alpha > 0$. The result does not require balanced source-class frequencies. See Appendix C.2.

From examples to the implemented batch. Classification shares one coefficient per batch and draws independent uniform one-hot targets, including the observed class. Its cross-example terms distinguish batch covariance (Eq. (63)) from Eq. (5). Appendix C.2 gives general-prior and independent/shared-coefficient formulas conditional on inputs and parameters before optimizer transformations, and discusses permutation-target dependence. Batch-dependent predictors differentiate through batch statistics (Appendix A.4).

4.3 Curvature Alignment and Empirical Geometry

To connect Eq. (5) to curvature, consider the actual CE GGN, $\tilde{\mathbf{H}} = \mathbb{E}_i[\mathbf{J}_i^\top(\mathrm{diag}(\boldsymbol{o}_i) - \boldsymbol{o}_i\boldsymbol{o}_i^\top)\mathbf{J}_i]$. It approximates the full Hessian when the model-second-derivative remainder is negligible. We expose its structure through an auxiliary ϵ-contamination prediction model (Huber, 1964), $\boldsymbol{o}_i = (1-\omega_i)\boldsymbol{y}_i + \omega_i\boldsymbol{u}_i$. Here $\omega_i \in [0,1]$ denotes prediction uncertainty, distinct from target smoothing ε; $\boldsymbol{u}_i$ is a permutation-invariant probability vector with mean $\mathbf{1}_K/K$ and covariance $\sigma_u^2\mathbf{P}$.

Proposition 4 (Modeled Hessian Structure)**.** *Assume that ω_i and $\boldsymbol{u}_i$ are independent and their required moments factor from the sample label and Jacobian. The model-averaged GGN is*

$$\tilde{\mathbf{H}}_{\text{mod}} = \frac{c_{\omega,u}}{K}\left(\mathbf{U}_{\text{base}} + \gamma_H\mathbf{U}_{\text{cls}}\right). \tag{6}$$

The actual GGN satisfies $\tilde{\mathbf{H}} \approx \tilde{\mathbf{H}}_{\text{mod}}$ when the model describes the learned predictions. Here $c_\omega = \mathbb{E}[\omega - \omega^2]$, $c_{\omega,u} = \mathbb{E}[\omega - K\sigma_u^2\omega^2]$, and $\gamma_H = c_\omega/c_{\omega,u} \in [0,1]$ when $c_{\omega,u} > 0$. The derivation and the full Hessian–GGN relation appear in Appendix C.3.

The alignment principle. Noise and modeled curvature share two PSD structural matrices. If $\gamma_{in} = \gamma_H$, then $\Sigma_{in} = (c_{in}/c_{\omega,u})\tilde{\mathbf{H}}_{\text{mod}}$. For unequal positive coefficients, directional PSD bounds tighten as their ratio approaches one (Appendix C.3). Shared structure alone does not imply identical leading eigenvectors because the two matrices need not commute.

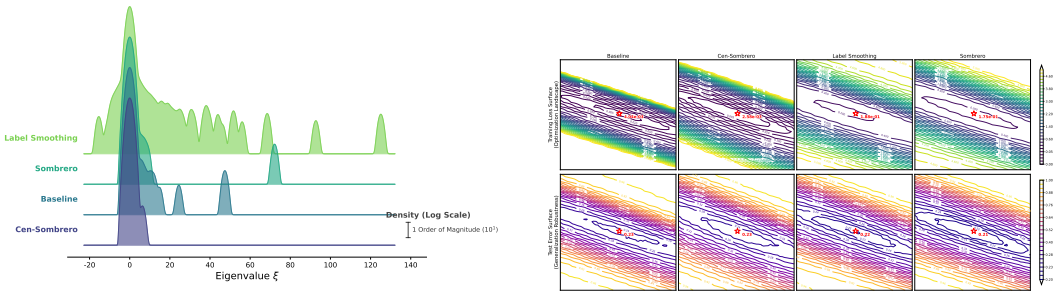
Estimating and calibrating the coefficients. Prediction moments make the model coefficients empirically accessible. Appendix C.4 gives the identification conditions and five-seed estimates across CIFAR-100, CIFAR-10, and MNIST (Table 8). When prediction uncertainty concentrates near ε and $K\sigma_u^2$ is small, $\gamma_H \approx 1-\varepsilon$, giving $\gamma_{in}/\gamma_H \approx (1+\alpha)^{-2}$. Figure 8 reports the empirical trends and deviations. As $\alpha \to 0$, the ratio approaches one but noise magnitude vanishes. Reported Dec-Sombrero matches coefficients under this approximation by selecting mean smoothing from its single free noise parameter (calibration and valid range: Appendices C.5 and B.3).

Curvature–noise interaction. For a quadratic loss with PSD curvature $\mathbf{H}$, a centered stochastic gradient perturbation of covariance Σ increases expected loss by $\eta^2\,\mathrm{Tr}(\mathbf{H}\Sigma)/2$ relative to the same noiseless step of size η. Thus $\mathcal{E}(\mathbf{H},\Sigma) = \mathrm{Tr}(\mathbf{H}\Sigma)$ measures local energy injection into curved directions (Zhu et al., 2019). At fixed positive covariance trace τ and $\mathrm{Tr}(\mathbf{H}) > 0$, compare aligned noise $\tau\mathbf{H}/\mathrm{Tr}(\mathbf{H})$ with isotropic noise $\tau I_p/p$:

$$\frac{\mathcal{E}(\mathbf{H}, \tau\mathbf{H}/\mathrm{Tr}(\mathbf{H}))}{\mathcal{E}(\mathbf{H}, \tau I_p/p)} = \frac{p\,\mathrm{Tr}(\mathbf{H}^2)}{\mathrm{Tr}(\mathbf{H})^2} \geq 1.$$

Unequal curvature eigenvalues make the inequality strict: directional allocation alone increases this interaction. Appendix C.6 derives the a^2 versus a comparison across minima with $\mathbf{H}_s = a\mathbf{H}_f$, $a > 1$, under a shared covariance proportionality factor; this comparison also changes total noise energy. These local results motivate investigating flat-minimum selection without explicit sharpness-aware updates. Basin-exit times also depend on barriers, drift, and noise away from the minimum; optimizer-specific conclusions require the implemented batch covariance and update geometry.

Observed curvature and landscapes. Matched LS has a larger second spectral moment than ERM on ResNet-18/CIFAR-100 (Table 2). Sombrero–LS and Cen-Sombrero–ERM comparisons show reductions in all three statistics, distinguishing geometry associated with smoothing from changes accompanying stochasticity. Filter-normalized loss and error landscapes (Figure 3) complement mean-matched accuracy; Appendix D.2 gives protocols and hard/smoothed evaluation-label comparisons.



(a) Hessian spectral density.

(b) Loss and error landscapes.

Figure 3: **Empirical geometry of the learned solutions.** Hessian spectral densities (left) and filter-normalized loss and error landscapes (right) for the reported checkpoints, comparing Sombrero–LS and Cen-Sombrero–ERM alongside Table 2.

5 Empirical Evaluation

Experiments test the extracted kernel's utility through original-input regularization, task-loss transfer, and target-prior flexibility, complementing Section 4's mean-matched mechanism tests. Classification uses independent uniform targets unless specified; detection uses random assignment.

5.1 Effectiveness and Stability on Original Inputs

Sombrero is competitive with input-space Mixup and hidden-state Manifold Mixup (Verma et al., 2019) on CIFAR-100; ImageNet-1K ResNet-18 also favors Sombrero (Tables 3a–3b). The ImageNet repeats use the same recipe as the corresponding point estimates. Across the four reported ImageNet-1K architectures, Sombrero improves the baselines, with Dec-Sombrero yielding further gains in several settings (Table 4). In the CIFAR-100 strength sweep (Figure 4), Sombrero varies by about one accuracy point over $\alpha \in [0.1, 2.0]$, has SDs ≤ 0.37 points, peaks at $\alpha = 0.6$, and approaches ERM at $\alpha = 0.01$ as the perturbation vanishes. Over $\alpha \in \{4, 8, 16, 32\}$, it stays within 77.33–77.61%, whereas Mixup declines to 72.17% at $\alpha = 32$. These full constructions differ in target sampling; Appendix D.5 reports all means and SDs.

Target mixing and the purified objective. Mixup-Target Only (Mixup-TO) (Wang et al., 2023) retains batch-paired label interpolation on unchanged inputs. With Pre-Activation ResNet-18 on CIFAR-100, Mixup-TO and Sombrero reach 77.10% and 78.49% in single runs, versus ERM's five-run mean of 76.23% (Appendix Table 9). This comparison jointly changes coefficient laws and target sampling; the mean-matched tests in Section 4 isolate stochastic perturbations more directly.

5.2 Transfer to Structured and Discrete Prediction

Task-specific prediction–target loss perturbations preserve object layouts and token sequences. On MS COCO (Lin et al., 2014), Sombrero combines Hungarian- and random-assignment classification

Table 3: **Repeated-run classification.** Top-1 accuracy (%, mean±SD). (a) Pre-Activation ResNet-18, 200 epochs, five seeds. (b) ResNet-18, 100 epochs, three seeds. Methods share a recipe within each panel. Manifold Mixup's α grid for the shorter recipe is in Appendix D.5.

(a) CIFAR-100.

Method	α	Top-1 (%)
ERM	–	76.23 ± 0.42
Mixup	1.0	77.94 ± 0.56
Manifold Mixup	1.5	78.34 ± 0.35
Sombrero	0.6	78.59 ± 0.30

(b) ImageNet-1K ($\alpha = 0.2$ for Sombrero).

Method	Top-1 (%)
ERM	69.65 ± 0.21
Mixup	70.41 ± 0.28
Sombrero	71.12 ± 0.23

Table 4: **ImageNet-1K classification.** Validation Top-1/Top-5 point estimates (%); ResNet-18 repeats: Table 3b. Bold: best result from this work per architecture and metric.

(a) ShuffleNet V2 1.0x and ResNet-18.

Method	**ShuffleNet V2 1.0x**		**ResNet-18**	
	Top-1	**Top-5**	**Top-1**	**Top-5**
ERM	69.2	88.5	69.7	89.1
Mixup	69.9	89.5	70.5	89.6
Sombrero	**70.5**	89.5	71.1	89.9
Dec-Somb.	70.3	**90.1**	**71.3**	**90.2**

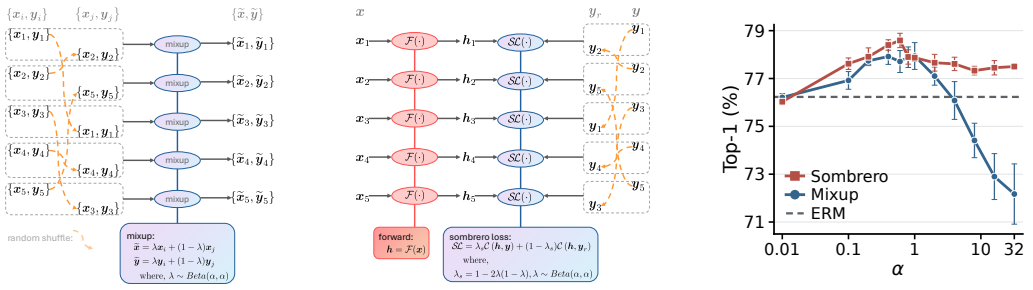
Notes. Sombrero: $\alpha = 0.2$; Dec-Somb.: $\alpha = 0.15$. Panel (a) reports our experiments; its ShuffleNet V2 ERM baseline includes label smoothing.

External recipes in (b): CutMix (Yun et al., 2019); DeiT uses Mixup, CutMix, and RandAugment (Touvron et al., 2021).

(b) ResNet-50 and DeiT-Small.

Method	**ResNet-50**		**DeiT-Small**	
	Top-1	**Top-5**	**Top-1**	**Top-5**
Experiments in this work				
ERM	76.7	92.8	72.6	93.8
Mixup	77.4	93.6	-	-
Sombrero	**78.2**	**94.5**	78.5	95.1
Dec-Somb.	-	-	**79.6**	**95.5**
External references				
CutMix	78.6	94.1	-	-
DeiT recipe	-	-	79.8	94.9

and box losses in Deformable-DETR (Zhu et al., 2021) after warmup (Appendix D.3). Overall AP rises from 44.0 to 44.4, close to task-specific LossMix's 44.6 (Vu et al., 2024) (Table 5). These results support loss-level transfer without image mixing. For translation, Transformer-Base (Vaswani et al., 2017) tests discrete-sequence regularization on WMT 2014 English–German/French (Bojar et al., 2014). Sombrero at $\alpha = 0.125$ has nominal mean smoothing $\varepsilon = \alpha/(2\alpha + 1) = 0.1$ and matches or improves BLEU over the LS baseline; $\alpha = 0.15$ improves both language pairs (Table 6).



(a) Mixup (b) Sombrero (c) Sensitivity to α

Figure 4: **Training pipelines and sensitivity to mixing strength.** Mixup interpolates inputs and targets (left); Sombrero perturbs the objective on original inputs (middle). Right: CIFAR-100 Top-1 accuracy (%) with Pre-Activation ResNet-18; markers show means and error bars SDs over five independent runs, with a logarithmic α axis. Numerical values: Table 13.

Table 5: **Object detection.** Average Precision (AP) point estimates: MS COCO *val2017*, Deformable-DETR (ResNet-50); Sombrero: $\alpha = 0.4$.

Method	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_S$	**AP**$_M$	**AP**$_L$
Baseline	44.0	62.6	47.8	**26.4**	47.6	58.2
LossMix	**44.6**	**62.8**	48.0	26.3	48.1	58.8
Sombrero	44.4	62.5	**48.3**	26.3	**48.2**	**59.1**

Table 6: **Machine translation.** BLEU point estimates on WMT 2014 English–German and English–French (newstest2014).

Method	En-De	En-Fr
Baseline (LS $\varepsilon = 0.1$)	27.4	38.4
Sombrero ($\alpha = 0.125$)	27.4	38.6
Sombrero ($\alpha = 0.15$)	**27.7**	**39.0**

5.3 Sampling Priors Under Class Imbalance

Batch-shuffled targets follow empirical class frequencies (Chou et al., 2020; Zhong et al., 2021); uniform targets relax this constraint (Figure 5). At $\gamma_{imb} = 0.001$, Uniform Sombrero improves CIFAR-100 majority Macro-F1 over ERM and Mixup and partially recovers the Mixup minority drop, with incomplete recovery relative to ERM. These group-dependent gains illustrate target-prior flexibility. Appendix D.4 reports full CIFAR-10/100 results, Balanced Accuracy, and group values (Table 11).

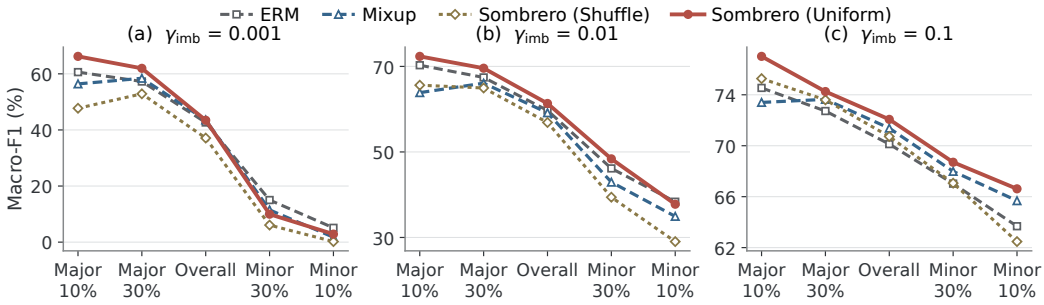


Figure 5: **Target priors under class imbalance.** Macro-F1 on long-tailed CIFAR-100 for $\gamma_{imb} = 0.001, 0.01, 0.1$ (left to right), comparing batch-coupled and independent uniform target sampling.

6 Discussion and Conclusion

We isolate a first-order regularization kernel from Mixup and realize it on original inputs through Sombrero Loss. With a shared coefficient and bidirectional pairing, its gradient exactly realizes the retained expression; independent targets extend the construction to chosen priors. Uniform Sombrero exposes mean label smoothing and Jacobian-shaped noise; centered and decoupled variants enable interventions on these effects. Mean-matched accuracy, spectra, and landscapes support regularization beyond average smoothing. Classification and strength scans demonstrate utility and stability in the tested settings, while detection, translation, and long-tail results illustrate the flexibility of preserving inputs and choosing target priors.

The geometric account conditions on examples or batches and their sampling rules: proportionality to modeled GGN requires coefficient matching, and extension to the full Hessian requires residual control. Training trajectories, reparameterizations, and detection assignment noise remain open directions. Spatial mixing such as CutMix and comparisons with SAM can connect the extracted kernel to domain priors and explicit parameter-space regularization. Larger sequence and language models call for suitable target perturbations and evaluation of their dynamics (Appendix F). These directions build on a tractable construction for analyzing and applying regularization independently of input interpolation.

Reproducibility Statement

The assumptions and proofs for the gradient decomposition, paired realization, and mean and covariance analysis are provided in Appendices A, B, and C. The empirical protocols, task-specific loss constructions, and evaluation settings are detailed in Appendix D. Appendix D.5 distinguishes repeated-run summaries from single-run comparisons and reports the complete mixing-strength sweep.

AI Use Statement

Generative AI tools assisted with manuscript organization and language editing, developing and checking parts of the theoretical derivations, interpreting existing experimental results, preparing the conceptual figure, and locating and verifying references. The authors are responsible for the final content of this work, including AI-assisted text and derivations.

References

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pp. 12–58. Association for Computational Linguistics, 2014.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization. *Journal of Machine Learning Research*, 23(325):1–31, 2022. URL https://jmlr.org/papers/v23/20-1385.html.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in Neural Information Processing Systems*, volume 13, 2000.

Muthu Chidambaram and Rong Ge. For better or for worse? learning minimum variance features with label augmentation. In *International Conference on Learning Representations*, pp. 52807–52831, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/83134e784cc1dd9584aeee20e96c2bf1-Abstract-Conference.html.

Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pp. 95–110. Springer, 2020.

Claudio Filipi Gonçalves dos Santos and João Paulo Papa. Rethinking regularization with random label smoothing. *Neural Processing Letters*, 56:157, 2024. doi: 10.1007/s11063-024-11579-z. URL https://link.springer.com/article/10.1007/s11063-024-11579-z.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Matthias Gelbrich. On a formula for the l2 wasserstein metric between measures on euclidean and hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3714–3722, 2019.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964.

Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, 2020.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Jihao Liu, Boxiao Liu, Hang Zhou, Hongsheng Li, and Yu Liu. Tokenmix: Rethinking image mixing for data augmentation in vision transformers. In *European Conference on Computer Vision (ECCV)*, pp. 455–471, 2022.

Zicheng Liu, Siyuan Li, Ge Wang, Lirong Wu, Cheng Tan, and Stan Z. Li. Harnessing hard mixed samples with decoupled regularizer. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52884–52906, 2023. doi: 10.52202/075280-2301. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a5c47c1b7adf19e8dc633812a4acf6d2-Abstract-Conference.html.

James Martens. Deep learning via hessian-free optimization. In *International Conference on Machine Learning (ICML)*, 2010.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Chanwoo Park, Sangdoo Yun, and Sanghyuk Chun. A unified analysis of mixed sample data augmentation: A loss function perspective. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35504–35518, 2022. doi: 10.52202/068431-2573. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/e6f32e64b9c27d153b46c94f0fe22b56-Abstract-Conference.html.

Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip HS Torr, and Puneet K Dokania. Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14608–14622, 2022. doi: 10.52202/068431-1062. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/5ddcfaad1cb72ce6f1a365e8f1ecf791-Abstract-Conference.html.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Stanley S Stevens. On the psychophysical law. *Psychological Review*, 64(3):153–181, 1957.

Michael Stokes, Matthew Anderson, Srinivasan Chandrasekar, and Ricardo Motta. A standard default color space for the internet—sRGB, 1996. URL https://www.w3.org/Graphics/Color/sRGB. Version 1.10, November 5, 1996.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, 2019.

Thanh Vu, Baochen Sun, Bodi Yuan, Alex Ngai, Yueqi Li, and Jan-Michael Frahm. Supervision interpolation via lossmix: Generalizing mixup for object detection and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5280–5288, 2024. doi: 10.1609/aaai.v38i6.28335. URL https://ojs.aaai.org/index.php/AAAI/article/view/28335.

Deng-Bao Wang, Lanqing Li, Peilin Zhao, Pheng-Ann Heng, and Min-Ling Zhang. On the pitfall of mixup for uncertainty calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Lei Wu, Mingze Wang, and Weijie Su. The alignment property of SGD noise and how it helps select flat minima: A stability analysis. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4680–4693, 2022. doi: 10.52202/068431-0338. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/1e55c38dd7d465c2526ae29d7ec85861-Abstract-Conference.html.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=8yKEo06dKNo.

Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16489–16498, 2021.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *International Conference on Machine Learning (ICML)*, 2019.

Difan Zou, Yuan Cao, Yuanzhi Li, and Quanquan Gu. The benefits of mixup for feature learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 43423–43479. PMLR, 2023. URL https://proceedings.mlr.press/v202/zou23a.html.

A Theoretical Analysis of Mixup

A.1 Proof of Theorem 5: Variance Contraction and Distributional Shift

We prove the covariance contraction and distributional-shift result used to motivate first-order purification. The theorem concerns the global second moments of independently paired samples. It distinguishes the distribution induced by interpolation from the original distribution and provides a geometric setting for investigating its optimization effects.

Theorem 5 (Variance contraction and distributional shift)**.** *Let two inputs be independent draws from a distribution $\mathcal{P}$ with finite covariance Σ, and let $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ be independent of them, with $\alpha > 0$. The mixed-input distribution $\mathcal{P}_\alpha$ has the same mean and covariance*

$$\Sigma_\alpha = c_\alpha \Sigma, \qquad c_\alpha = \frac{\alpha + 1}{2\alpha + 1} < 1.$$

If $\mathrm{Tr}(\Sigma) > 0$, its covariance trace is strictly smaller, and

$$W_2(\mathcal{P}, \mathcal{P}_\alpha) \geq (1 - \sqrt{c_\alpha})\sqrt{\mathrm{Tr}(\Sigma)} > 0. \tag{7}$$

Proof. **Setup and Notations.** Let $\boldsymbol{\mu}$ denote the mean of distribution $\mathcal{P}$. Let $\boldsymbol{v}, \boldsymbol{v}' \sim \mathcal{P}$ be two independent and identically distributed (i.i.d.) random variables drawn from $\mathcal{P}$. (In supervised learning, each sample contains input and label $\boldsymbol{v} = (\boldsymbol{x}, \boldsymbol{y})$.) The Mixup distribution $\mathcal{P}_\alpha$ is defined as the distribution of the random vector $\tilde{\boldsymbol{v}}$, generated by the convex combination $\tilde{\boldsymbol{v}} = \lambda \boldsymbol{v} + (1 - \lambda)\boldsymbol{v}'$, where $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$.

1. Variance Contraction. We first derive the first and second moments of the induced distribution $\mathcal{P}_\alpha$. By the Law of Total Expectation and the independence of $\lambda, \boldsymbol{v}, \boldsymbol{v}'$:

$$\mathbb{E}[\tilde{\boldsymbol{v}}] = \mathbb{E}_\lambda \left[\mathbb{E}_{\boldsymbol{v}, \boldsymbol{v}'}[\lambda \boldsymbol{v} + (1 - \lambda)\boldsymbol{v}' \mid \lambda]\right] = \mathbb{E}_\lambda[\lambda \boldsymbol{\mu} + (1 - \lambda)\boldsymbol{\mu}] = \boldsymbol{\mu}.$$

Thus, Mixup preserves the global mean of the distribution. Using the Law of Total Variance:

$$\mathrm{Cov}[\tilde{\boldsymbol{v}}] = \mathbb{E}_\lambda[\mathrm{Cov}[\tilde{\boldsymbol{v}} \mid \lambda]] + \mathrm{Cov}_\lambda[\mathbb{E}[\tilde{\boldsymbol{v}} \mid \lambda]]. \tag{8}$$

The second term vanishes because $\mathbb{E}[\tilde{\boldsymbol{v}} \mid \lambda] = \lambda \boldsymbol{\mu} + (1 - \lambda)\boldsymbol{\mu} = \boldsymbol{\mu}$, which is constant with respect to λ. Thus, $\mathrm{Cov}_\lambda[\boldsymbol{\mu}] = \mathbf{0}$.

Focusing on the first term, conditioned on a fixed λ, the vectors $\boldsymbol{v}$ and $\boldsymbol{v}'$ are independent. Hence:

$$\mathrm{Cov}[\tilde{\boldsymbol{v}} \mid \lambda] = \lambda^2 \Sigma + (1 - \lambda)^2 \Sigma = (\lambda^2 + (1 - \lambda)^2)\Sigma.$$

The covariance of the mixed distribution, denoted as Σ_α, is therefore:

$$\Sigma_\alpha = \mathbb{E}_\lambda[\lambda^2 + (1 - \lambda)^2]\Sigma. \tag{9}$$

For $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$, the distribution is symmetric around 0.5. We have $\mathbb{E}[\lambda] = 0.5$ and $\mathrm{Var}(\lambda) = \frac{1}{4(2\alpha+1)}$. The second moment is given by:

$$\mathbb{E}[\lambda^2] = \mathrm{Var}(\lambda) + \mathbb{E}[\lambda]^2 = \frac{\alpha + 1}{2(2\alpha + 1)}. \tag{10}$$

By symmetry, $\mathbb{E}[(1 - \lambda)^2] = \mathbb{E}[\lambda^2]$. Consequently, the scaling factor c_α is:

$$c_\alpha \triangleq \mathbb{E}[\lambda^2 + (1 - \lambda)^2] = 2\mathbb{E}[\lambda^2] = \frac{\alpha + 1}{2\alpha + 1}. \tag{11}$$

We conclude that:

$$\Sigma_\alpha = c_\alpha \Sigma. \tag{12}$$

Since $\alpha > 0$, it follows that $2\alpha + 1 > \alpha + 1$, implying $c_\alpha < 1$. This proves the stated covariance contraction. If $\mathrm{Tr}(\Sigma) = 0$, the covariance identity still holds, while the strict inequalities become equalities. The independent-pair population result is distinct from finite-batch pairing schemes that can reuse the same example.

2. Distributional Shift. We now quantify the distributional shift using the 2-Wasserstein distance, $W_2(\mathcal{P}, \mathcal{P}_\alpha)$. The Gelbrich bound provides a sharp lower bound for the W_2 distance between any two

distributions with given first two moments. For distributions $\mathcal{P}$ and $\mathcal{P}_\alpha$ with identical means $\boldsymbol{\mu}$ and covariance matrices Σ and Σ_α respectively, the bound is:

$$W_2^2(\mathcal{P}, \mathcal{P}_\alpha) \geq \mathrm{Tr}\left[\Sigma + \Sigma_\alpha - 2(\Sigma^{1/2}\Sigma_\alpha\Sigma^{1/2})^{1/2}\right]. \tag{13}$$

Substituting $\Sigma_\alpha = c_\alpha\Sigma$:

$$(\Sigma^{1/2}\Sigma_\alpha\Sigma^{1/2})^{1/2} = [\Sigma^{1/2}(c_\alpha\Sigma)\Sigma^{1/2}]^{1/2} = \sqrt{c_\alpha}\Sigma. \tag{14}$$

The lower bound simplifies to:

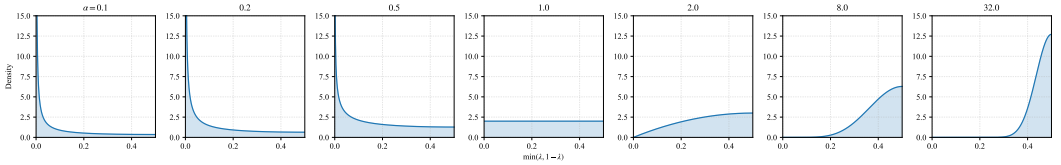
$$W_2^2(\mathcal{P}, \mathcal{P}_\alpha) \geq \mathrm{Tr}\left(\Sigma + c_\alpha\Sigma - 2\sqrt{c_\alpha}\Sigma\right) = (1 - \sqrt{c_\alpha})^2\,\mathrm{Tr}(\Sigma). \tag{15}$$

Taking the square root yields the final inequality:

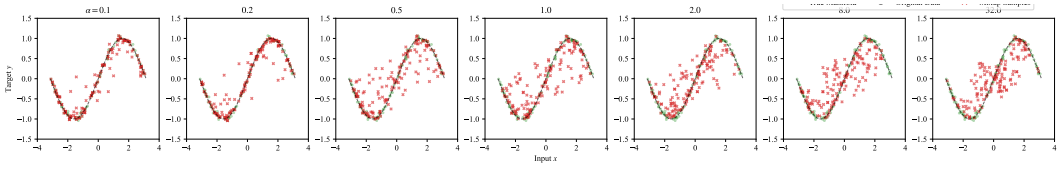
$$W_2(\mathcal{P}, \mathcal{P}_\alpha) \geq (1 - \sqrt{c_\alpha})\sqrt{\mathrm{Tr}(\Sigma)}. \tag{16}$$

Since $c_\alpha < 1$ for all $\alpha > 0$, the term $(1 - \sqrt{c_\alpha})$ is positive. Furthermore, as α increases, c_α decreases monotonically (approaching 0.5 as $\alpha \to \infty$), causing the lower bound of the Wasserstein distance to increase. Visualizations of this contraction and the resulting artifacts are provided in Figure 6. □

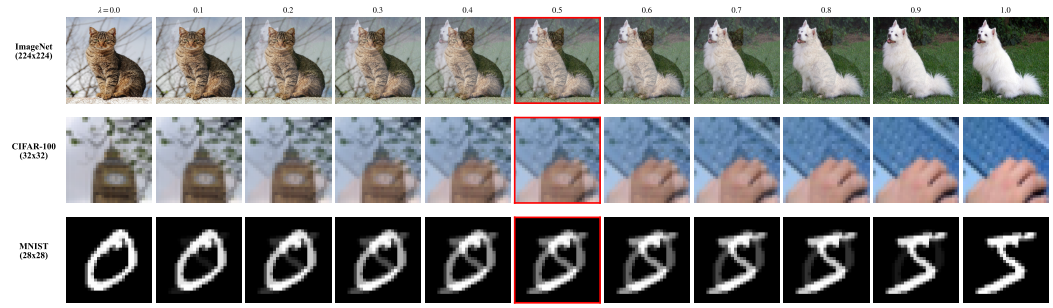
Remark 6 (The Geometry of Contraction). *The identity $\Sigma_\alpha = c_\alpha\Sigma$ quantifies contraction of the global second moments, which we refer to as manifold implosion in the visual examples. Convex interpolation remains in the convex hull of the original support and can simultaneously populate locations absent from that support. Thus, covariance contraction and support expansion can coexist. This geometry motivates examining the regularization retained after removing interpolation, without imposing covariance expansion as a defining requirement of data augmentation.*



(a) Weight Distribution



(b) Manifold Implosion



(c) Visual Artifacts

Figure 6: **Mixup induces distribution contraction and manifold deviation.** (a) Distribution of the minor component weight $\min(\lambda, 1-\lambda)$ across different α. (b) Manifold Implosion on a Toy Regression Task: For a sinusoidal data manifold $y = \sin(x) + \epsilon$, increasing α causes the mixed samples to collapse towards the geometric center. (c) Unnatural Artifacts in High-Dimensional Data: High α values produce "ghostly" superposition artifacts.

A.2 Proof of Lemma 1

We first state the smoothness conditions for the local gradient decomposition.

Assumption 7 (Smoothness)**.** *At fixed parameters and model state, assume that the model Jacobian $\mathbf{J}_{\mathcal{F}}(\boldsymbol{x})$ and the composed criterion gradient $\mathbf{g}_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{y})$ have locally Lipschitz first derivatives on a convex neighborhood containing the expansion point and the endpoints and cross-target evaluations used below. Their values and local derivative bounds are finite on this neighborhood. These assumptions justify quadratic remainders for the displayed expansions; the construction of the retained objective does not require the resulting interpolation residual to be small.*

Lemma 1 (Restated). *Assuming sufficient smoothness (As stated in Assumption 7), the mixed-sample gradient admits the following local decomposition into the retained expression and a quadratic remainder:*

$$\begin{aligned}\mathbf{g}\left(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}; \boldsymbol{\theta}\right) = &\underbrace{\lambda^2 \mathbf{g}\left(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}\right) + (1-\lambda)^2 \mathbf{g}\left(\boldsymbol{x}_j, \boldsymbol{y}_j; \boldsymbol{\theta}\right)}_{\text{Effective Terms}} \\ &+ \underbrace{\lambda(1-\lambda)\left[\mathbf{g}\left(\boldsymbol{x}_i, \boldsymbol{y}_j; \boldsymbol{\theta}\right) + \mathbf{g}\left(\boldsymbol{x}_j, \boldsymbol{y}_i; \boldsymbol{\theta}\right)\right]}_{\text{Perturbation Terms}} + \mathcal{O}\left(\rho^2\right),\end{aligned}$$

Here ρ bounds the distances of both endpoints from the common expansion point in the joint input–target space. The four displayed gradient terms define the retained expression $\mathbf{g}_{ij}^{\mathrm{ret}}$ used in the main text.

Proof. Recall the objective for a single sample $(\boldsymbol{x}, \boldsymbol{y})$: $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \mathcal{C}(\mathcal{F}(\boldsymbol{x}; \boldsymbol{\theta}), \boldsymbol{y})$. Applying the chain rule, the gradient w.r.t. $\boldsymbol{\theta}$ is decomposed as:

$$\mathbf{g}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) \triangleq \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \left[\nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{x}; \boldsymbol{\theta})\right]^{\top} \nabla_{\boldsymbol{h}} \mathcal{C}(\boldsymbol{h}, \boldsymbol{y})\Big|_{\boldsymbol{h}=\mathcal{F}(\boldsymbol{x}; \boldsymbol{\theta})}. \tag{17}$$

Consider a reference point $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$. We perform a first-order Taylor expansion of the Jacobian of the model, $\mathbf{J}_{\mathcal{F}}(\boldsymbol{x}) \triangleq \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{x}; \boldsymbol{\theta})$, and the gradient of the criterion, $\mathbf{g}_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \nabla_{\boldsymbol{h}} \mathcal{C}(\boldsymbol{h}, \boldsymbol{y})$:

$$\mathbf{J}_{\mathcal{F}}(\boldsymbol{x}) = \underbrace{\mathbf{J}_{\mathcal{F}}(\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{x}} \mathbf{J}_{\mathcal{F}}(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})}_{\mathcal{R}(\boldsymbol{x})} + \mathcal{O}(\|\Delta \boldsymbol{x}\|^2), \tag{18}$$

$$\mathbf{g}_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{y}) = \underbrace{\mathbf{g}_{\mathcal{C}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) + \nabla_{\boldsymbol{y}} \mathbf{g}_{\mathcal{C}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})(\boldsymbol{y} - \bar{\boldsymbol{y}}) + \nabla_{\boldsymbol{x}} \mathbf{g}_{\mathcal{C}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})(\boldsymbol{x} - \bar{\boldsymbol{x}})}_{\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})} + \mathcal{O}(\rho^2). \tag{19}$$

where $\rho = \sqrt{\|\Delta \boldsymbol{x}\|^2 + \|\Delta \boldsymbol{y}\|^2}$ represents the perturbation magnitude. For notational brevity, let $\mathcal{R}(\boldsymbol{x})$ and $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ denote the affine approximations (the constant and first order terms) in Eq. (18) and Eq. (19), respectively. Substituting these into Eq. (17) yields:

$$\mathbf{g}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \mathcal{R}(\boldsymbol{x})^{\top} \mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) + \mathcal{O}(\rho^2). \tag{20}$$

Now, consider the Mixup sample $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ formed by the convex combination of $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ with coefficient λ. Due to the linearity of the affine operators $\mathcal{R}(\cdot)$ and $\mathcal{S}(\cdot)$, the gradient at the mixed point can be expanded as:

$$\begin{aligned}\mathbf{g}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}; \boldsymbol{\theta}) &\approx \mathcal{R}(\tilde{\boldsymbol{x}})^{\top} \mathcal{S}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \\ &= \left[\lambda \mathcal{R}(\boldsymbol{x}_i) + (1-\lambda)\mathcal{R}(\boldsymbol{x}_j)\right]^{\top} \left[\lambda \mathcal{S}(\boldsymbol{x}_i, \boldsymbol{y}_i) + (1-\lambda)\mathcal{S}(\boldsymbol{x}_j, \boldsymbol{y}_j)\right] \\ &= \lambda^2 \underbrace{\mathcal{R}(\boldsymbol{x}_i)^{\top} \mathcal{S}(\boldsymbol{x}_i, \boldsymbol{y}_i)}_{\approx \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta})} + (1-\lambda)^2 \underbrace{\mathcal{R}(\boldsymbol{x}_j)^{\top} \mathcal{S}(\boldsymbol{x}_j, \boldsymbol{y}_j)}_{\approx \mathbf{g}(\boldsymbol{x}_j, \boldsymbol{y}_j; \boldsymbol{\theta})} \\ &\quad + \lambda(1-\lambda)\left[\mathcal{R}(\boldsymbol{x}_i)^{\top} \mathcal{S}(\boldsymbol{x}_j, \boldsymbol{y}_j) + \mathcal{R}(\boldsymbol{x}_j)^{\top} \mathcal{S}(\boldsymbol{x}_i, \boldsymbol{y}_i)\right].\end{aligned} \tag{21}$$

Note that:

$$\begin{aligned}\mathcal{R}(\boldsymbol{x}_i)^\top \mathcal{S}(\boldsymbol{x}_j, \boldsymbol{y}_j) &= [\mathbf{J}_\mathcal{F}(\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{x}} \mathbf{J}_\mathcal{F}(\bar{\boldsymbol{x}}) \Delta \boldsymbol{x}_i]^\top \\ &\quad \times [\mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) + \nabla_{\boldsymbol{y}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{y}_j + \nabla_{\boldsymbol{x}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{x}_j] \\ &= \underbrace{\mathcal{R}(\boldsymbol{x}_i)^\top \mathcal{S}(\boldsymbol{x}_i, \boldsymbol{y}_j)}_{\approx \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_j; \boldsymbol{\theta})} + \mathbf{J}_\mathcal{F}(\bar{\boldsymbol{x}})^\top [\nabla_{\boldsymbol{x}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{x}_j - \nabla_{\boldsymbol{x}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{x}_i] \\ &\quad + \underbrace{\Delta \boldsymbol{x}_i^\top \nabla_{\boldsymbol{x}} \mathbf{J}_\mathcal{F}(\bar{\boldsymbol{x}})^\top [\nabla_{\boldsymbol{x}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{x}_j - \nabla_{\boldsymbol{x}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{x}_i]}_{\mathcal{O}(\rho^2)} \\ &\approx \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_j; \boldsymbol{\theta}) + \mathbf{J}_\mathcal{F}(\bar{\boldsymbol{x}})^\top [\nabla_{\boldsymbol{x}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{x}_j - \nabla_{\boldsymbol{x}} \mathbf{g}_\mathcal{C}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \Delta \boldsymbol{x}_i]. \end{aligned} \tag{22}$$

It follows straightforwardly that:

$$\mathcal{R}(\boldsymbol{x}_i)^\top \mathcal{S}(\boldsymbol{x}_j, \boldsymbol{y}_j) + \mathcal{R}(\boldsymbol{x}_j)^\top \mathcal{S}(\boldsymbol{x}_i, \boldsymbol{y}_i) \approx \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_j; \boldsymbol{\theta}) + \mathbf{g}(\boldsymbol{x}_j, \boldsymbol{y}_i; \boldsymbol{\theta}). \tag{23}$$

By substituting this back into the original expansion, we obtain:

$$\mathbf{g}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}; \boldsymbol{\theta}) \approx \lambda^2 \mathbf{g}_i + (1-\lambda)^2 \mathbf{g}_j + \lambda(1-\lambda) \left[\mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_j; \boldsymbol{\theta}) + \mathbf{g}(\boldsymbol{x}_j, \boldsymbol{y}_i; \boldsymbol{\theta})\right]. \tag{24}$$

Reintroducing the quadratic remainder establishes the local expansion. The two input-linear corrections in the cross terms cancel, while products of first-order deviations remain part of the quadratic remainder. □

Exact endpoint interpretation for target-affine gradients. If $\mathbf{g}(\boldsymbol{x}, \boldsymbol{y})$ is affine in $\boldsymbol{y}$ at fixed input, expansion of the mixed target gives the exact identity

$$\mathbf{g}_{ij}^{\text{ret}} = \lambda \mathbf{g}(\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}) + (1-\lambda) \mathbf{g}(\boldsymbol{x}_j, \widetilde{\boldsymbol{y}}). \tag{25}$$

This representation defines the same retained expression for arbitrary endpoint separation. Its discrepancy from the mixed-input gradient is

$$\mathbf{r}_{ij}^{\text{inp}} = \mathbf{g}(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) - \mathbf{g}_{ij}^{\text{ret}}. \tag{26}$$

The local proof above interprets this discrepancy through a joint Taylor expansion. For target-affine gradients, the endpoint identity gives a sharper input-only interpretation without requiring nearby labels. Set $h(\boldsymbol{x}) = \mathbf{g}(\boldsymbol{x}, \widetilde{\boldsymbol{y}})$. If Dh is L_x-Lipschitz on the segment, the input-only residual obeys

$$\|\mathbf{r}_{ij}^{\text{inp}}\|_2 \leq \frac{L_x}{2} \lambda(1-\lambda) \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2. \tag{27}$$

To prove this bound, expand each endpoint around $\widetilde{\boldsymbol{x}}$:

$$h(\boldsymbol{x}_i) = h(\widetilde{\boldsymbol{x}}) + Dh(\widetilde{\boldsymbol{x}})(\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}) + \boldsymbol{e}_i, \qquad \|\boldsymbol{e}_i\| \leq \tfrac{L_x}{2} \|\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}\|^2,$$

and likewise for j. The weighted linear terms cancel because $\lambda(\boldsymbol{x}_i - \widetilde{\boldsymbol{x}}) + (1-\lambda)(\boldsymbol{x}_j - \widetilde{\boldsymbol{x}}) = 0$. Using $\boldsymbol{x}_i - \widetilde{\boldsymbol{x}} = (1-\lambda)(\boldsymbol{x}_i - \boldsymbol{x}_j)$ and $\boldsymbol{x}_j - \widetilde{\boldsymbol{x}} = -\lambda(\boldsymbol{x}_i - \boldsymbol{x}_j)$ yields

$$\|\mathbf{r}_{ij}^{\text{inp}}\| \leq \lambda \|\boldsymbol{e}_i\| + (1-\lambda) \|\boldsymbol{e}_j\| \leq \frac{L_x}{2} \lambda(1-\lambda) \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2. \tag{28}$$

The target dependence at both endpoints is exact; even a bilinear input–target gradient has zero residual. The input curvature in this bound may depend on $\widetilde{\boldsymbol{y}}$. Active purification studies the retained objective after removing this input-nonlinearity term, including regimes where its magnitude is appreciable. Pure target derivatives of order two and higher vanish at fixed input, while the endpoint construction keeps the target effects on the original inputs.

A.3 Proof of Corollary 8

Corollary 8 (Retained Mini-Batch Gradient)**.** *For target-affine criteria, a batch $\mathcal{B}$ of size b, a permutation $f \in S_b$, and a single coefficient λ shared across its pairs, set $\lambda_s = 1 - 2\lambda(1-\lambda)$. Reindexing the retained terms gives*

$$\begin{aligned}\mathbf{g}_\mathcal{B}^{\text{ret}} &= \frac{1}{b} \sum_{i=1}^{b} \left[\lambda_s \mathbf{g}_i + \frac{1-\lambda_s}{2} \sum_{\pi \in \{\{f, f^{-1}\}\}} \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)}) \right], \\ \mathbf{g}_{\mathcal{B}_m} &= \mathbf{g}_\mathcal{B}^{\text{ret}} + \mathbf{r}_\mathcal{B}^{\text{inp}}, \qquad \mathbf{r}_\mathcal{B}^{\text{inp}} = \frac{1}{b} \sum_i \mathbf{r}_{i,f(i)}^{\text{inp}}. \end{aligned} \tag{29}$$

The multiset preserves both contributions, including when $f = f^{-1}$. Under the local assumptions and a common bound ρ on paired deviations, the batch residual is $\mathcal{O}(\rho^2)$. The reindexing proof follows below.

With example-specific coefficients, reindexing also moves their weights; the displayed formula uses the stated shared-coefficient protocol. Model state and other randomness are fixed. Smoothness is needed for the local remainder estimate, while the retained expression and objective remain algebraically defined for standard networks.

Proof. We derive the mini-batch gradient for Mixup with a shared mixing coefficient λ and permutation-based pairing. The empirical gradient over the augmented batch $\mathcal{B}_m$ is:

$$\mathbf{g}_{\mathcal{B}_m} = \frac{1}{b}\sum_{i=1}^{b} \mathbf{g}\left(\widetilde{\boldsymbol{x}}_{i,f(i),\lambda}, \widetilde{\boldsymbol{y}}_{i,f(i),\lambda}; \boldsymbol{\theta}\right). \tag{30}$$

Invoking the single-sample approximation established in **Lemma 1**, we expand this summation:

$$\begin{aligned}
\mathbf{g}_{\mathcal{B}_m} &\approx \frac{1}{b}\sum_{i=1}^{b}\left[\lambda^2\mathbf{g}_i + (1-\lambda)^2\mathbf{g}_{f(i)} + \lambda(1-\lambda)\left[\mathbf{g}\left(\boldsymbol{x}_i, \boldsymbol{y}_{f(i)}; \boldsymbol{\theta}\right) + \mathbf{g}\left(\boldsymbol{x}_{f(i)}, \boldsymbol{y}_i; \boldsymbol{\theta}\right)\right]\right] \\
&= \frac{1}{b}\Bigg[\sum_{i=1}^{b}\lambda^2\mathbf{g}_i + \sum_{j=1}^{b}(1-\lambda)^2\mathbf{g}_j \\
&\qquad + \sum_{i=1}^{b}\lambda(1-\lambda)\mathbf{g}\left(\boldsymbol{x}_i, \boldsymbol{y}_{f(i)}; \boldsymbol{\theta}\right) + \sum_{j=1}^{b}\lambda(1-\lambda)\mathbf{g}\left(\boldsymbol{x}_j, \boldsymbol{y}_{f^{-1}(j)}; \boldsymbol{\theta}\right)\Bigg] \\
&= \frac{1}{b}\sum_{i=1}^{b}\left[\left(\lambda^2 + (1-\lambda)^2\right)\mathbf{g}_i + \lambda(1-\lambda)\left[\mathbf{g}\left(\boldsymbol{x}_i, \boldsymbol{y}_{f(i)}; \boldsymbol{\theta}\right) + \mathbf{g}\left(\boldsymbol{x}_i, \boldsymbol{y}_{f^{-1}(i)}; \boldsymbol{\theta}\right)\right]\right],
\end{aligned} \tag{31}$$

where f^{-1} is the inverse permutation of f. Let $\lambda_s = 1 - 2\lambda(1-\lambda)$, we observe that:

$$\begin{aligned}
\lambda^2 + (1-\lambda)^2 &= 1 + 2\lambda^2 - 2\lambda = \lambda_s, \\
\lambda(1-\lambda) &= \frac{1}{2}\left(2\lambda - 2\lambda^2\right) = \frac{1}{2}\left(1 - \lambda_s\right).
\end{aligned} \tag{32}$$

Thus,

$$\mathbf{g}_{\mathcal{B}_m} \approx \frac{1}{b}\sum_{i=1}^{b}\left[\lambda_s\mathbf{g}_i + (1-\lambda_s)\sum_{\pi\in\{\{f, f^{-1}\}\}}\frac{1}{2}\mathbf{g}\left(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)}; \boldsymbol{\theta}\right)\right]. \tag{33}$$

For arbitrary endpoint separation, define $\mathbf{g}^{\text{ret}}_{\mathcal{B}} = b^{-1}\sum_i \mathbf{g}^{\text{ret}}_{i,f(i)}$ and $\mathbf{r}^{\text{inp}}_{\mathcal{B}} = b^{-1}\sum_i \mathbf{r}^{\text{inp}}_{i,f(i)}$. Then the exact decomposition is $\mathbf{g}_{\mathcal{B}_m} = \mathbf{g}^{\text{ret}}_{\mathcal{B}} + \mathbf{r}^{\text{inp}}_{\mathcal{B}}$. Under the local assumptions, the final term is $\mathcal{O}(\rho^2)$ when ρ bounds all paired deviations. The shared coefficient is essential to the displayed reindexing: with example-specific coefficients, the weights must also be reindexed. □

A.4 Purification for Batch-Dependent, Permutation-Equivariant Predictors

The endpoint interpretation extends to predictors whose outputs depend on the entire batch. This includes training-mode batch normalization: batch statistics are recomputed at each input batch and remain part of the differentiated forward map. Let $\mathbf{X} \in \mathbb{R}^{b\times d}$ and $\mathbf{Y} \in \mathbb{R}^{b\times K}$ collect inputs and targets by rows, and define

$$\mathbf{Z} = \mathcal{F}^{\text{batch}}_{\boldsymbol{\theta}}(\mathbf{X}), \qquad \mathcal{L}_{\mathcal{B}}(\mathbf{X}, \mathbf{Y}) = \frac{1}{b}\sum_{i=1}^{b}\mathcal{C}(\mathbf{Z}_i, \mathbf{Y}_i), \qquad \mathbf{G}(\mathbf{X}, \mathbf{Y}) = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{B}}(\mathbf{X}, \mathbf{Y}).$$

All compared gradients use the same parameters and persistent model state. Any other fixed randomness is assumed to preserve the permutation equivariance stated below. The target-affine condition is imposed on $\mathbf{G}$, rather than on the scalar loss: softmax cross-entropy and squared error both satisfy it, although squared error is not affine in its target.

Proposition 9 (Batch-Matrix Purification)**.** *Let* $\mathbf{P}$ *be a* $b \times b$ *permutation matrix and suppose, as an identity in the trainable parameters, that*

$$\mathcal{F}_{\boldsymbol{\theta}}^{\mathrm{batch}}(\mathbf{PX}) = \mathbf{P}\mathcal{F}_{\boldsymbol{\theta}}^{\mathrm{batch}}(\mathbf{X}).$$

Assume the batch criterion is invariant under simultaneous row permutation of predictions and targets, and $\mathbf{G}(\mathbf{X}, \mathbf{Y})$ *is affine in* $\mathbf{Y}$ *at fixed* $\mathbf{X}$*. For a shared coefficient* $\lambda \in [0, 1]$*, set*

$$\widetilde{\mathbf{X}} = \lambda\mathbf{X} + (1-\lambda)\mathbf{PX}, \qquad \widetilde{\mathbf{Y}} = \lambda\mathbf{Y} + (1-\lambda)\mathbf{PY}, \qquad \lambda_s = \lambda^2 + (1-\lambda)^2.$$

Then the endpoint expression at the common mixed target satisfies the exact identity

$$\begin{aligned}\mathbf{G}_{\mathcal{B}}^{\mathrm{ret}} &\triangleq \lambda\mathbf{G}(\mathbf{X}, \widetilde{\mathbf{Y}}) + (1-\lambda)\mathbf{G}(\mathbf{PX}, \widetilde{\mathbf{Y}}) \\ &= \lambda_s\mathbf{G}(\mathbf{X}, \mathbf{Y}) + \lambda(1-\lambda)\left[\mathbf{G}(\mathbf{X}, \mathbf{PY}) + \mathbf{G}(\mathbf{X}, \mathbf{P}^{-1}\mathbf{Y})\right]. \end{aligned} \tag{34}$$

Consequently, $\mathbf{G}_{\mathcal{B}}^{\mathrm{ret}}$ *is the gradient of the paired objective*

$$\mathcal{S}_{\mathcal{B}}^{\mathrm{pair}} = \lambda_s\mathcal{L}_{\mathcal{B}}(\mathbf{X}, \mathbf{Y}) + \frac{1-\lambda_s}{2}\left[\mathcal{L}_{\mathcal{B}}(\mathbf{X}, \mathbf{PY}) + \mathcal{L}_{\mathcal{B}}(\mathbf{X}, \mathbf{P}^{-1}\mathbf{Y})\right]. \tag{35}$$

The discrepancy from the full mixed-input gradient is exactly

$$\mathbf{R}_{\mathcal{B}}^{\mathrm{inp}} = \mathbf{G}(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}) - \lambda\mathbf{G}(\mathbf{X}, \widetilde{\mathbf{Y}}) - (1-\lambda)\mathbf{G}(\mathbf{PX}, \widetilde{\mathbf{Y}}). \tag{36}$$

If the input derivative of $\mathbf{H}(\mathbf{X}') \triangleq \mathbf{G}(\mathbf{X}', \widetilde{\mathbf{Y}})$ *is* L_X*-Lipschitz along the segment between* $\mathbf{X}$ *and* $\mathbf{PX}$*, using the Frobenius norm on input matrices and Euclidean norm on parameter gradients, then*

$$\left\|\mathbf{R}_{\mathcal{B}}^{\mathrm{inp}}\right\|_2 \leq \frac{L_X}{2}\lambda(1-\lambda)\left\|\mathbf{X} - \mathbf{PX}\right\|_F^2. \tag{37}$$

Proof. Permutation equivariance and simultaneous-permutation invariance of the criterion give

$$\mathcal{L}_{\mathcal{B}}(\mathbf{PX}, \mathbf{Y}') = \mathcal{L}_{\mathcal{B}}(\mathbf{X}, \mathbf{P}^{-1}\mathbf{Y}')$$

for every target matrix $\mathbf{Y}'$. Differentiating this identity at the same parameters yields

$$\mathbf{G}(\mathbf{PX}, \mathbf{Y}') = \mathbf{G}(\mathbf{X}, \mathbf{P}^{-1}\mathbf{Y}'). \tag{38}$$

In particular, $\mathbf{P}^{-1}\widetilde{\mathbf{Y}} = \lambda\mathbf{P}^{-1}\mathbf{Y} + (1-\lambda)\mathbf{Y}$. Affineness in the target therefore gives

$$\lambda\mathbf{G}(\mathbf{X}, \widetilde{\mathbf{Y}}) = \lambda^2\mathbf{G}(\mathbf{X}, \mathbf{Y}) + \lambda(1-\lambda)\mathbf{G}(\mathbf{X}, \mathbf{PY}),$$
$$(1-\lambda)\mathbf{G}(\mathbf{PX}, \widetilde{\mathbf{Y}}) = \lambda(1-\lambda)\mathbf{G}(\mathbf{X}, \mathbf{P}^{-1}\mathbf{Y}) + (1-\lambda)^2\mathbf{G}(\mathbf{X}, \mathbf{Y}).$$

Adding these equations proves Eq. (34). Differentiating Eq. (35) proves the objective realization. Both cross-target terms retain their multiplicity when $\mathbf{P} = \mathbf{P}^{-1}$.

For the residual bound, expand $\mathbf{H}$ around $\widetilde{\mathbf{X}}$. The Lipschitz derivative assumption means

$$\|D\mathbf{H}(\mathbf{A}) - D\mathbf{H}(\mathbf{B})\|_{F\to 2} \leq L_X\|\mathbf{A} - \mathbf{B}\|_F$$

on the segment. The first-order remainder at either endpoint $\mathbf{A}$ is consequently bounded by $L_X\|\mathbf{A} - \widetilde{\mathbf{X}}\|_F^2/2$. The weighted linear terms cancel because $\lambda(\mathbf{X} - \widetilde{\mathbf{X}}) + (1-\lambda)(\mathbf{PX} - \widetilde{\mathbf{X}}) = 0$. Thus

$$\begin{aligned}\|\mathbf{R}_{\mathcal{B}}^{\mathrm{inp}}\|_2 &\leq \frac{L_X}{2}\left[\lambda\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2 + (1-\lambda)\|\mathbf{PX} - \widetilde{\mathbf{X}}\|_F^2\right] \\ &= \frac{L_X}{2}\lambda(1-\lambda)\|\mathbf{X} - \mathbf{PX}\|_F^2,\end{aligned}$$

which proves Eq. (37). □

Input purification with batch normalization. For a feature coordinate, training-mode batch normalization computes its mean and variance over the rows. Row permutation leaves both statistics unchanged, so normalization is permutation equivariant. The statistics at $\widetilde{\mathbf{X}}$ can differ from those at $\mathbf{X}$; their effect is included in $\mathbf{R}_{\mathcal{B}}^{\mathrm{inp}}$. No freezing of batch statistics is needed for the exact identities. Target interpolation is represented exactly at the two endpoint batches, and active purification removes the input Jensen gap, including changes induced by batch-dependent normalization along the interpolation path. The construction permits arbitrary target separation and remains defined when this gap is appreciable. The quadratic bound additionally requires its stated input smoothness condition. Other stochastic components are covered when their conditional realizations preserve the same permutation equivariance.

Batch-conditioned Jacobians in the covariance analysis. For a batch-dependent predictor, define

$$\mathbf{J}_i^{\mathcal{B}} = \nabla_{\boldsymbol{\theta}} \left[\mathcal{F}_{\boldsymbol{\theta}}^{\text{batch}}(\mathbf{X}) \right]_i \in \mathbb{R}^{K \times d_\theta}.$$

The per-example contribution to the CE gradient is still $\mathbf{g}_i = (\mathbf{J}_i^{\mathcal{B}})^\top (\boldsymbol{o}_i - \boldsymbol{y}_i)$, but the conditioning includes the *entire input batch*, parameters, and fixed model randomness. In particular, $\mathbf{J}_i^{\mathcal{B}}$ includes differentiation through batch statistics; it is not a Jacobian computed with those statistics detached. With $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}_r]$, $\mathbf{V}_r = \text{Cov}(\boldsymbol{y}_r)$, $w = 1 - \lambda_s$, $\varepsilon = \mathbb{E}[w]$, $v_s = \text{Var}(w)$, and $c_{in} = \mathbb{E}[w^2]$, set $\mathbf{d}_i^{\mathcal{B}} = (\mathbf{J}_i^{\mathcal{B}})^\top (\boldsymbol{y}_i - \boldsymbol{\mu})$. At this conditioning, independent coefficient–target pairs across rows give

$$\text{Cov}\left(\frac{1}{b} \sum_i \boldsymbol{\zeta}_i \,\middle|\, \mathcal{B}, \boldsymbol{\theta} \right) = \frac{1}{b^2} \sum_i \left[v_s \mathbf{d}_i^{\mathcal{B}} (\mathbf{d}_i^{\mathcal{B}})^\top + c_{in} (\mathbf{J}_i^{\mathcal{B}})^\top \mathbf{V}_r \mathbf{J}_i^{\mathcal{B}} \right].$$

For one shared w and independently sampled targets, this becomes

$$v_s \overline{\mathbf{d}}_{\mathcal{B}} \overline{\mathbf{d}}_{\mathcal{B}}^\top + \frac{c_{in}}{b^2} \sum_i (\mathbf{J}_i^{\mathcal{B}})^\top \mathbf{V}_r \mathbf{J}_i^{\mathcal{B}}, \qquad \overline{\mathbf{d}}_{\mathcal{B}} = \frac{1}{b} \sum_i \mathbf{d}_i^{\mathcal{B}}.$$

Here $\boldsymbol{\zeta}_i = (w_i - \varepsilon)\mathbf{d}_i^{\mathcal{B}} - w_i (\mathbf{J}_i^{\mathcal{B}})^\top (\boldsymbol{y}_{r,i} - \boldsymbol{\mu})$, with $w_i = w$ in the shared-coefficient case. Thus, batch-dependent forward maps preserve the conditional covariance calculation once the Jacobians and conditioning are interpreted consistently. Averaging over data batches must use their actual sampling law, and the paired permutation construction retains its joint target dependence.

B THE SOMBRERO LOSS FRAMEWORK

The name Sombrero evokes retaining a learning signal while exploring how structured perturbations can guide training toward broad regions of the landscape.

B.1 EXACT REALIZATION OF THE RETAINED FIRST-ORDER EXPRESSION

The paired Sombrero objective exactly implements the retained gradient expression obtained from Mixup. We establish the identity conditional on the same batch, shared mixing coefficient, and permutation, and then distinguish it from independent perturbation-target sampling.

Proposition 10 (Exact Paired Realization)**.** *For any given mini-batch $\mathcal{B}$ and a permutation $f \in S_b$, consider the Sombrero Loss $\mathcal{SL}_{\mathcal{B}}^{(1)}$ under the specific sampling strategy defined as:*

$$\mathcal{SL}_{\mathcal{B}}^{(1)} = \frac{1}{2b} \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{B}} \sum_{\pi \in \{\{f, f^{-1}\}\}} \mathcal{SL}\left(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{y}_{\pi(i)}; \boldsymbol{\theta} \right), \tag{39}$$

where the Sombrero Loss for a single sample is given by:

$$\mathcal{SL}\left(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{y}_r; \boldsymbol{\theta} \right) = \lambda_s \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta} \right) + (1 - \lambda_s) \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{y}_r; \boldsymbol{\theta} \right), \tag{40}$$

and f^{-1} denotes the inverse permutation of f. Then $\nabla_{\boldsymbol{\theta}} \mathcal{SL}_{\mathcal{B}}^{(1)} = \mathbf{g}_{\mathcal{B}}^{\text{ret}}$: its gradient is exactly the retained batch expression for the same $\mathcal{B}$, λ, and f.

Proof. Since λ_s is independent of the trainable parameters, differentiation can be taken term by term:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathcal{SL}_{\mathcal{B}}^{(1)} &= \frac{1}{2b} \sum_{i=1}^{b} \sum_{\pi \in \{\{f, f^{-1}\}\}} \left[\lambda_s \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_i) + (1 - \lambda_s) \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)}) \right] \\ &= \frac{1}{b} \sum_{i=1}^{b} \left[\lambda_s \mathbf{g}_i + \frac{1 - \lambda_s}{2} \left\{ \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_{f(i)}) + \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_{f^{-1}(i)}) \right\} \right]. \end{aligned} \tag{41}$$

The first term is counted twice in the multiset sum. Substituting $\lambda_s = \lambda^2 + (1-\lambda)^2$ and $(1-\lambda_s)/2 = \lambda(1-\lambda)$ recovers the retained expression in Eq. (31). This proves the identity, including permutations satisfying $f = f^{-1}$. □

Independent-target extension. For a fixed batch and model state, let $f \sim \mathcal{U}(S_b)$ independently of the shared coefficient. Each forward and inverse target then has the batch empirical mean; for target-affine gradients, replacing these targets with independent draws from the same empirical prior preserves the gradient mean averaged over the pairing and coefficient. Cross-example dependence, and hence batch covariance, can change. The practical Uniform variant samples a one-hot $\boldsymbol{y}_r$ independently of the source example and the retention coefficient. This keeps the loss construction and retention law while changing the target marginal and cross-example dependence. The paired identity describes the construction's connection to Mixup; the mean and covariance analysis below characterizes the chosen perturbation-target process directly.

B.2 Statistical Properties of λ_s

In this section, we provide a statistical characterization of the stochastic coefficient λ_s, defined as $\lambda_s \triangleq 1 - 2\lambda(1-\lambda)$. While the Sombrero Loss admits general mixing distributions, we restrict our analysis to the standard symmetric setting where $\lambda \sim \text{Beta}(\alpha, \alpha)$, as this covers the vast majority of applications.

B.2.1 Probability Density Function

The transformation $\lambda_s = 2(\lambda - 0.5)^2 + 0.5$ folds the support $\lambda \in [0, 1]$ onto $\lambda_s \in [0.5, 1]$. By applying the change of variables method and exploiting the symmetry of the Beta distribution about 0.5, we derive the Probability Density Function (PDF) of λ_s as:

$$f_S(\lambda_s) = \frac{\left(\frac{1-\lambda_s}{2}\right)^{\alpha-1}}{B(\alpha,\alpha)\sqrt{2\lambda_s - 1}}, \quad \text{for } \lambda_s \in (0.5, 1]. \tag{42}$$

Here, $B(\cdot,\cdot)$ represents the Beta function.

Geometric Singularity vs. Probability Mass.

1. **Singularity at 0.5:** As $\lambda_s \to 0.5$, the density $f_S(\lambda_s) \to \infty$ because the inverse Jacobian term $(2\lambda_s - 1)^{-1/2}$ diverges at the vertex of the quadratic fold.

2. **Mass Concentration:** Despite this singularity, the probability mass distribution depends heavily on α. As shown in Figure 7, as α increases, the probability mass becomes more concentrated at 0.5, which means the Perturbation Terms of Sombrero Loss is enhanced.

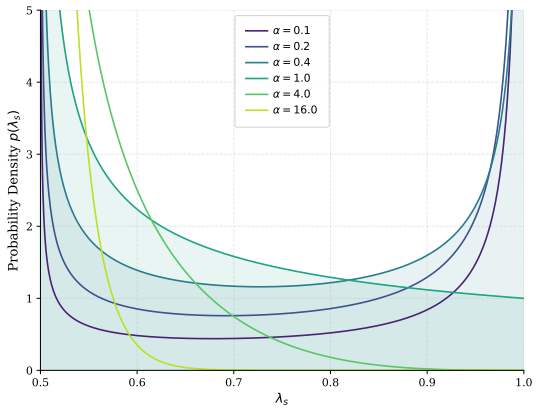


Figure 7: Distribution of λ_s with varying α.

B.2.2 Moments and Stability Bounds

Using the moments of the symmetric Beta distribution, we derive closed-form expressions for the mean and variance of λ_s.

$$\mu_{\lambda_s} = 1 - \frac{\alpha}{2\alpha + 1}, \tag{43}$$

$$\sigma^2_{\lambda_s} = \frac{\alpha}{(2\alpha + 1)^2(2\alpha + 3)}. \tag{44}$$

Bounded Variance. To verify the stability of the induced noise, we analyze the extrema of Eq. (44). Analytical maximization with respect to α demonstrates that the variance is bounded:

$$\sup_{\alpha>0} \sigma^2_{\lambda_s} = \frac{11\sqrt{33} - 59}{128} \approx 0.0327 \quad \left(\text{attained at } \alpha = \frac{\sqrt{33} - 3}{8} \approx 0.343\right). \tag{45}$$

This bounds the variance of the retention coefficient for every $\alpha > 0$. Parameter-gradient variance additionally depends on the model Jacobian and the perturbation-target distribution. For completeness, the second moment entering the intrinsic-noise covariance is $c_{in} = \mathbb{E}[(1 - \lambda_s)^2] = \alpha(\alpha + 1)/[(2\alpha + 1)(2\alpha + 3)]$.

B.3 Variants of Sombrero Loss

B.3.1 Centered Sombrero Loss

Proposition 2 identifies Sombrero's conditional mean gradient with that of label smoothing. To separate this mean target-smoothing effect from the conditional stochastic perturbations, we construct Centered Sombrero Loss (Cen-Sombrero):

$$\mathcal{SL}_{cen} = \mathcal{SL} - \mathbb{E}_{\lambda_s, \boldsymbol{y}_r}[\mathcal{SL}] + \mathcal{L}. \tag{46}$$

This construction restores the ERM mean objective, $\mathbb{E}[\mathcal{SL}_{cen}] = \mathcal{L}$, while preserving Sombrero's conditional algorithmic gradient covariance at fixed example and parameters. For a batch, the same statement holds when centering uses the expectation conditional on that batch and its sampling protocol. The resulting mean field, data-sampling covariance, and training trajectory can differ from those of standard Sombrero.

Substituting the definition of Sombrero Loss into Eq. (46) expands to:

$$\mathcal{SL}_{cen} = (1 + \lambda_s - \mu_{\lambda_s})\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) + (1 - \lambda_s)\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}_r; \boldsymbol{\theta}) - (1 - \mu_{\lambda_s})\mathbb{E}_{\boldsymbol{y}_r}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}_r; \boldsymbol{\theta})]. \tag{47}$$

Drawing on the derivations in Appendix C.1, for a criterion whose gradient $\nabla_{\boldsymbol{h}}\mathcal{C}(\boldsymbol{h}, \boldsymbol{y})$ is affine with respect to the target $\boldsymbol{y}$, we can bypass output-level computations. Instead, we project Eq. (47) into a target-mixing formulation that preserves identical gradients, significantly simplifying implementation:

$$\boldsymbol{y}^{(\mathcal{SL}_{cen})} = (1 + \lambda_s - \mu_{\lambda_s})\boldsymbol{y} + (1 - \lambda_s)\boldsymbol{y}_r - (1 - \mu_{\lambda_s})\boldsymbol{\mu}_{y_r}. \tag{48}$$

For probability-valued source and perturbation targets, this representation has unit total mass but can contain negative entries. It is therefore a signed target representation that expresses the objective's gradient algebraically.

B.3.2 Decoupled Sombrero Loss

As analyzed in Appendix C.5, the first-order smoothing effect and the second-order noise effect in the standard Sombrero Loss are intrinsically coupled via the single hyperparameter α. This coupling can restrict the capacity to simultaneously optimize both mechanisms. To circumvent this, we propose the Decoupled Sombrero Loss (Dec-Sombrero), which introduces α_1 and α_2 to govern the smoothing and noise effects independently:

$$\mathcal{SL}_{dec} = \mathbb{E}_{\lambda_s, \boldsymbol{y}_r}[\mathcal{SL}(\alpha_1) - \mathcal{SL}(\alpha_2)] + \mathcal{SL}(\alpha_2). \tag{49}$$

Under the standard settings of Proposition 3, we can derive a coefficient-matching calibration within the prediction model used for the curvature analysis. We enforce the condition $\varepsilon(\alpha_1) = \varepsilon^*$ to preserve

the desired smoothing target while achieving structural alignment with the Hessian. Substituting $\varepsilon(\alpha_1) = \frac{\alpha_1}{2\alpha_1+1}$ and the calibrated target $\varepsilon^* = 1 - \frac{1}{(2\alpha_2+1)(\alpha_2+1)}$ from Appendix C.5, we obtain the explicit algebraic constraint:

$$\begin{aligned} \frac{\alpha_1}{2\alpha_1+1} = \varepsilon^* &= 1 - \frac{1}{(2\alpha_2+1)(\alpha_2+1)} \\ \implies \alpha_1 &= \frac{\alpha_2(2\alpha_2+3)}{1-2\alpha_2^2-3\alpha_2}, \quad \text{where } \alpha_2 \in (0, \underbrace{\frac{\sqrt{17}-3}{4}}_{\approx 0.281}). \end{aligned} \tag{50}$$

The upper bound for α_2 ensures the positivity of the denominator, guaranteeing a valid hyperparameter $\alpha_1 > 0$.

This mapping selects α_1 from one free parameter α_2 within the stated range. It separates the mean and noise controls and matches their modeled curvature coefficients under the approximation used to define ε^*; it is not an optimization theorem for generalization performance. The corresponding target-mixing formulation is derived as:

$$\begin{aligned} \boldsymbol{y}^{(\mathcal{SL}_{dec})} &= [\mu_{\lambda_s}(\alpha_1) - \mu_{\lambda_s}(\alpha_2) + \lambda_s]\, \boldsymbol{y} + [\mu_{\lambda_s}(\alpha_2) - \mu_{\lambda_s}(\alpha_1)]\, \boldsymbol{\mu}_{y_r} + (1-\lambda_s)\boldsymbol{y}_r \\ &= \left[\lambda_s - \frac{\alpha_2(\alpha_2+2)}{(2\alpha_2+1)(\alpha_2+1)}\right] \boldsymbol{y} + \frac{\alpha_2(\alpha_2+2)}{(2\alpha_2+1)(\alpha_2+1)} \boldsymbol{\mu}_{y_r} + (1-\lambda_s)\boldsymbol{y}_r. \end{aligned} \tag{51}$$

Crucially, to ensure the injected noise aligns closely with the intended geometric structure, the stochastic coefficient λ_s in Eq. (51) must be drawn from the distribution parameterized solely by α_2, i.e., derived from $\lambda \sim$ Beta(α_2, α_2).

B.4 Input and Supervision Constructions Across Regularizers

Table 7 compares how the methods construct inputs and supervision, and how these choices relate to classification, detection, and sequence prediction. Sombrero uses the original input with losses under the observed and perturbation targets, allowing task-specific targets to be defined through existing loss components. The table distinguishes an implemented task instance from the additional constructions needed for an extension.

Table 7: **Input and supervision constructions across regularizers.** The cited formulations differ in their input operations, target construction, and task requirements. Task notes distinguish demonstrated uses, transfer of pretrained representations, and the constructions needed for additional applications.

Method	Input / feature operation	Target or loss construction	Task instances and extension requirements
Mixup (Zhang et al., 2018)	Interpolate paired inputs.	Interpolate paired class targets with the same coefficient.	Classification; detection and Seq2Seq extensions must specify compatible input mixing and structured or token-level supervision.
CutMix (Yun et al., 2019)	Replace an image patch using another image.	Weight class targets by patch area.	Classification; the cited work also transfers pretrained representations to detection and captioning. Training-time task extensions require supervision consistent with the mixed regions.
PuzzleMix (Kim et al., 2020)	Mix and transport image regions using saliency and local statistics.	Interpolate class targets according to the mixing ratio.	Image classification; structured or sequence applications additionally require task-appropriate saliency, mixing, and supervision rules.
TokenMix (Liu et al., 2022)	Mix image regions at the visual-token level.	Form content-aware class targets using teacher activation maps.	Image classification with vision transformers; detection or Seq2Seq use requires a corresponding structure or token-alignment and supervision rule.
LossMix (Vu et al., 2024)	Interpolate paired inputs.	Weight losses under the original targets; no interpolated target representation is required.	Classification-compatible loss construction and demonstrated object detection; sequence use still requires an appropriate input-mixing rule.
Manifold Mixup (Verma et al., 2019)	Interpolate paired hidden representations.	Interpolate their class targets.	Classification; detection and Seq2Seq extensions require compatible feature representations, alignment, and supervision.
LS / RLS (Szegedy et al., 2016; dos Santos & Papa, 2024)	Keep the original inputs and features.	Use fixed smoothed targets (LS) or stochastic soft targets (RLS).	Classification; token-level LS is the translation baseline here. Applying either rule to detection requires selecting the supervised loss components and targets.
Sombrero	Keep the original inputs and features.	Combine the original-target loss with a perturbation-target loss; the perturbation prior can be chosen separately.	Demonstrated here in classification, detection, and Seq2Seq. Detection uses assignment-based targets; translation uses token-level targets with unchanged inputs.

For cross-entropy, soft-target losses can also be evaluated as weighted sums of losses. The distinction between an explicit target representation and a loss-level implementation therefore does not, by itself, determine the regularization mechanism or task scope.

C Theoretical Analysis of Sombrero Loss

C.1 Proof of Proposition 2: Expectation Equivalence

Proposition 2 (Restated). *Consider a criterion whose gradient $\nabla_{\boldsymbol{h}}\mathcal{C}(\boldsymbol{h}, \boldsymbol{y})$ is affine with respect to the target $\boldsymbol{y}$. If the perturbation target is independent of λ_s and has uniform mean ($\boldsymbol{\mu}_{y_r} = \frac{1}{K}\mathbf{1}_K$), and the hyperparameters satisfy $\varepsilon = 1 - \mu_{\lambda_s} = \frac{\alpha}{2\alpha+1}$, then the expected gradient of Sombrero Loss $\mathcal{SL}$ is identical to that of Label Smoothing (LS) with parameter ε:*

$$\mathbb{E}_{\lambda_s, \boldsymbol{y}_r}\left[\mathbf{g}_i^{(\mathcal{SL})}\right] = \mathbf{g}_i^{(LS)} \implies \mathbb{E}_{i,\lambda_s,\boldsymbol{y}_r}\left[\mathbf{g}_i^{(\mathcal{SL})}\right] = \mathbb{E}_i\left[\mathbf{g}_i^{(LS)}\right].$$

Proof. Recall the chain rule, the parameter gradient is decomposed as $\left[\nabla_{\boldsymbol{\theta}}\mathcal{F}(\boldsymbol{x};\boldsymbol{\theta})\right]^\top \nabla_{\boldsymbol{h}}\mathcal{C}(\boldsymbol{h}, \boldsymbol{y})$. The proposition premises that $\nabla_{\boldsymbol{h}}\mathcal{C}(\boldsymbol{h}, \boldsymbol{y})$ is affine w.r.t. the target $\boldsymbol{y}$. This implies that for any $\boldsymbol{y}$, the gradient can be decomposed as:

$$\mathbf{g} = \mathbf{A}(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{y} + \mathbf{b}(\boldsymbol{x};\boldsymbol{\theta}), \tag{52}$$

where $\mathbf{A}$ and $\mathbf{b}$ are matrix and vector functions dependent on the input and parameters but independent of the target $\boldsymbol{y}$.

1. Expected Gradient of Sombrero Loss: Recalling the definition of Sombrero Loss, the parameter gradient generated by a single sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is:

$$\mathbf{g}_i^{(\mathcal{SL})} = \lambda_s \mathbf{g}_i + (1 - \lambda_s)\mathbf{g}_{r,i}, \tag{53}$$

where $\mathbf{g}_i = \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta})$ is derived from the Effective Terms, and $\mathbf{g}_{r,i} = \mathbf{g}(\boldsymbol{x}_i, \boldsymbol{y}_r; \boldsymbol{\theta})$ is induced by the Perturbation Terms. Taking the expectation over the random variables λ_s and $\boldsymbol{y}_r$:

$$\begin{aligned}
\mathbb{E}_{\lambda_s, \boldsymbol{y}_r}\left[\mathbf{g}_i^{(\mathcal{SL})}\right] &= \mathbb{E}[\lambda_s]\mathbf{g}_i + (1 - \mathbb{E}[\lambda_s])\mathbb{E}_{\boldsymbol{y}_r}[\mathbf{g}_{r,i}] \\
&= \mu_{\lambda_s}(\mathbf{A}\boldsymbol{y}_i + \mathbf{b}) + (1 - \mu_{\lambda_s})(\mathbf{A}\mathbb{E}[\boldsymbol{y}_r] + \mathbf{b}) \\
&= \mathbf{A}\left[\mu_{\lambda_s}\boldsymbol{y}_i + (1 - \mu_{\lambda_s})\boldsymbol{\mu}_{y_r}\right] + \mathbf{b},
\end{aligned} \tag{54}$$

(Note: λ_s and $\boldsymbol{y}_r$ are mutually independent).

2. Gradient of LS: LS constructs a smoothed target $\boldsymbol{y}^{(LS)} = (1-\varepsilon)\boldsymbol{y}_i + \varepsilon\boldsymbol{\mu}_{y_r}$ (typically $\boldsymbol{\mu}_{y_r} = \frac{1}{K}\mathbf{1}$). The gradient is:

$$\mathbf{g}_i^{(LS)} = \mathbf{A}\left[(1 - \varepsilon)\boldsymbol{y}_i + \varepsilon\boldsymbol{\mu}_{y_r}\right] + \mathbf{b}. \tag{55}$$

3. Equivalence Condition: Comparing Eq. (54) and Eq. (55), matching the target coefficients guarantees identical gradients: choose $\varepsilon = 1 - \mu_{\lambda_s}$. This is a sufficient identity for all target-affine criteria, including cases with a degenerate sample Jacobian. Substituting $\mu_{\lambda_s} = 1 - \frac{\alpha}{2\alpha+1}$ yields the condition $\varepsilon = \frac{\alpha}{2\alpha+1}$. □

Remark 11 (Common Criterion Functions). *For the following criterion functions, the affine assumption holds naturally:*

- ***Cross-Entropy with Softmax (Softmax CE):*** $\mathbf{g} = \mathbf{J}^\top(\boldsymbol{o} - \boldsymbol{y})$. *Here,* $\mathbf{A} = -\mathbf{J}^\top$ *and* $\mathbf{b} = \mathbf{J}^\top\boldsymbol{o}$, *where* $\mathbf{J} = \nabla_{\boldsymbol{\theta}}\mathcal{F}(\boldsymbol{x};\boldsymbol{\theta})$ *represents the Jacobian matrix, and* $\boldsymbol{o}$ *denotes the predicted probabilities.*
- ***Mean Squared Error (MSE):*** *for* $\mathcal{C}(\boldsymbol{h}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{h} - \boldsymbol{y}\|^2$, $\mathbf{g} = \mathbf{J}^\top(\boldsymbol{h} - \boldsymbol{y})$, *where* $\mathbf{J}$ *is the prediction Jacobian. The same affine structure applies; the factor of two is restored for an MSE convention without* $1/2$.

Thus, Proposition 2 applies directly to these criterion functions.

C.2 Proof of Proposition 3: Structure of Intrinsic Noise

Proposition 3 (Restated). *For Softmax CE with one-hot source labels and uniformly sampled one-hot perturbation targets independent of λ_s, the average conditional per-example Intrinsic Noise covariance at fixed parameters decomposes as:*

$$\Sigma_{in} = \frac{c_{in}}{K}\left(\mathbf{U}_{\text{base}} + \gamma_{in}\mathbf{U}_{\text{cls}}\right).$$

Here, $\mathbf{U}_{base}$ represents a general geometric structure determined by the centering matrix, and $\mathbf{U}_{cls}$ represents a class-specific structure pointing towards the true class. The coefficients c_{in} and $\gamma_{in} \in (0,1)$ are governed by the hyperparameter α.

Proof. **1. Decomposition via Law of Total Variance.** Recall that the Intrinsic Noise covariance is defined as $\Sigma_{in} = \mathbb{E}_i\left[\mathrm{Cov}_{(\lambda_s,\boldsymbol{y}_r)}\left[\mathbf{g}_i^{(\mathcal{SL})}\right]\right]$. We first analyze the inner covariance term for a fixed sample i. By applying the Law of Total Variance, we decompose the randomness:

$$\mathrm{Cov}_{(\lambda_s,\boldsymbol{y}_r)}\left[\mathbf{g}_i^{(\mathcal{SL})}\right] = \mathrm{Cov}_{\lambda_s}\left[\mathbb{E}_{\boldsymbol{y}_r}\left[\mathbf{g}_i^{(\mathcal{SL})}\right]\right] + \mathbb{E}_{\lambda_s}\left[\mathrm{Cov}_{\boldsymbol{y}_r}\left[\mathbf{g}_i^{(\mathcal{SL})}\right]\right]. \tag{56}$$

Substituting the gradient form of Eq. (53): $\mathbf{g}_i^{(\mathcal{SL})} = \lambda_s \mathbf{g}_i + (1-\lambda_s)\mathbf{g}_{r,i}$, and noting independence between λ_s and $\boldsymbol{y}_r$:

- **First Term (Variance from λ_s):**

$$\begin{aligned}\mathrm{Cov}_{\lambda_s}\left[\mathbb{E}_{\boldsymbol{y}_r}[\mathbf{g}_i^{(\mathcal{SL})}]\right] &= \mathrm{Var}[\lambda_s]\,(\mathbf{g}_i - \bar{\mathbf{g}}_{r,i})\,(\mathbf{g}_i - \bar{\mathbf{g}}_{r,i})^\top \\ &= \sigma_{\lambda_s}^2\,(\mathbf{g}_i - \bar{\mathbf{g}}_{r,i})\,(\mathbf{g}_i - \bar{\mathbf{g}}_{r,i})^\top,\end{aligned} \tag{57}$$

  where $\bar{\mathbf{g}}_{r,i} = \mathbb{E}_{\boldsymbol{y}_r}[\mathbf{g}_{r,i}]$.

- **Second Term (Variance from $\boldsymbol{y}_r$):**

$$\mathbb{E}_{\lambda_s}\left[\mathrm{Cov}_{\boldsymbol{y}_r}[(1-\lambda_s)\mathbf{g}_{r,i}]\right] = \mathbb{E}_{\lambda_s}[(1-\lambda_s)^2]\,\mathrm{Cov}_{\boldsymbol{y}_r}[\mathbf{g}_{r,i}] = c_{in}\,\mathrm{Cov}_{\boldsymbol{y}_r}[\mathbf{g}_{r,i}], \tag{58}$$

  where $c_{in} = \sigma_{\lambda_s}^2 + (1-\mu_{\lambda_s})^2$.

2. Geometric Structure of Gradients. We consider Softmax CE, where $\mathbf{g} = \mathbf{J}^\top(\mathbf{o} - \boldsymbol{y})$. A uniformly sampled one-hot perturbation target $\boldsymbol{y}_r$ satisfies the following moments, regardless of the source-label frequencies:

$$\mathbb{E}[\boldsymbol{y}_r] = \frac{1}{K}\mathbf{1}_K, \quad \mathrm{Cov}[\boldsymbol{y}_r] = \frac{1}{K}\underbrace{\left(I_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)}_{\mathbf{P}\text{ (Centering Matrix)}}.$$

Using these, we simplify the gradient terms:

$$\mathrm{Cov}_{\boldsymbol{y}_r}[\mathbf{g}_{r,i}] = \mathbf{J}_i^\top\,\mathrm{Cov}[\boldsymbol{y}_r]\mathbf{J}_i = \frac{1}{K}\mathbf{J}_i^\top\mathbf{P}\mathbf{J}_i, \tag{59}$$

$$\mathbf{g}_i - \bar{\mathbf{g}}_{r,i} = \mathbf{J}_i^\top(\mathbf{o}_i - \boldsymbol{y}_i) - \mathbf{J}_i^\top(\mathbf{o}_i - \mathbb{E}[\boldsymbol{y}_r]) = \mathbf{J}_i^\top\left(\frac{1}{K}\mathbf{1}_K - \boldsymbol{y}_i\right). \tag{60}$$

3. Synthesis and Structural Alignment. Substituting Eq. (59) and Eq. (60) back and averaging over the dataset:

$$\Sigma_{in} = \mathbb{E}_i\left[\frac{c_{in}}{K}\mathbf{J}_i^\top\mathbf{P}\mathbf{J}_i + \sigma_{\lambda_s}^2\mathbf{J}_i^\top\left(\frac{1}{K}\mathbf{1}_K - \boldsymbol{y}_i\right)\left(\frac{1}{K}\mathbf{1}_K - \boldsymbol{y}_i\right)^\top\mathbf{J}_i\right].$$

To reveal the structure, we factor out $\frac{c_{in}}{K}$ and rewrite the vector $\frac{1}{K}\mathbf{1}_K - \boldsymbol{y}_i = \frac{1}{\sqrt{K}}\left(\frac{1}{\sqrt{K}}\mathbf{1}_K - \sqrt{K}\boldsymbol{y}_i\right)$:

$$\Sigma_{in} = \frac{c_{in}}{K}\mathbb{E}_i\left[\mathbf{J}_i^\top\left(\mathbf{P} + \frac{\sigma_{\lambda_s}^2}{c_{in}}\underbrace{\left(\frac{1}{\sqrt{K}}\mathbf{1}_K - \sqrt{K}\boldsymbol{y}_i\right)\left(\frac{1}{\sqrt{K}}\mathbf{1}_K - \sqrt{K}\boldsymbol{y}_i\right)^\top}_{\mathbf{Q}_i}\right)\mathbf{J}_i\right].$$

Defining $\gamma_{in} = \frac{\sigma_{\lambda_s}^2}{c_{in}}$ and the structural matrices $\mathbf{U}_{\text{base}} \triangleq \mathbb{E}_i\left[\mathbf{J}_i^\top\mathbf{P}\mathbf{J}_i\right]$ and $\mathbf{U}_{\text{cls}} \triangleq \mathbb{E}_i\left[\mathbf{J}_i^\top\mathbf{Q}_i\mathbf{J}_i\right]$, we obtain:

$$\Sigma_{in} = \frac{c_{in}}{K}\left(\mathbf{U}_{\text{base}} + \gamma_{in}\mathbf{U}_{\text{cls}}\right),$$

where $\mathbf{U}_{\text{base}}$ depends on the centering matrix $\mathbf{P}$ (General Geometric Structure), and $\mathbf{U}_{\text{cls}}$ depends on the class-specific residual $\mathbf{Q}_i$ (Class-Specific Structure). Since $c_{in} > \sigma_{\lambda_s}^2$, it follows that $\gamma_{in} \in (0,1)$. □

From conditional per-example noise to batch updates. Set $w = 1 - \lambda_s$, $\varepsilon = \mathbb{E}[w]$, $v_s = \mathrm{Var}(w) = \sigma^2_{\lambda_s}$, $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}_r]$, $\mathbf{V}_r = \mathrm{Cov}(\boldsymbol{y}_r)$, and $\mathbf{d}_i = \mathbf{J}_i^\top(\boldsymbol{y}_i - \boldsymbol{\mu})$. The centered CE gradient at fixed example and parameters is

$$\boldsymbol{\zeta}_i = (w - \varepsilon)\mathbf{d}_i - w\mathbf{J}_i^\top(\boldsymbol{y}_{r,i} - \boldsymbol{\mu}). \tag{61}$$

Independence of w and $\boldsymbol{y}_{r,i}$ makes the cross expectation zero, reproducing $\mathrm{Cov}(\boldsymbol{\zeta}_i) = v_s\mathbf{d}_i\mathbf{d}_i^\top + c_{in}\mathbf{J}_i^\top\mathbf{V}_r\mathbf{J}_i$. For a fixed batch $\mathcal{B}$, independent coefficient–target pairs across examples give

$$\mathrm{Cov}\left(\frac{1}{b}\sum_i \boldsymbol{\zeta}_i \,\middle|\, \mathcal{B}, \boldsymbol{\theta}\right) = \frac{1}{b^2}\sum_i \left(v_s\mathbf{d}_i\mathbf{d}_i^\top + c_{in}\mathbf{J}_i^\top\mathbf{V}_r\mathbf{J}_i\right). \tag{62}$$

If one coefficient w is shared across the batch, and the targets remain mutually independent and independent of w, then

$$\mathrm{Cov}\left(\frac{1}{b}\sum_i \boldsymbol{\zeta}_i \,\middle|\, \mathcal{B}, \boldsymbol{\theta}\right) = v_s\overline{\mathbf{d}}_{\mathcal{B}}\overline{\mathbf{d}}_{\mathcal{B}}^\top + \frac{c_{in}}{b^2}\sum_i \mathbf{J}_i^\top\mathbf{V}_r\mathbf{J}_i, \quad \overline{\mathbf{d}}_{\mathcal{B}} = \frac{1}{b}\sum_i \mathbf{d}_i. \tag{63}$$

The first term now includes cross-example contributions. Permutation-based targets have additional dependence and require their joint target covariance. These formulas condition on the inputs, parameters, and any other model randomness. They describe gradient covariance before momentum or optimizer preconditioning.

Total batch-gradient covariance also contains variability of the conditional mean:

$$\mathrm{Cov}(\mathbf{g}_{\mathcal{B}}) = \mathbb{E}_{\mathcal{B}}[\mathrm{Cov}(\mathbf{g}_{\mathcal{B}} \mid \mathcal{B})] + \mathrm{Cov}_{\mathcal{B}}[\mathbb{E}(\mathbf{g}_{\mathcal{B}} \mid \mathcal{B})]. \tag{64}$$

Cen-Sombrero preserves the first component under the same conditional sampling protocol, while replacing the LS mean by the ERM mean changes the second component in general.

C.3 Proof of Proposition 4: Structure of the Hessian Matrix

To facilitate the proof process, we first establish a lemma characterizing the covariance structure of symmetric noise on the probability simplex. Throughout the derivation, we reuse the notations $\mathbf{P}$, $\mathbf{Q}_i$, $\mathbf{U}_{\text{base}}$, and $\mathbf{U}_{\text{cls}}$ defined in Appendix C.2.

Lemma 12 (Covariance Structure of Symmetric Probability Noise)**.** *Let $\boldsymbol{u} \in \mathbb{R}^K$ be a random vector defined on the probability simplex (i.e., $\boldsymbol{u} \succeq 0$ and $\mathbf{1}_K^\top\boldsymbol{u} = 1$) following a permutation-invariant distribution. The covariance matrix of $\boldsymbol{u}$ takes the form:*

$$\Sigma_u = \sigma_u^2\mathbf{P} \tag{65}$$

where the scalar $\sigma_u^2 \triangleq \frac{K}{K-1}\mathrm{Var}(\boldsymbol{u}^{(j)})$ is bounded within $[0, \frac{1}{K}]$ and quantifies the noise intensity. The marginal variance of any component in $\boldsymbol{u}$ is identical, meaning $\mathrm{Var}(\boldsymbol{u}^{(j)}) = \mathrm{Var}(\boldsymbol{u}^{(k)})$.

Proof. By virtue of permutation invariance, the variance of any component $\boldsymbol{u}^{(j)}$ is identical, denoted by v. Similarly, the covariance between any distinct pair $\boldsymbol{u}^{(j)}$ and $\boldsymbol{u}^{(k)}$ ($j \neq k$) is identical, denoted by c. Consequently, the covariance matrix can be written as $\Sigma_u = (v - c)I_K + c\mathbf{1}_K\mathbf{1}_K^\top$.

The simplex constraint $\mathbf{1}_K^\top\boldsymbol{u} = 1$ implies zero variance for the sum of components:

$$\mathrm{Var}(\mathbf{1}_K^\top\boldsymbol{u}) = \mathbf{1}_K^\top\Sigma_u\mathbf{1}_K = K(v - c) + K^2c = 0 \implies c = -\frac{v}{K-1}. \tag{66}$$

This negative correlation is necessary to satisfy the normalization constraint. Substituting c back into Σ_u yields:

$$\Sigma_u = \frac{Kv}{K-1}I_K - \frac{v}{K-1}\mathbf{1}_K\mathbf{1}_K^\top = \frac{vK}{K-1}\underbrace{\left(I_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)}_{\mathbf{P}}. \tag{67}$$

Defining $\sigma_u^2 \triangleq \frac{vK}{K-1}$ completes the proof. Based on the extrema of the simplex (centroid vs. vertices), σ_u^2 is bounded in $[0, \frac{1}{K}]$. □

Proposition 4 (Restated). *Under the prediction model and moment-factorization assumptions stated below, the model-averaged Generalized Gauss–Newton (GGN) matrix decomposes as:*

$$\tilde{\mathbf{H}}_{\text{mod}} = \frac{c_{\omega,u}}{K}\left[\mathbf{U}_{\text{base}} + \gamma_H \mathbf{U}_{\text{cls}}\right],$$

For $c_{\omega,u} > 0$, the coefficient satisfies $\gamma_H \in [0,1]$. The coefficients describe the uncertainty of the auxiliary prediction model.

Proof. The Hessian matrix $\mathbf{H}$ admits the GGN decomposition: $\mathbf{H} = \mathbf{J}^\top \mathbf{H}_{\text{loss}} \mathbf{J} + \sum_j \boldsymbol{r}^{(j)} \nabla^2_{\boldsymbol{\theta}} \boldsymbol{h}^{(j)}$, where $\mathbf{H}_{\text{loss}} \triangleq \nabla^2_{\boldsymbol{h}} \mathcal{L}$ denotes the Hessian of the loss with respect to the logits $\boldsymbol{h}$, and $\boldsymbol{r} \triangleq \nabla_{\boldsymbol{h}} \mathcal{L}$ is typically affine with respect to the residual. The second term is the model-second-derivative residual. In regimes where it is negligible, the GGN provides a Hessian approximation (Nocedal & Wright, 2006; Martens, 2010; Bottou et al., 2018). We write the actual GGN as $\tilde{\mathbf{H}} = \mathbb{E}_i[\mathbf{J}_i^\top \mathbf{H}_{\text{loss},i} \mathbf{J}_i]$ and distinguish it from its model-averaged representation $\tilde{\mathbf{H}}_{\text{mod}}$ below. For CE with target $\boldsymbol{q}_i$, the residual uses $\boldsymbol{r}_i = \boldsymbol{o}_i - \boldsymbol{q}_i$; convergence alone does not make it zero.

For Softmax CE, we have $\mathbf{H}_{\text{loss}} = \operatorname{diag}(\boldsymbol{o}) - \boldsymbol{o}\boldsymbol{o}^\top$. To model the output distribution $\boldsymbol{o}_i$ under uncertainty, we employ the robust ϵ-contamination model (Huber, 1964):

$$\boldsymbol{o}_i = (1-\omega_i)\boldsymbol{y}_i + \omega_i \boldsymbol{u}_i \,. \tag{68}$$

Here, $\boldsymbol{y}_i$ is the one-hot label representing the true signal, $\boldsymbol{u}_i$ is the symmetric probability noise, and $\omega_i \in [0,1]$ is a random variable representing the uncertainty level. (We adopt the notation ω to substitute for the random variable ϵ in the standard ϵ-contamination model, to prevent notational collisions with the smoothing coefficient ε utilized in LS.) We assume that ω_i and $\boldsymbol{u}_i$ are independent and that their required moments factor from $(\boldsymbol{y}_i, \mathbf{J}_i)$. These assumptions define an auxiliary prediction model: the expectations below average its uncertainty variables while holding the sample label and Jacobian fixed. Their applicability to a learned network is an additional modeling question, distinct from the exact intrinsic-noise calculation.

Substituting the model for $\boldsymbol{o}_i$ into $\mathbf{H}_{\text{loss},i}$:

$$\begin{aligned}\mathbf{H}_{\text{loss},i} &= (1-\omega_i)\operatorname{diag}(\boldsymbol{y}_i) + \omega_i \operatorname{diag}(\boldsymbol{u}_i) \\ &\quad - \left[(1-\omega_i)^2 \boldsymbol{y}_i\boldsymbol{y}_i^\top + \omega_i^2 \boldsymbol{u}_i\boldsymbol{u}_i^\top + \omega_i(1-\omega_i)(\boldsymbol{y}_i\boldsymbol{u}_i^\top + \boldsymbol{u}_i\boldsymbol{y}_i^\top)\right] \qquad (69)\\ &= (\omega_i - \omega_i^2)\left[\boldsymbol{y}_i\boldsymbol{y}_i^\top - (\boldsymbol{y}_i\boldsymbol{u}_i^\top + \boldsymbol{u}_i\boldsymbol{y}_i^\top)\right] + \omega_i \operatorname{diag}(\boldsymbol{u}_i) - \omega_i^2 \boldsymbol{u}_i\boldsymbol{u}_i^\top, \qquad (70)\end{aligned}$$

and taking the expectation over $\boldsymbol{u}$:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{u}}\left[\mathbf{H}_{\text{loss},i}\right] &= (\omega_i - \omega_i^2)\left[\boldsymbol{y}_i\boldsymbol{y}_i^\top - (\boldsymbol{y}_i\boldsymbol{\mu}_u^\top + \boldsymbol{\mu}_u\boldsymbol{y}_i^\top)\right] + \omega_i \operatorname{diag}(\boldsymbol{\mu}_u) - \omega_i^2(\Sigma_u + \boldsymbol{\mu}_u\boldsymbol{\mu}_u^\top) \\ &= (\omega_i - \omega_i^2)\left[(\boldsymbol{\mu}_u - \boldsymbol{y}_i)(\boldsymbol{\mu}_u - \boldsymbol{y}_i)^\top\right] + \omega_i(\operatorname{diag}(\boldsymbol{\mu}_u) - \boldsymbol{\mu}_u\boldsymbol{\mu}_u^\top) - \omega_i^2\Sigma_u \,. \qquad (71)\end{aligned}$$

then using $\boldsymbol{\mu}_u = \frac{1}{K}\mathbf{1}$ and Σ_u from Lemma 12:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{u}}\left[\mathbf{H}_{\text{loss},i}\right] &= (\omega_i - \omega_i^2)\left[\left(\frac{1}{K}\mathbf{1}_K - \boldsymbol{y}_i\right)\left(\frac{1}{K}\mathbf{1}_K - \boldsymbol{y}_i\right)^\top\right] \qquad (72)\\ &\quad + \omega_i\left(\frac{1}{K}I_K - \frac{1}{K^2}\mathbf{1}_K\mathbf{1}_K^\top\right) - \omega_i^2\sigma_u^2\left(I_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right) \\ &= \frac{1}{K}\left[(\omega_i - \omega_i^2)\mathbf{Q}_i + (\omega_i - \omega_i^2\sigma_u^2 K)\mathbf{P}\right]. \qquad (73)\end{aligned}$$

Next, taking the expectation w.r.t. the uncertainty magnitude ω:

$$\mathbb{E}_{\boldsymbol{u},\omega}[\mathbf{H}_{\text{loss},i}] = \frac{1}{K}\left(c_\omega \mathbf{Q}_i + c_{\omega,u}\mathbf{P}\right), \tag{74}$$

where we define the coefficients based on the moments of ω: $c_\omega \triangleq \mathbb{E}[\omega - \omega^2]$, $c_{\omega,u} \triangleq \mathbb{E}[\omega - \omega^2\sigma_u^2 K]$.

Finally, transforming this conditional expected structure by the sample Jacobian and averaging over samples gives the model-averaged GGN:

$$\tilde{\mathbf{H}}_{\text{mod}} = \mathbb{E}_i\left[\mathbf{J}_i^\top \frac{c_{\omega,u}}{K}\left(\mathbf{P} + \frac{c_\omega}{c_{\omega,u}}\mathbf{Q}_i\right)\mathbf{J}_i\right] = \frac{c_{\omega,u}}{K}\left(\underbrace{\mathbb{E}_i[\mathbf{J}_i^\top \mathbf{P}\mathbf{J}_i]}_{\mathbf{U}_{\text{base}}} + \gamma_H \underbrace{\mathbb{E}_i[\mathbf{J}_i^\top \mathbf{Q}_i\mathbf{J}_i]}_{\mathbf{U}_{\text{cls}}}\right), \tag{75}$$

where $\gamma_H = c_\omega / c_{\omega,u}$ when $c_{\omega,u} > 0$. Since $0 \le \sigma_u^2 K \le 1$ and $\omega \in [0,1]$, we have $0 \le c_\omega \le c_{\omega,u}$, so $\gamma_H \in [0,1]$. When $c_{\omega,u} = 0$, the modeled curvature is zero and this ratio is undefined. This establishes the modeled curvature's decomposition in the same two structural matrices as the intrinsic noise. □

Coefficient matching and directional comparison. Both $\mathbf{U}_{\text{base}}$ and $\mathbf{U}_{\text{cls}}$ are positive semidefinite. If $\gamma_{in} = \gamma_H$, then $\Sigma_{in} = (c_{in}/c_{\omega,u})\tilde{\mathbf{H}}_{\text{mod}}$. For $\gamma_H > 0$, write $r_\gamma = \gamma_{in}/\gamma_H$. Comparing the coefficients of the two PSD terms yields

$$\min(1, r_\gamma)\frac{\tilde{\mathbf{H}}_{\text{mod}}}{c_{\omega,u}} \preceq \frac{\Sigma_{in}}{c_{in}} \preceq \max(1, r_\gamma)\frac{\tilde{\mathbf{H}}_{\text{mod}}}{c_{\omega,u}}. \tag{76}$$

For example, when $r_\gamma \le 1$, the difference between the middle matrix and the lower bound is $(1 - r_\gamma)\mathbf{U}_{\text{base}}/K$, and the difference between the upper bound and the middle matrix is $(\gamma_H - \gamma_{in})\mathbf{U}_{\text{cls}}/K$. Both are PSD. This bounds the corresponding quadratic forms in every parameter direction.

Common structural terms alone need not produce common leading eigendirections. For $\mathbf{A} = \mathbf{U}_{\text{base}}$ and $\mathbf{B} = \mathbf{U}_{\text{cls}}$, the commutator of $\mathbf{A} + \gamma_{in}\mathbf{B}$ and $\mathbf{A} + \gamma_H\mathbf{B}$ is $(\gamma_H - \gamma_{in})[\mathbf{A}, \mathbf{B}]$. Exact coefficient matching establishes proportionality, whereas simultaneous diagonalization under unequal coefficients additionally follows if $\mathbf{A}$ and $\mathbf{B}$ commute. Even simultaneous diagonalization does not by itself fix the ordering of leading eigenvalues.

C.4 Empirical Estimation of the Hessian Alignment Coefficient γ_H

In this section, we provide a framework to estimate the latent distributional parameters $(\mu_\omega, \chi_\omega, \sigma_u^2)$ of the robust ϵ-contamination model introduced in Proposition 4. By statistically inferring these values from converged models, we compute the data-dependent coefficients $c_\omega, c_{\omega,u}$, and subsequently their ratio γ_H. The estimates characterize these coefficients across the reported minima and connect the prediction model to the observations in Sec. 4.3.

C.4.1 Estimation via the Method of Moments

Consider a model trained on a K-class classification task that has converged to a local minimum m_t. Following Proposition 4, the prediction for the i-th sample is modeled as $\boldsymbol{o}_i = (1 - \omega_i)\boldsymbol{y}_i + \omega_i \boldsymbol{u}_i$. Here, $\omega_i \in [0,1]$ represents uncertainty, and $\boldsymbol{u}_i$ is the symmetric probability noise. We assume ω, $\boldsymbol{u}$, and the target $\boldsymbol{y}$ are mutually independent.

We can establish a system of equations linking the moments of the latent variables to the observable empirical moments of the model's predictions. We define the empirical first and second-order observable statistics as:

- $M_1 \triangleq \mathbb{E}_i\left[\boldsymbol{o}_i^\top \boldsymbol{y}_i\right]$
- $M_2 \triangleq \mathbb{E}_i\left[\|\boldsymbol{o}_i\|^2\right]$
- $M_3 \triangleq \mathbb{E}_i\left[\|\boldsymbol{o}_i \odot (\mathbf{1}_K - \boldsymbol{y}_i)\|^2\right]$

By expanding the first-order moment M_1 using the contamination model $\boldsymbol{o}_i$ and the covariance structure Σ_u derived in Lemma 12, we can express the expectation μ_ω directly:

$$\mu_\omega = \frac{K}{K-1}\left(1 - M_1\right). \tag{77}$$

For the second-order moments M_2 and M_3, letting $\chi_\omega = \mathbb{E}[\omega^2]$, their expansions yield a linear system governing χ_ω and σ_u^2:

$$\begin{bmatrix} 1 & K \\ \frac{1}{K} & K-1 \end{bmatrix}\begin{bmatrix} \chi_\omega \\ \chi_\omega \sigma_u^2 \end{bmatrix} = \frac{K}{K-1}\begin{bmatrix} M_2 - 2M_1 + 1 \\ M_3 \end{bmatrix}. \tag{78}$$

The two rows follow explicitly from

$$M_2 - 2M_1 + 1 = \frac{K-1}{K}\chi_\omega(1 + K\sigma_u^2),$$
$$M_3 = \frac{K-1}{K}\chi_\omega\left[\frac{1}{K} + (K-1)\sigma_u^2\right]. \tag{79}$$

The determinant of the coefficient matrix is $K - 2$. For $K > 2$, empirical estimates of M_1, M_2, M_3 identify χ_ω and the product $\chi_\omega\sigma_u^2$ within the model; recovering σ_u^2 additionally requires $\chi_\omega > 0$. The inferred moments should satisfy $0 \le \mu_\omega \le 1$, $\mu_\omega^2 \le \chi_\omega \le \mu_\omega$, and $0 \le \sigma_u^2 \le 1/K$. Consequently, we obtain the estimates for $c_\omega = \mu_\omega - \chi_\omega$ and $c_{\omega,u} = \mu_\omega - \chi_\omega\sigma_u^2 K$, as well as $\gamma_H = c_\omega/c_{\omega,u}$.

Remark 13 (Singularity at $K = 2$). *For binary classification ($K = 2$), the linear system in Eq. (78) becomes singular. This arises from the inherent symmetry of the 1-simplex geometry, where the variance contribution from the uncertainty magnitude ω and the noise direction $\mathbf{u}$ become indistinguishable up to the second order. Second-order prediction statistics therefore do not identify both latent contributions when $K = 2$; additional distributional restrictions or independent information would be needed. The reported empirical analysis uses $K > 2$ (CIFAR-100, CIFAR-10, and MNIST).*

C.4.2 Empirical Results across Minima

To observe the behavior of these parameters, we conducted extensive experiments across diverse datasets (CIFAR-100, CIFAR-10, MNIST) and network architectures (ResNet-18, ShuffleNet, VGG-16). All models were trained to full convergence. Following the preceding section, we estimate the latent variables $\hat{\mu}_\omega$, $\hat{\chi}_\omega$, and $\hat{\sigma}_u^2$ at these minima, which subsequently allows us to derive the structural coefficients $\hat{c}_\omega$ and $\hat{c}_{\omega,u}$. Table 8 presents the estimation results.

The reported estimates show two patterns across these datasets and architectures:

1. **Coefficient variability across runs:** The table reports the mean and standard deviation of c_ω, $c_{\omega,u}$, and γ_H over five seeds for each setting. These estimates describe the stability of the model coefficients within the reported settings. They do not independently establish moment factorization with the Jacobian or equality of the measured matrices' leading eigenspaces.
2. **Impact of Smoothing:** Compared with ERM and Cen-Sombrero, the LS and standard Sombrero estimates have smaller modeled directional variance ($\hat{\sigma}_u^2 K \to 0$) and relate the uncertainty magnitude to the equivalent smoothing parameter ($\hat{\mu}_\omega \approx \varepsilon$, $\hat{\sigma}_\omega \ll \hat{\mu}_\omega \implies \hat{\chi}_\omega \approx \hat{\mu}_\omega^2$). When the directional variance is small and the uncertainty magnitude concentrates near ε, the model gives the approximation:

$$c_\omega \approx \mu_\omega - \mu_\omega^2\,, \quad c_{\omega,u} \approx \mu_\omega \implies \gamma_H = \frac{c_\omega}{c_{\omega,u}} \approx 1 - \mu_\omega \approx 1 - \varepsilon\,. \tag{80}$$

Figure 8 reports the corresponding sweep over ε. Comparing the empirical estimates of μ_ω, c_ω, $c_{\omega,u}$, and γ_H with the model approximations shows how changing the mean smoothing level is associated with the inferred curvature coefficients. The coefficient estimates and their deviations provide the empirical context for using $\gamma_H \approx 1 - \varepsilon$ in the following calibration.

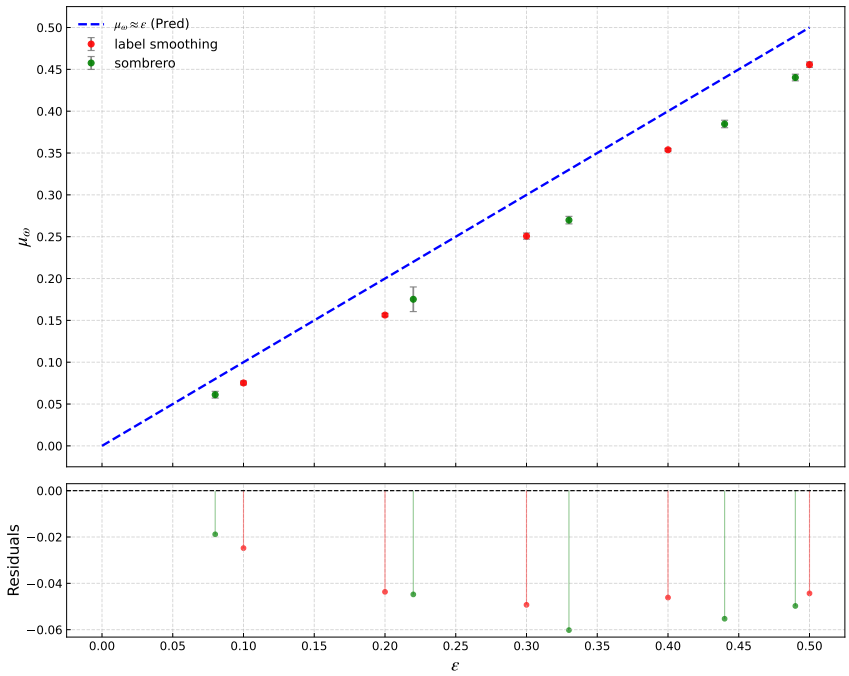
Table 8: **Empirical estimation across diverse tasks.** Mean ± standard deviation over five independent runs. ε is the LS coefficient or Sombrero's equivalent mean-smoothing factor. Sombrero and Cen-Sombrero use $\alpha = 1.0$. The two panels report the same runs; $\sigma_\omega = \sqrt{\chi_\omega - \mu_\omega^2}$.

(a) Mean smoothing and uncertainty magnitude.

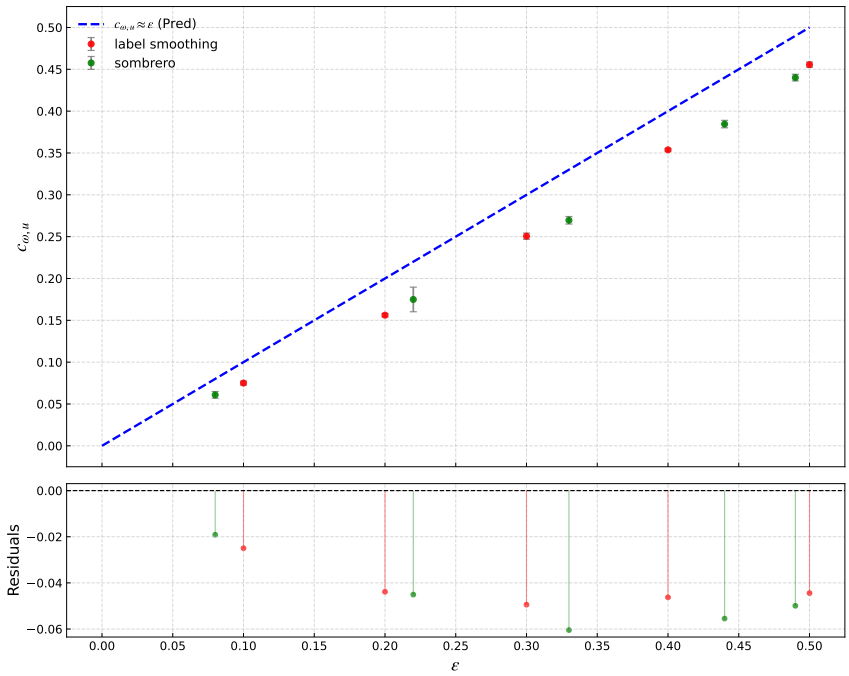
Setting	**Method**	ε	$\hat{\mu}_\omega$	$\hat{\chi}_\omega$ $(\times 10^{-1})$	$\hat{\sigma}_\omega$
CIFAR-100 ResNet-18	Baseline (ERM)	0.000	0.003±.000	0.003±.000	0.017±.000
	Cen-Somb.	0.000	0.005±.000	0.008±.001	0.028±.002
	Label Smoothing	0.200	0.156±.002	0.271±.007	0.052±.001
	Sombrero	0.333	0.270±.004	0.781±.025	0.073±.002
CIFAR-10 ShuffleNet	Baseline (ERM)	0.000	0.009±.001	0.022±.004	0.046±.004
	Cen-Somb.	0.000	0.038±.003	0.165±.016	0.123±.006
	Label Smoothing	0.200	0.209±.002	0.462±.009	0.050±.001
	Sombrero	0.333	0.343±.010	1.227±.063	0.072±.004
MNIST VGG-16	Baseline (ERM)	0.000	0.001±.000	0.002±.002	0.014±.005
	Cen-Somb.	0.000	0.019±.009	0.125±.061	0.107±.024
	Label Smoothing	0.200	0.203±.001	0.415±.005	0.019±.002
	Sombrero	0.333	0.349±.016	1.224±.115	0.021±.001

(b) Directional variance and curvature coefficients.

Setting	**Method**	$\hat{\sigma}_u^2 K$ $(\times 10^{-1})$	$\hat{c}_\omega$ $(\times 10^{-1})$	$\hat{c}_{\omega,u}$ $(\times 10^{-1})$	$\hat{\gamma}_H$
CIFAR-100 ResNet-18	Baseline (ERM)	8.765±.065	0.023±.001	0.023±.001	0.985±.001
	Cen-Somb.	8.469±.196	0.038±.004	0.039±.004	0.968±.002
	Label Smoothing	0.068±.002	1.292±.013	1.561±.020	0.828±.002
	Sombrero	0.033±.002	1.917±.020	2.696±.045	0.711±.004
CIFAR-10 ShuffleNet	Baseline (ERM)	7.770±.083	0.067±.007	0.072±.007	0.932±.005
	Cen-Somb.	7.074±.194	0.211±.012	0.260±.018	0.814±.012
	Label Smoothing	0.282±.024	1.627±.011	2.076±.020	0.784±.002
	Sombrero	0.255±.032	2.200±.033	3.396±.096	0.648±.009
MNIST VGG-16	Baseline (ERM)	7.923±.576	0.004±.001	0.004±.001	0.902±.036
	Cen-Somb.	7.306±.702	0.065±.030	0.101±.051	0.662±.049
	Label Smoothing	0.059±.016	1.613±.005	2.025±.009	0.796±.001
	Sombrero	0.019±.004	2.264±.050	3.486±.165	0.650±.016

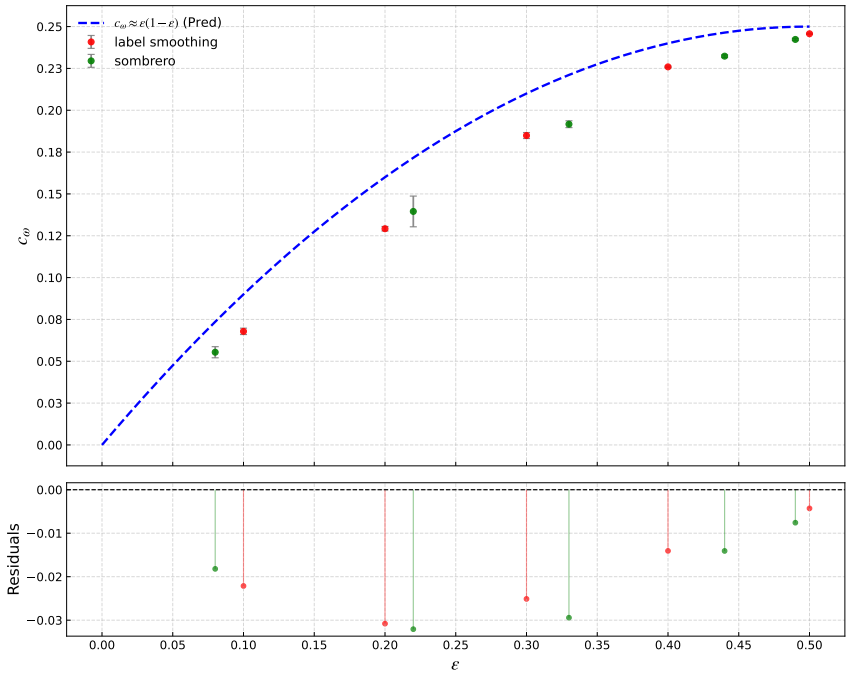


(a) μ_ω

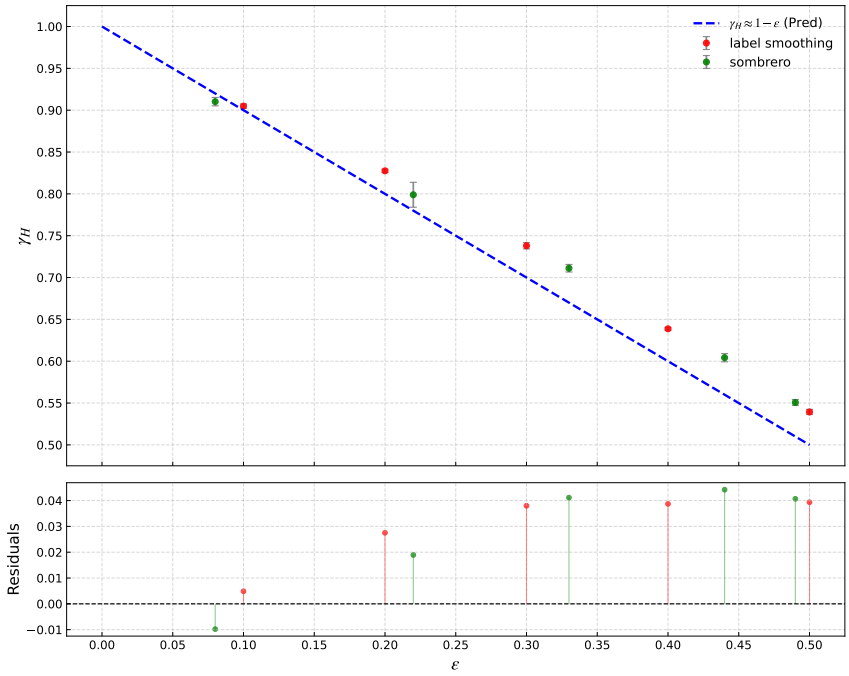


(b) $c_{\omega,u}$

Figure 8: **Empirical validation of the impact of smoothing.** The four subplots illustrate the relationship between the (equivalent) smoothing factor ε (x-axis) and the estimated latent variables (y-axis). Markers represent the empirical estimates derived from fully converged ResNet-18 models on CIFAR-100 under Label Smoothing or Sombrero Loss. The dashed lines denote our theoretical approximations ($\mu_\omega \approx \varepsilon$, $c_\omega \approx \varepsilon(1-\varepsilon)$, $c_{\omega,u} \approx \varepsilon$, and $\gamma_H \approx 1-\varepsilon$). The comparison displays the empirical trends and their deviations from these model approximations. Lower panels display residuals from the corresponding approximations.



(c) c_ω



(d) γ_H

Figure 8: **Empirical validation of the impact of smoothing (continued).** Estimates of c_ω and γ_H, with the corresponding theoretical approximations and residuals.

C.5 ALIGNMENT OF γ_{in} AND γ_H

In this section, we theoretically bridge the Hessian structure (γ_H) with the Sombrero Loss Intrinsic Noise covariance (γ_{in}).

First, we formalize the coefficient γ_{in}, which governs the alignment with the class-specific structure. By substituting the variance $\sigma^2_{\lambda_s}$ and expectation μ_{λ_s} of the interpolation coefficient (derived in Appendix B.2) into the definition of γ_{in}, we obtain:

$$\gamma_{in} = \frac{\sigma^2_{\lambda_s}}{\sigma^2_{\lambda_s} + (1-\mu_{\lambda_s})^2} = \frac{1}{(2\alpha+1)(\alpha+1)}. \tag{81}$$

Concurrently, following Proposition 2, the equivalent smoothing factor is determined by α as $\varepsilon = \frac{\alpha}{2\alpha+1}$. By substituting this into the empirically validated approximation $\gamma_H \approx 1-\varepsilon$ (Eq. (80)), we construct the alignment ratio between γ_{in} and γ_H:

$$\frac{\gamma_{in}}{\gamma_H} \approx \frac{\gamma_{in}}{1-\varepsilon} = \frac{1}{(\alpha+1)^2} \in (0,1). \tag{82}$$

The trajectory of this ratio is visualized in Figure 9. Crucially, in the small-α regime (e.g., $\alpha \to 0$), the ratio converges to 1. Within the approximation $\gamma_H \approx 1-\varepsilon$, this brings the normalized covariance and modeled GGN closer in the directional bounds of Eq. (76). The noise magnitude also vanishes as $\alpha \to 0$, so coefficient matching and noise strength remain distinct quantities. For typical Sombrero Loss hyperparameter settings (e.g., small $\alpha \in [0.1, 0.4]$), this ratio still remains greater than 0.5.

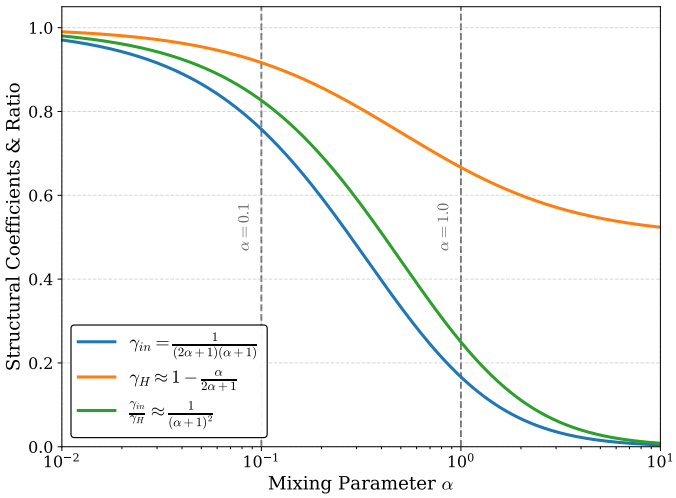


Figure 9: **The alignment ratio γ_{in}/γ_H under varying α.** The ratio approaches 1 as $\alpha \to 0$, indicating coefficient matching within the approximation $\gamma_H \approx 1-\varepsilon$ in the small-α regime.

Discussion: The Coupling Dilemma and Analytical Calibration.

Eq. (82) shows that the coefficient ratio approaches one as $\alpha \to 0$ within the stated curvature approximation. In the same limit, both the mean target-smoothing level and the intrinsic-noise magnitude vanish. This coupling motivates controlling the mean and stochastic components separately.

The Dec-Sombrero framework in Appendix B.3 separates the mean smoothing level from the noise parameter. A one-free-parameter calibration then matches γ_{in} to the modeled curvature coefficient. Since $\gamma_H \approx 1-\varepsilon$, we require $1-\varepsilon = \gamma_{in}$. This gives us the calibrated smoothing factor:

$$\varepsilon^* = 1 - \gamma_{in} = 1 - \frac{1}{(2\alpha+1)(\alpha+1)}. \tag{83}$$

Using ε^* for smoothing and α for intrinsic noise matches the two coefficients under the prediction-model approximation. Because ε^* is determined by α, the calibrated variant retains one free parameter. Its empirical performance is reported in Sec. 5.1; the calibration supplies a geometric design principle rather than a guarantee of the optimum over all training settings.

C.6 Escape Efficiency through Curvature–Noise Interaction

The preceding analysis relates the intrinsic covariance to local curvature. We now examine the resulting *escape-efficiency proxy*: the local loss increase produced by a centered stochastic update. This makes explicit the role of both noise direction and noise magnitude in the optimization interpretation discussed in Sec. 4.3.

Local energy injection. Let $L(\boldsymbol{\theta})$ be a twice continuously differentiable scalar objective, let $\mathbf{H} = \nabla^2 L(\boldsymbol{\theta})$, and let $\boldsymbol{\zeta}$ have conditional mean zero and covariance Σ at the fixed parameter vector. Consider the update $\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta(\overline{\mathbf{g}} + \boldsymbol{\zeta})$, where $\overline{\mathbf{g}}$ is its conditional mean gradient. Relative to the same update with zero noise, the second-order Taylor model has expected loss difference

$$\mathbb{E}\big[L(\boldsymbol{\theta}^+) \mid \boldsymbol{\theta}\big] - L(\boldsymbol{\theta} - \eta\overline{\mathbf{g}}) = \frac{\eta^2}{2}\operatorname{Tr}(\mathbf{H}\Sigma) + o(\eta^2). \tag{84}$$

For this asymptotic equality, the Taylor remainder must also be integrable; bounded noise in a local neighborhood, or a suitable Hessian bound and moment condition, suffices. For a quadratic objective the equality holds exactly without the remainder. At a stationary point the comparison is simply to $L(\boldsymbol{\theta})$.

Proof. Expand L at $\boldsymbol{\theta}$ along both updates. The linear difference is $-\eta\nabla L(\boldsymbol{\theta})^\top \mathbb{E}[\boldsymbol{\zeta}] = 0$. In the quadratic difference, the cross term also vanishes, leaving $\eta^2\mathbb{E}[\boldsymbol{\zeta}^\top\mathbf{H}\boldsymbol{\zeta}]/2$. Using $\mathbb{E}[\boldsymbol{\zeta}\boldsymbol{\zeta}^\top] = \Sigma$ gives Eq. (84). □

Accordingly, define $\mathcal{E}(\mathbf{H}, \Sigma) = \operatorname{Tr}(\mathbf{H}\Sigma)$. For PSD curvature this quantity measures how noise deposits energy along positively curved directions. One may use the Hessian at a local minimum, or the GGN as a PSD curvature approximation under the assumptions above. The trace expression motivates studying escape from narrow basins (Zhu et al., 2019); translating it into a finite first-passage time additionally requires the basin boundary, potential barriers, drift, and diffusion along the trajectory.

Fixed covariance scale across two minima. Let $\mathbf{H}_s, \mathbf{H}_f \succeq 0$ denote curvature at two minima, with positive denominators in the ratios below. Isotropic noise with the same variance σ^2 in every parameter direction at both minima gives

$$R_{iso} = \frac{\mathcal{E}(\mathbf{H}_s, \sigma^2 I)}{\mathcal{E}(\mathbf{H}_f, \sigma^2 I)} = \frac{\operatorname{Tr}(\mathbf{H}_s)}{\operatorname{Tr}(\mathbf{H}_f)}. \tag{85}$$

For aligned noise satisfying $\Sigma_t = \kappa\mathbf{H}_t$, $t \in \{s, f\}$, with the *same* positive proportionality coefficient κ, the analogous ratio is

$$R_{in} = \frac{\operatorname{Tr}(\mathbf{H}_s^2)}{\operatorname{Tr}(\mathbf{H}_f^2)} = \frac{\sum_j \xi_{s,j}^2}{\sum_j \xi_{f,j}^2}, \tag{86}$$

where $\xi_{t,j}$ denotes an eigenvalue of $\mathbf{H}_t$. In particular, if $\mathbf{H}_s = a\mathbf{H}_f$ for $a > 1$, then $R_{iso} = a$ and $R_{in} = a^2$. This is the precise quadratic curvature scaling behind the escape-efficiency interpretation. Without this proportional-curvature condition, ordering the two ratios requires additional information about both spectra. If the matching coefficient is instead $\kappa_t = c_{in}/c_{\omega,u,t}$, the ratio acquires the factor κ_s/κ_f. The coefficient estimates in Table 8 provide empirical context for this scale, rather than an identity across all minima.

The fixed-κ comparison includes an increase in total noise energy, since $\operatorname{Tr}(\Sigma_t) = \kappa \operatorname{Tr}(\mathbf{H}_t)$. The following comparison separates directional allocation from that change in magnitude.

Proposition 14 (Curvature Allocation at Equal Noise Energy)**.** *For a PSD matrix $\mathbf{H} \in \mathbb{R}^{d_\theta \times d_\theta}$ with positive trace and a fixed covariance trace $\tau > 0$, consider $\Sigma_{align} = \tau\mathbf{H}/\operatorname{Tr}(\mathbf{H})$ and $\Sigma_{iso} = \tau I_{d_\theta}/d_\theta$. Then*

$$\frac{\mathcal{E}(\mathbf{H}, \Sigma_{align})}{\mathcal{E}(\mathbf{H}, \Sigma_{iso})} = \frac{d_\theta \operatorname{Tr}(\mathbf{H}^2)}{\operatorname{Tr}(\mathbf{H})^2} \geq 1, \tag{87}$$

with equality if and only if all eigenvalues of $\mathbf{H}$ are equal.

Proof. The numerator interaction is $\tau \operatorname{Tr}(\mathbf{H}^2)/\operatorname{Tr}(\mathbf{H})$, and the denominator is $\tau \operatorname{Tr}(\mathbf{H})/d_\theta$. Cauchy–Schwarz yields $(\sum_j \xi_j)^2 \leq d_\theta \sum_j \xi_j^2$, with equality precisely when the eigenvalues are identical. Substitution proves the result. □

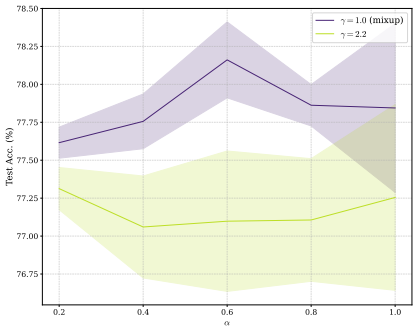


Figure 10: **Sensitivity to the input-mixing rule.** Test accuracy across different distribution parameters α and mixing mechanisms γ. Results are averaged over 5 independent runs, with solid lines denoting the mean and shaded regions representing the standard deviation. The radiometrically motivated power-law mixing rule ($\gamma = 2.2$) underperforms standard linear interpolation ($\gamma = 1.0$) at the evaluated strengths.

Thus, within this equal-energy comparison, curvature-proportional allocation produces at least as much local energy injection as isotropic allocation at the *same* curvature matrix. It is not the unrestricted maximizer over all PSD covariances of trace τ: that maximum is attained by concentrating covariance in a top eigendirection. Across two different minima with the same trace budget, the aligned ratio becomes

$$R_{align}^{(\tau)} = \frac{\mathrm{Tr}(\mathbf{H}_s^2)/\,\mathrm{Tr}(\mathbf{H}_s)}{\mathrm{Tr}(\mathbf{H}_f^2)/\,\mathrm{Tr}(\mathbf{H}_f)}. \tag{88}$$

For $\mathbf{H}_s = a\mathbf{H}_f$, this ratio is a, demonstrating why the noise normalization must be specified when interpreting quadratic scaling.

Connection to Sombrero and the measured landscapes. Coefficient matching supplies the proportional-covariance premise within the auxiliary curvature model. More generally, the directional inequalities in Eq. (76) also bound the interaction: for PSD $\mathbf{H}$, taking $\mathrm{Tr}(\mathbf{H}\,\cdot)$ preserves the inequalities. The shared-coefficient batch formula in Eq. (63) must be used when that is the update protocol; the per-example matching result alone does not establish proportionality of every mini-batch covariance. These results explain how structured target perturbations can allocate stochastic updates according to model geometry. The Hessian spectra and two-dimensional landscapes in Table 2 and Figure 3 then characterize the solutions reached by the reported training runs. Together, they support investigating curvature-sensitive regularization and flat-minimum selection without requiring explicit second-order computation in the training objective.

D Additional Empirical Results and Analyses

D.1 Gixup: Varying the Input-Mixing Operation

Input-mixing methods can incorporate useful domain structure, as in CutMix (Yun et al., 2019) and PuzzleMix (Kim et al., 2020). To examine how a different mixing rule affects training in our setting, we use Generalized Mixup (Gixup):

$$\widetilde{\boldsymbol{x}}^{(gix)} = \left(\lambda \boldsymbol{x}_i^{\gamma} + (1-\lambda)\boldsymbol{x}_j^{\gamma}\right)^{1/\gamma}. \tag{89}$$

The parameter γ specifies a power-law transformation of nonnegative pixel intensities before interpolation. At $\gamma = 1.0$, this is standard linear pixel mixing. At $\gamma = 2.2$, it approximates decoding into a linear light-intensity domain, interpolating there, and re-encoding (Stokes et al., 1996). This radiometric motivation uses a power-law approximation to image encoding; it does not establish that the resulting mixed image belongs to a natural-image manifold. The distinction between encoded and physical intensity is detailed below.

Figure 10 reports means and standard deviations across five runs. In this protocol, adopting the radiometrically motivated transformation does not improve generalization: $\gamma = 2.2$ underperforms

$\gamma = 1.0$ at every evaluated α. The observation motivates studying the optimization effects of the mixing operation in addition to its visual properties. It does not distinguish all possible data-augmentation explanations, since changing γ also changes the mathematical perturbation introduced during training.

D.1.1 Physical Interpretation of Gixup: Radiometric vs. Perceptual Mixing

We explain the light-intensity interpretation of the $\gamma = 2.2$ power-law model and illustrate the comparison with standard Mixup ($\gamma = 1.0$).

The Physics of Gamma Correction. Digital images commonly store nonlinearly encoded intensities. Power-law descriptions of perceptual response have a psychophysical basis (Stevens, 1957); the numerical encoding convention has a separate technical specification. The original sRGB proposal describes a piecewise transfer function close to a gamma-2.2 curve and explains why optical mixing operations should use decoded linear intensities (Stokes et al., 1996). We use the corresponding power-law approximation, under which a stored pixel value x and linear light intensity u satisfy:

$$x = u^{\frac{1}{\gamma}}, \quad \text{where } \gamma \approx 2.2. \tag{90}$$

Standard Mixup performs linear interpolation in the compressed pixel space (x-space). However, the physical superposition of light (e.g., two projected images superposed on a screen) is additive in the linear radiometric domain (u-space). Therefore, a physically faithful mixing operation should first decode the pixel to linear intensity, mix, and then re-encode:

$$\widetilde{\boldsymbol{x}}_{physical} = \left(\lambda \boldsymbol{x}_i^{\gamma} + (1-\lambda)\boldsymbol{x}_j^{\gamma}\right)^{\frac{1}{\gamma}}. \tag{91}$$

Within this power-law model, the decode–mix–encode operation is exactly the Gixup formulation with $\gamma = 2.2$.

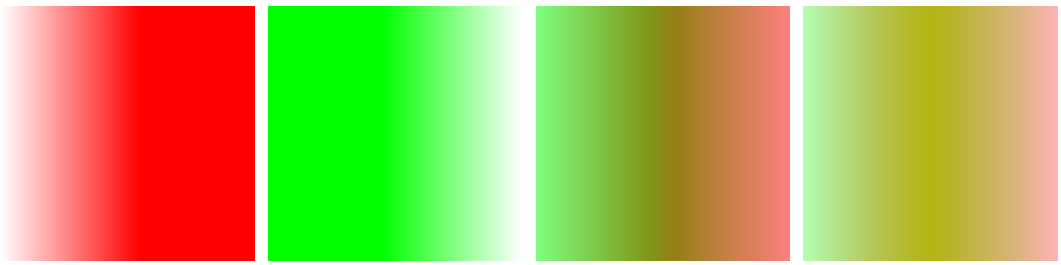
Visual Artifacts and Jensen's Inequality. The two mixing rules produce a systematic brightness difference under the model. Since $u = x^{\gamma}$ is convex for $\gamma > 1$ on nonnegative intensities, Jensen's inequality gives:

$$\underbrace{(\lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j)^{\gamma}}_{\text{Light intensity of Mixup}} \leq \underbrace{\lambda \boldsymbol{x}_i^{\gamma} + (1-\lambda)\boldsymbol{x}_j^{\gamma}}_{\text{Physical light intensity}}. \tag{92}$$

Applying the monotonic increasing encoding function $(\cdot)^{\frac{1}{\gamma}}$ to both sides, we obtain:

$$\underbrace{\lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j}_{\text{Mixup}} \leq \underbrace{\left(\lambda \boldsymbol{x}_i^{\gamma} + (1-\lambda)\boldsymbol{x}_j^{\gamma}\right)^{\frac{1}{\gamma}}}_{\text{Gixup}}. \tag{93}$$

Thus, relative to the decode–mix–encode reference under the assumed power law, linear mixing yields a darker transition between distinct intensities. Figure 11 illustrates this effect for two colors. This visual property and the downstream classification result are different measurements: the brighter transition is not accompanied by better accuracy in Figure 10.



(a) $\boldsymbol{x}_i$ (b) $\boldsymbol{x}_j$ (c) $\gamma = 1.0$ (Mixup) (d) $\gamma = 2.2$ (Gixup)

Figure 11: **Visual comparison of mixing mechanisms.** Color-transition example comparing linear interpolation in encoded pixel space ($\gamma = 1.0$) and decode–mix–encode interpolation under a power-law intensity model ($\gamma = 2.2$). The latter produces a brighter transition.

D.2 Detailed Experimental Setup for Sec. 4

D.2.1 Hessian Computation Details

To compute the Hessian spectrum statistics efficiently for deep neural networks, we utilized the `PyHessian` [1] library. Specifically, we employed the Lanczos algorithm to approximate the top-1 eigenvalues and Stochastic Lanczos Quadrature (SLQ) to estimate the spectral density. All measurements were conducted on a fixed, stratified subset of the training set (10%) at the final epoch of training.

D.2.2 Landscape Visualization Methodology

The loss landscape is visualized using **Filter Normalization** (Li et al., 2018), which scales perturbation directions by the corresponding filter norms to reduce the influence of weight-scale differences on the comparison. We utilized a modified version of the `loss-landscapes` [2] library. For a parameter vector $\boldsymbol{\theta}$ and a random direction vector $\boldsymbol{\delta}$, the normalized direction $\hat{\boldsymbol{\delta}}$ for each filter l is scaled as:

$$\hat{\boldsymbol{\delta}}^{(l)} = \frac{\boldsymbol{\delta}^{(l)}}{\|\boldsymbol{\delta}^{(l)}\|_F} \|\boldsymbol{\theta}^{(l)}\|_F. \tag{94}$$

We visualize the landscape within the range $[-0.1, 0.1]$ along two orthogonal random directions. The Training Loss Surface (Optimization Landscape) and Test Error Surface (Generalization Robustness) were computed using fixed subsets of the training set (10%) and test set (50%), respectively.

D.2.3 Hyperparameter Equivalence

We match the conditional mean gradient of the uniform-target Sombrero construction to LS under the assumptions of Proposition 2. For **Sombrero Loss**, $\alpha = 0.6$. The corresponding **LS** coefficient is $\varepsilon = \alpha/(2\alpha + 1) \approx 0.273$. The LS model in Table 2 uses this matched value. This removes a difference in the expected smoothing strength at fixed parameters; stochastic realizations and the resulting training trajectories remain different.

D.2.4 Robustness of Landscape Analysis

We recompute the Hessian at the learned parameters using hard and smoothed evaluation labels. Figure 12 shows both comparisons, with training and smoothed-label Hessian evaluation using $\varepsilon = 0.273$. The narrower Sombrero spectrum relative to LS is visible with both label choices. This checks sensitivity to the evaluation labels within the stated parameterization and measurement protocol.

D.3 Detailed Experimental Setup for Section 5

To ensure reproducibility, we detail the specific hyperparameter configurations, datasets, and training environments utilized across the diverse experiments presented in Sec. 5. We mention that all experiments were conducted on computing clusters equipped with NVIDIA RTX 4090 and A800 GPUs.

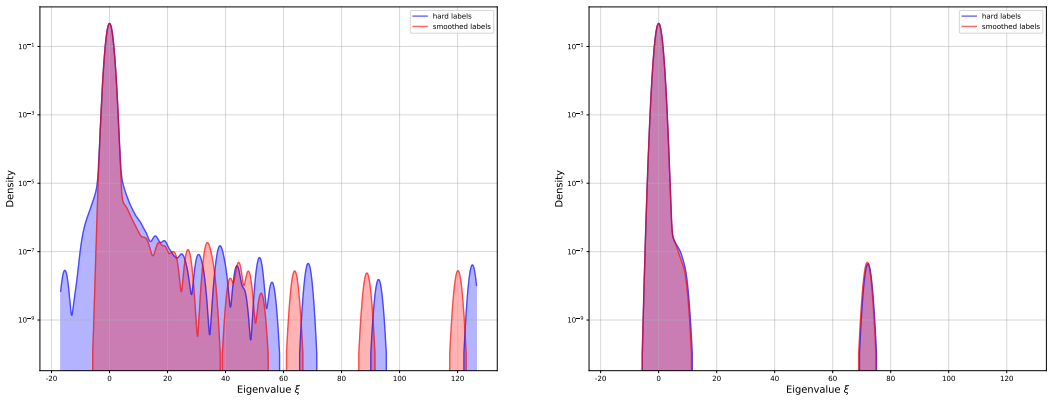
D.3.1 Image Classification Setup

CIFAR-100. We train a Pre-Activation ResNet-18 for 200 epochs using Stochastic Gradient Descent (SGD) with a momentum of 0.9 and weight decay. The initial learning rate is set to 0.1 and decayed by a factor of 0.1 at epochs 100 and 150. Unless otherwise specified, the perturbation target $\boldsymbol{y}_r$ is sampled uniformly from the label space.

ImageNet-1K. ResNet models (ResNet-18 and ResNet-50) are trained for 100 epochs using SGD (momentum 0.9, weight decay 1×10^{-4}, batch size 256) with a base learning rate of 0.1 that decays at epochs 30, 60, and 90. Standard augmentations, including RandomResizedCrop and

[1] `https://github.com/amirgholami/PyHessian`

[2] `https://github.com/marcellodebernardi/loss-landscapes`



(a) Label Smoothing (b) Sombrero Loss

Figure 12: **Robustness of Spectral Analysis.** Comparison of Hessian eigenvalue densities computed using smoothed labels (matching the training objective with $\varepsilon = 0.273$) and hard one-hot labels. In both scenarios, Sombrero Loss consistently exhibits a significantly more compressed spectrum compared to Label Smoothing. The comparison shows that the observed ordering is retained under both evaluation-label choices in this parameterization.

Table 9: **Target-level regularization on CIFAR-100.** Top-1 accuracy (%) with Pre-Activation ResNet-18. ERM: mean±SD over five runs; other entries: single runs. Mixup-TO applies batch-paired label interpolation to unmodified inputs.

Method	Top-1 (%)
Baseline (ERM)	76.23 ± 0.42
Mixup-TO (Wang et al., 2023)	77.10
Sombrero	**78.49**

RandomHorizontalFlip, are applied during training. For the lightweight ShuffleNet V2 1.0x, we use a learning rate of 0.5, weight decay 4×10^{-5}, batch size 1024 and LS ($\varepsilon = 0.1$) for baseline. For the Vision Transformer (DeiT-Small (Touvron et al., 2021)), we adopt the standard 300-epoch training recipe using AdamW optimizer with a base learning rate of 1×10^{-3}, weight decay of 0.05, and a cosine learning rate decay schedule. Standard data augmentations including RandAugment, Mixup, and CutMix are disabled in the ERM baseline to rigorously isolate the effect of our loss formulation.

Regarding the hyperparameters of our methods, we perform a grid search on the `Mini-ImageNet` [3] dataset to determine the optimal α. Specifically, α is swept from 0.1 to 0.5 for Sombrero Loss and from 0.1 to 0.25 for its variant, Dec-Sombrero (Appendix B.3). Based on this search, we select $\alpha = 0.2$ for Sombrero Loss and $\alpha = 0.15$ for Dec-Sombrero.

D.3.2 Ablation Study Setup: Mixup-TO

Mixup-Target Only (Mixup-TO) retains the standard Mixup pairing and label interpolation within a batch while leaving the input images unchanged. This gives a target-level baseline related to the label-mixing analysis of Wang et al. (2023). Its comparison to Sombrero changes both the coefficient law and the target-sampling rule. Table 9 measures the accuracy difference associated with this joint change on CIFAR-100. The LS/Sombrero and ERM/Cen comparisons in Table 1 provide complementary mean-matched comparisons.

D.3.3 Object Detection Setup

We integrate Sombrero Loss into the Deformable-DETR framework (ResNet-50 backbone) trained on the *train2017* split and evaluated on the MS COCO *val2017* split. Models are optimized using the

[3] `https://www.kaggle.com/datasets/arjunashok33/miniimagenet`

AdamW optimizer with a base learning rate of 2×10^{-4} (scaled down to 2×10^{-5} for the backbone) and a weight decay of 1×10^{-4}. We follow the standard 50-epoch schedule, implementing a learning rate drop at epoch 40. In practice, we observed that immediately applying target perturbations during the highly unstable initial phases can disrupt training convergence. To ensure stability, we adopt a fade-in strategy, enabling Sombrero Loss after 5 epochs.

Adapting Sombrero Loss to set prediction requires handling the discrete assignment mechanism. Let $\hat{\boldsymbol{y}}$ be the model predictions and $\boldsymbol{\sigma}_H$ be the optimal matching derived via the Hungarian algorithm. We explored two perturbation strategies:

1. **Shuffled Targets ($\mathcal{SL}_{shuf}$):** Generates a shuffled target $\boldsymbol{y}_r$ and computes a secondary optimal matching $\boldsymbol{\sigma}_H(\hat{\boldsymbol{y}}, \boldsymbol{y}_r)$. While mathematically intuitive, this doubles the matching complexity.
2. **Random Assignment ($\mathcal{SL}_{rand}$):** Bypasses the secondary Hungarian algorithm by drawing a random assignment permutation $\boldsymbol{\sigma}_r \sim \mathcal{U}(S_N)$:

$$\mathcal{SL}_{rand} = \lambda_s \mathcal{L}_{det}\left(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\sigma}_H\left(\hat{\boldsymbol{y}}, \boldsymbol{y}\right)\right) + (1-\lambda_s)\mathcal{L}_{det}\left(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\sigma}_r\right)$$

Both terms use the same detector predictions; the perturbation enters through the assignment within the detection loss. This is a task-specific realization of the loss-level construction. Table 10 reports the complete point-estimate comparison. Random assignment avoids the second Hungarian matching and yields 44.4 AP, compared with 44.3 for shuffled targets and 44.0 for the baseline. We use $\mathcal{SL}_{rand}$ for the main detection result. Uniformity over assignment permutations does not by itself imply isotropic parameter-gradient noise.

Table 10: **Object Detection Results.**

Method	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_S$	**AP**$_M$	**AP**$_L$
Baseline	44.0	**62.6**	47.8	**26.4**	47.6	58.2
Sombrero ($\mathcal{SL}_{shuf}$, $\alpha = 0.4$)	44.3	62.1	48.0	26.3	48.0	58.5
Sombrero ($\mathcal{SL}_{rand}$, $\alpha = 0.4$)	**44.4**	62.5	**48.3**	26.3	**48.2**	**59.1**

D.3.4 Machine Translation Setup

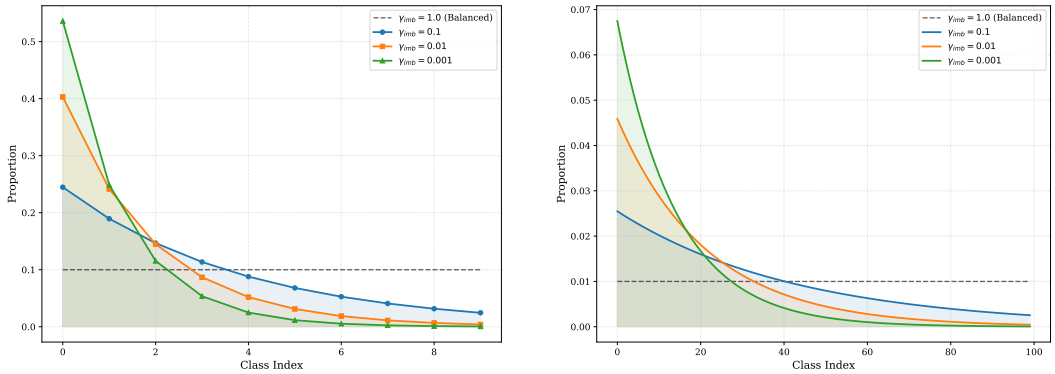
Adapted from the official `fairseq` implementation, we employ the Transformer-Base architecture ($d_{model} = 512$, 8 attention heads) on the WMT 2014 English-German (En-De) and English-French (En-Fr) datasets. Models are optimized via Adam with an inverse square root learning rate schedule, 4000 warmup steps and a dropout rate of 0.1. The baseline utilizes standard LS ($\varepsilon = 0.1$). Alongside $\alpha = 0.15$, we test $\alpha = 0.125$, corresponding to nominal mean smoothing $\varepsilon = 0.1$ under the uniform-target relation in Proposition 2. This comparison tests the practical value of stochastic supervision in translation, complementing the formal geometry analysis for classification.

D.4 Extended Experiments on Class Imbalance

Imbalanced Dataset Construction. We construct fixed-total resampled long-tailed variants of CIFAR-10 and CIFAR-100 to examine how perturbation-target priors affect performance under varying training class frequencies. We adopt an exponential decay model to determine the frequency of each class. Let K denote the total number of classes. The proportion of samples allocated to the i-th class ($i \in \{0, 1, \ldots, K-1\}$) is defined as:

$$p_i = \frac{\gamma_{imb}^{\frac{i}{K-1}}}{\sum_{j=0}^{K-1} \gamma_{imb}^{\frac{j}{K-1}}}, \tag{95}$$

where $\gamma_{imb} \in (0, 1]$ is the imbalance factor. A smaller γ_{imb} induces a steeper exponential decay, resulting in a more extreme long-tailed distribution (see Figure 13 for visual examples). We maintain the total number of training samples equivalent to the original dataset by uniformly oversampling the allocated majority classes and undersampling the minority classes.



(a) $K = 10$ (CIFAR-10)

(b) $K = 100$ (CIFAR-100)

Figure 13: **Class proportion distributions under varying imbalance factors γ_{imb}.** As γ_{imb} decreases from 1.0 (balanced) to 0.001, the exponential decay becomes progressively steeper, exacerbating the severity of the long-tailed distribution.

Detailed Experimental Results. All models (Pre-Activation ResNet-18) on the imbalanced datasets were trained with a batch size of 128 using SGD (momentum $= 0.9$, weight decay $= 1 \times 10^{-4}$) for 200 epochs. The initial learning rate was 0.1, decayed by a factor of 0.1 at epochs 100 and 150.

We report **Macro-F1** and **Balanced Accuracy** to make class-level performance visible in addition to aggregate prediction accuracy. Both average class-specific quantities, so the contribution of a class to the reported average does not scale with its number of training examples.

Extended Visualizations. Figures 14 and 15 provide the comprehensive results for both CIFAR-10 and CIFAR-100 datasets across all evaluated metrics (Macro-F1 and Balanced Accuracy) under three distinct imbalance severities ($\gamma_{imb} \in \{0.1, 0.01, 0.001\}$). The plots compare the entire reported set of configurations, including Sombrero(Shuffle). Uniform target sampling improves several aggregate and majority-group results, while the most severe minority groups remain difficult. The numerical summaries below make this boundary explicit.

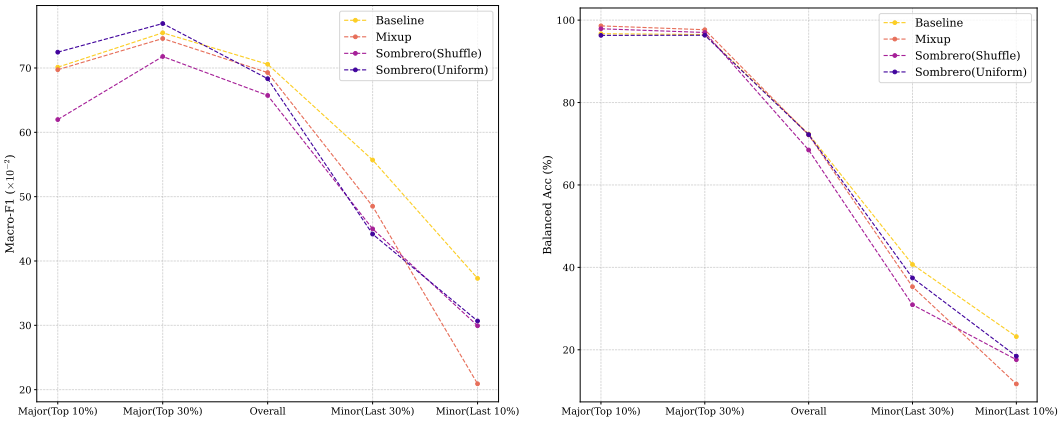
Numerical group summaries at severe imbalance. Table 11 reports group-level Macro-F1 values at $\gamma_{imb} = 0.001$. For CIFAR-100, the minority and majority groups are the bottom and top 10% of classes by training frequency. Uniform Sombrero improves the minority score relative to Mixup on both datasets but does not recover the ERM minority score. The full figures above also include the batch-shuffled Sombrero comparison, for which this numerical summary does not report additional values.

Table 11: **Reported Macro-F1 at $\gamma_{imb} = 0.001$.** Group-level point estimates complement the complete long-tail figures.

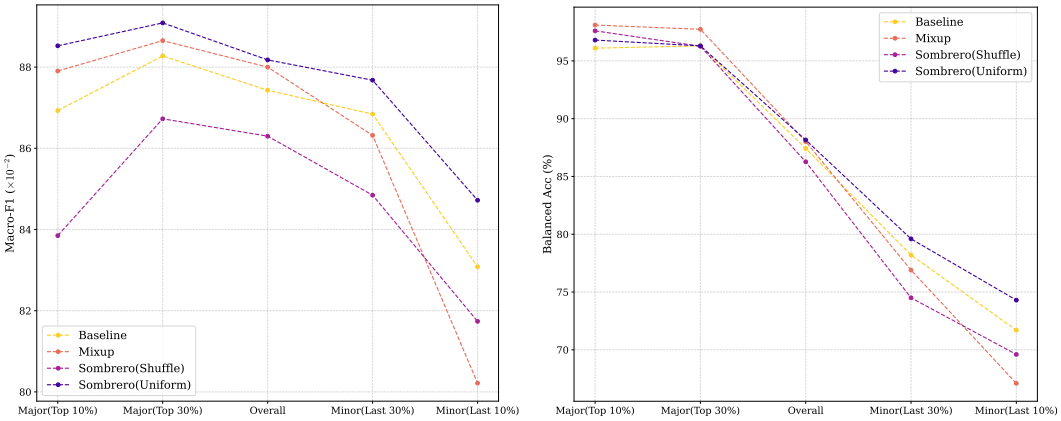
Dataset	Group	ERM	Mixup	Sombrero (Uniform)
CIFAR-100	Minority (bottom 10%)	5.10	1.75	2.86
CIFAR-100	Majority (top 10%)	60.62	56.41	66.23
CIFAR-100	Overall	42.66	43.56	43.44
CIFAR-10	Minority	37.30	20.91	30.68

D.5 Repeated Runs, Sensitivity, and Hyperparameter Selection

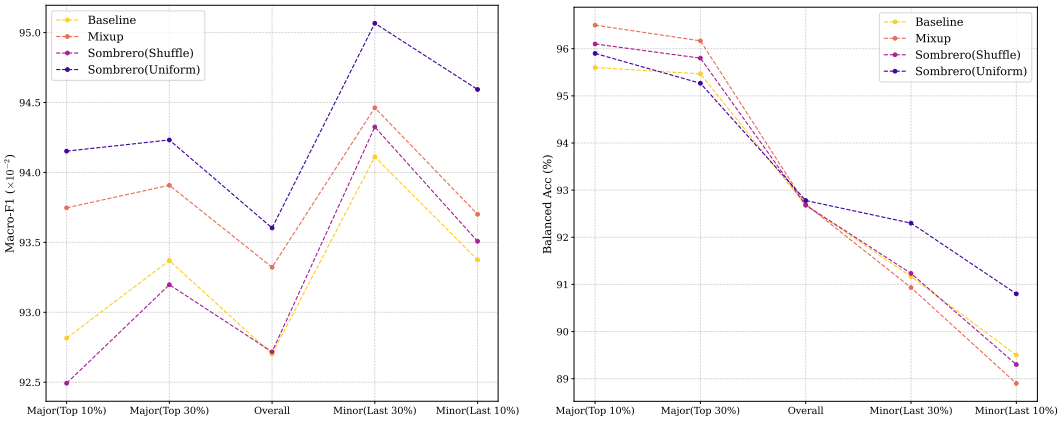
Reporting conventions. The architecture comparison (Table 4) and task-transfer tables report point estimates. The repeated-run summaries below measure variability across independent training seeds. The CIFAR-100 ERM reference is $76.23 \pm 0.42\%$ over five runs, reused in Tables 1, 3a, and 9 and the mixing-strength sweep. Other mechanism-ablation, translation, and detection entries remain single-run point estimates. In the classification sweep, perturbation targets are sampled



(a) Macro-F1, $\gamma_{imb} = 0.001$ (b) BalAcc, $\gamma_{imb} = 0.001$

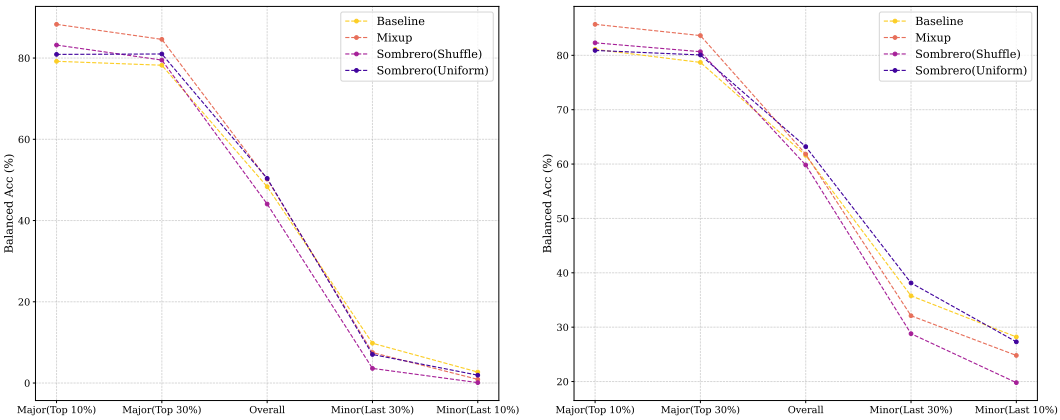


(c) Macro-F1, $\gamma_{imb} = 0.01$ (d) BalAcc, $\gamma_{imb} = 0.01$

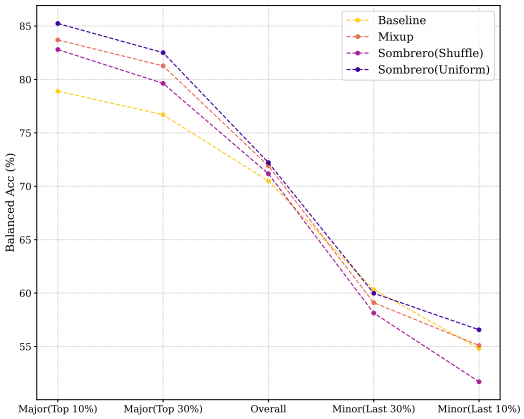


(e) Macro-F1, $\gamma_{imb} = 0.1$ (f) BalAcc, $\gamma_{imb} = 0.1$

Figure 14: **Extended evaluation on the long-tailed CIFAR-10 dataset.** Left: Macro-F1; right: Balanced Accuracy. Rows correspond to $\gamma_{imb} = 0.001, 0.01, 0.1$, respectively. Classes are grouped by frequency. Uniform sampling changes the distribution of performance across frequency groups; the minority-group score at $\gamma_{imb} = 0.001$ remains below ERM despite improving over Mixup.



(a) $\gamma_{imb} = 0.001$ (b) $\gamma_{imb} = 0.01$



(c) $\gamma_{imb} = 0.1$

Figure 15: **Balanced Accuracy evaluation on the long-tailed CIFAR-100 dataset.** Performance tracking across major and minor class splits under varying imbalance factors: $\gamma_{imb} \in \{0.001, 0.01, 0.1\}$. The results mirror the Macro-F1 trends presented in Figure 5.

uniformly from the class label space; the gradient probe in Appendix D.6 instead uses the specified permutation-paired construction.

Table 12: **Repeated-run summaries.** Top-1 accuracy (%), with means and standard deviations over the listed numbers of independent runs. Tables 1, 9, and 4 also include single-run results.

Dataset / model	Method	Seeds	Mean	Std.
CIFAR-100 / PreAct-ResNet-18	ERM	5	76.23	0.42
	Mixup ($\alpha = 1.0$)	5	77.94	0.56
	Manifold Mixup ($\alpha = 1.5$)	5	78.34	0.35
	Sombrero ($\alpha = 0.6$)	5	78.59	0.30
ImageNet-1K / ResNet-18	ERM	3	69.65	0.21
	Mixup	3	70.41	0.28
	Sombrero ($\alpha = 0.2$)	3	71.12	0.23

CIFAR comparison and sensitivity. The CIFAR-100 repeated-run comparison uses the same 200-epoch Pre-Activation ResNet-18 recipe as the main experiment. The Mixup/Sombrero sweep uses SGD with initial learning rate 0.1, momentum 0.9, and weight decay 10^{-4}; the learning rate is multiplied by 0.1 at epochs 100 and 150. Each of the 12 mixing strengths is evaluated with five independently initialized training seeds. Manifold Mixup selects α from $\{0.2, 0.5, 1.0, 1.5, 2.0, 2.5\}$, with 1.5 selected under this recipe. This schedule is shorter than the original Manifold Mixup schedule and its selected value differs from the original CIFAR recommendation of 2.0. The selected Mixup value is 1.0 and the Sombrero peak is at 0.6. Figure 4 shows the five-seed sweep, with all numerical entries in Table 13.

Table 13: **Numerical entries from the five-seed CIFAR-100 sweep.** Top-1 accuracy (%), mean±standard deviation. The ERM reference is $76.23 \pm 0.42\%$ over five runs.

α	Mixup	Sombrero
0.01	76.21 ± 0.16	76.03 ± 0.09
0.1	76.92 ± 0.37	77.62 ± 0.24
0.2	77.74 ± 0.15	77.91 ± 0.37
0.4	77.92 ± 0.33	78.40 ± 0.22
0.6	77.71 ± 0.46	78.59 ± 0.30
0.8	77.89 ± 0.43	77.90 ± 0.30
1.0	77.94 ± 0.56	77.87 ± 0.15
2.0	77.10 ± 0.38	77.66 ± 0.35
4.0	76.08 ± 0.79	77.61 ± 0.28
8.0	74.41 ± 0.72	77.33 ± 0.18
16.0	72.90 ± 0.96	77.45 ± 0.30
32.0	72.17 ± 1.26	77.50 ± 0.12

Selection protocols. Sombrero and its centered/decoupled variants, Mixup, and LossMix use grid searches on validation or proxy splits, after which their selected hyperparameters are fixed. The protocols are task-specific, as described next. On ImageNet, the Mini-ImageNet proxy search spans 0.1–0.5 for Sombrero and 0.1–0.25 for Dec-Sombrero, selecting 0.2 and 0.15, respectively. For detection, Sombrero and LossMix use the same validation-search protocol and budget, with Sombrero selecting $\alpha = 0.4$. The translation comparison evaluates $\alpha \in \{0.125, 0.15\}$. Entries in the external-reference block of Table 4 are cited results using their source protocols; they are not additional local training runs. ERM uses the stated architecture-specific recipes, including LS in the ShuffleNet baseline.

The relation $\varepsilon = \alpha/(2\alpha+1)$ supplies an interpretable mean-regularization scale: $\varepsilon = 0.1$ corresponds to $\alpha = 0.125$, and $\alpha = 0.6$ corresponds to $\varepsilon \approx 0.273$. The alignment-ratio analysis in Appendix C.5 additionally characterizes how the geometry model changes with α. In practice, proxy validation determines the setting for each task; the sensitivity results show why a neighborhood of values can be

useful rather than only a single optimum. The selected values are often in $[0.1, 0.4]$, while CIFAR-100 peaks near 0.6. Decoupled Sombrero permits separate adjustment of the relevant components as described in Appendix B.3.

D.6 Gradient Probe of the Paired First-Order Construction

We compare the full Mixup gradient with the permutation-paired first-order expression at fixed, trained parameters. The probe uses CIFAR-100 and Pre-Activation ResNet-18. For each α, we take the final checkpoint of one Mixup training run and evaluate 64 batches. Within each batch, both gradients use the same coefficient draw and permutation (λ, f); the retained expression includes the forward and inverse pairings (f, f^{-1}). Define

$$\mathrm{RelGap} = \frac{\|\mathbf{g}_{\mathrm{mix}} - \mathbf{g}_{1\mathrm{st}}\|_2}{\|\mathbf{g}_{\mathrm{mix}}\|_2}, \qquad \mathrm{Cosine} = \frac{\langle \mathbf{g}_{\mathrm{mix}}, \mathbf{g}_{1\mathrm{st}} \rangle}{\|\mathbf{g}_{\mathrm{mix}}\|_2 \|\mathbf{g}_{1\mathrm{st}}\|_2}. \tag{96}$$

Table 14 reports batchwise means and standard deviations. Its checkpoint accuracies are single-run values for the specific probed models, distinct from the five-seed accuracy summaries.

Table 14: **Full-gradient versus paired-expression probe.** RelGap and cosine are mean$\pm$standard deviation over 64 batches at one final Mixup checkpoint per α; these are not across-seed uncertainties. The final column is the Top-1 accuracy (%) of that checkpoint.

α	RelGap	Cosine	Checkpoint accuracy
0.2	0.80 ± 0.54	0.78 ± 0.22	77.63
0.6	1.14 ± 0.56	0.45 ± 0.32	77.78
1.0	1.40 ± 0.41	0.27 ± 0.26	77.81
2.0	1.71 ± 0.35	0.14 ± 0.20	77.10
8.0	1.94 ± 0.19	0.11 ± 0.12	74.73

The measurements distinguish a construction that realizes a selected first-order expression from a numerically close approximation to the complete Mixup update. The two gradients already differ at moderate α, and their directional agreement decreases across the tested settings. RelGap can exceed one because it measures vector discrepancy relative to the norm of the full gradient. For target-affine criteria and a samplewise predictor, Eq. (25) identifies the discrepancy as the batch average of input-interpolation residuals at the mixed targets. For a permutation-equivariant batch-dependent predictor, including training-mode batch normalization, the corresponding identity is the batch-matrix residual in Eq. (36); the map includes the batch statistics of each forward pass. The probe measures this finite-separation discrepancy rather than a local Taylor coefficient. The probe therefore describes the relation between the two update expressions without requiring their difference to be small. The utility of the retained structure is evaluated by the training results elsewhere in the paper.

E Extended Background and Related Work

E.1 Learning objectives and stochastic targets

Let $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, with inputs $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and targets $\boldsymbol{y}_i \in \mathcal{Y} \subseteq \mathbb{R}^K$. A predictor $\mathcal{F}(\cdot; \boldsymbol{\theta})$ produces scores $\boldsymbol{h}_i$, with per-example loss

$$\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta}) = \mathcal{C}(\mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\theta}), \boldsymbol{y}_i). \tag{97}$$

Write $\mathbf{g}_i = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta})$, $\mathbf{H}_i = \nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i; \boldsymbol{\theta})$, and $\mathbf{J}_i = \nabla_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\theta}) \in \mathbb{R}^{K \times p}$, where p is the parameter count. The Jacobian maps parameter perturbations to score perturbations. Subscript $\mathcal{B}$ denotes averages over a batch of size b; $\mathbf{1}_K$ and I_K denote the all-ones vector and identity matrix, and $\{\{\cdot\}\}$ preserves multiplicity.

Mixup. For $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ and a random batch permutation f, Mixup forms

$$\widetilde{\boldsymbol{x}}_i = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_{f(i)}, \qquad \widetilde{\boldsymbol{y}}_i = \lambda \boldsymbol{y}_i + (1-\lambda)\boldsymbol{y}_{f(i)}, \tag{98}$$

and minimizes $\mathcal{L}_{\mathcal{B}}^{(mix)} = b^{-1} \sum_i \mathcal{L}(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i; \boldsymbol{\theta})$. The paired derivation uses one shared λ and both f and its inverse, distinguished from independently sampled perturbation targets.

Label smoothing and target-affine gradients. Label smoothing replaces a one-hot classification target with

$$\boldsymbol{y}^{(LS)} = (1-\varepsilon)\boldsymbol{y} + \varepsilon\boldsymbol{\mu}, \qquad \boldsymbol{\mu} = \frac{1}{K}\mathbf{1}_K. \tag{99}$$

For softmax cross-entropy, $\nabla_{\boldsymbol{h}}\mathcal{C} = \mathrm{softmax}(\boldsymbol{h}) - \boldsymbol{y}$; for $\mathcal{C} = \frac{1}{2}\|\boldsymbol{h}-\boldsymbol{y}\|^2$, it is $\boldsymbol{h}-\boldsymbol{y}$. Both are affine in $\boldsymbol{y}$, allowing exact target interpolation at fixed input and separating the conditional mean gradient from target-sampling noise without requiring nearby labels.

Understanding Mixup through regularization and feature learning. Mixup instantiates vicinal risk minimization through input and target interpolation (Chapelle et al., 2000; Zhang et al., 2018). Zhang et al. (2021) expand its expected loss into empirical risk, first- and second-order regularization, and a remainder, connecting these terms to robustness and generalization. Carratino et al. (2022) exactly rewrite Mixup as transformed data with zero-mean perturbations, then characterize input, target, and cross-covariances and expected-risk penalties; the zero mean follows from their change of variables. Park et al. (2022) extend loss-based analysis to mixed-sample augmentation, relating mixing masks to input-gradient, input-Hessian, and first-layer regularization. Objective modifications offer complementary control: RegMixup (Pinto et al., 2022) adds mixed-sample loss to clean-sample cross-entropy to improve accuracy and uncertainty behavior, while Decoupled Mixup (Liu et al., 2023) adds a discriminative regularizer that removes competition between mixed classes in its probability calculation, helping exploit hard mixed samples. Both retain input mixing. Feature-learning theory also explains how mixing common and rare features can promote rare-feature learning in the model of Zou et al. (2023). We isolate a first-order gradient structure and realize it on original inputs, making its mean and fluctuations separately accessible.

Label smoothing and stochastic supervision. Label smoothing regularizes through the target distribution (Szegedy et al., 2016; Müller et al., 2019); Random Label Smoothing randomizes the correct-class probability and the allocation of remaining mass among other classes (dos Santos & Papa, 2024). Chidambaram & Ge (2025) connect LS and Mixup through a preference for low-variance features, including models where low-variance spurious features harm generalization. For cross-entropy, Sombrero belongs to this stochastic-soft-target family. Its distinguishing object is the retained Mixup gradient: the paired construction realizes it, and independent targets extend it under a specified sampling law. The mean-gradient identity and conditional covariance then expose separate regularization effects.

Stochastic optimization and curvature. Noise geometry can affect optimization beyond its mean (Zhu et al., 2019). Wu et al. (2022) connect SGD noise–curvature alignment to local stability, with formal results for overparameterized squared loss. Sharpness-Aware Minimization (SAM) (Foret et al., 2021) instead explicitly optimizes loss over a parameter neighborhood through approximate inner maximization. Sombrero induces anisotropic noise through its retention coefficient and target distribution, whose covariance we connect to GGN curvature under a classification model. Centered and decoupled variants separate mean and noise, while spectral measurements examine the learned geometry without an added parameter-space maximization.

Supervision interpolation and task transfer. LossMix (Vu et al., 2024) interpolates supervision at the loss level, evaluating different-target losses on mixed inputs and applying this construction to detection. Sombrero evaluates the retained structure on original inputs. Its detection and translation applications examine this objective-level mechanism beyond classification, with task-appropriate perturbation targets; classification supplies the detailed noise analysis.

F Extended Discussion and Future Directions

We isolate a first-order regularization kernel from Mixup and realize it on original inputs through Sombrero Loss. Its exact paired realization anchors an extension with independently chosen target priors. Uniform Sombrero exposes mean label smoothing and Jacobian-shaped noise; centered and decoupled variants enable interventions on these effects.

Mean-matched accuracy, spectra, and landscapes support regularization beyond average smoothing. Classification and strength scans show utility and stability in tested settings; detection, translation, and long-tail results show flexibility through preserved inputs and chosen target priors.

The target-interpolation and mean-gradient identities apply to criteria with target-affine prediction gradients; other target dependences require a separate derivation. The geometric account conditions on examples or batches and sampling rules. Proportionality to modeled GGN requires coefficient matching; extension to the full Hessian requires residual control. Full trajectories, reparameterizations, and detection assignment noise remain open directions.

Spatial mixing such as CutMix offers complementary gradient structures and domain priors. Comparisons with SAM could connect structured noise to explicit parameter-space regularization. Larger sequence and language models need suitable target perturbations and evaluation of their dynamics.