Suspicious Labels, Useful Text: Out-of-Fold Weighting for Few-Shot Text Regression

Anonymous authors
Paper under double-blind review

Abstract

When multiple raters provide labels, an unreliable label does not necessarily make its text useless. Many noisy-label methods reduce the weight of examples with large training losses, and some discard them. With limited data, removing an example also removes its text. We use an out-of-fold (OOF) text–label predictor as a proxy for label reliability: each example is scored by a model that was not trained on it. We use this score to reduce the example's loss weight while keeping its text in training. Our analysis shows that OOF prediction removes the direct effect of an example's label on its own residual and gives conditions under which corrupted examples tend to receive larger residuals than clean examples. On two related real multi-rater datasets, OOF weighting has higher mean R^2 than equal weighting and oracle deletion. Shuffling the weights across examples removes the gain. These results support reducing the influence of questionable labels without automatically discarding their text.

1 Introduction

Noise in few-shot text regression is often not a statement that an entire example is worthless. Rather, the label may be unreliable while the text still contains a rare, boundary, or not-yet-learned pattern. This situation is common when labels come from different raters: disagreement can reflect an ambiguous judgment, a rater-specific perspective, or a pattern that the student has not yet represented. A high training loss therefore mixes two facts that should be kept conceptually separate: the label may be poor supervision, while the input may still be useful.

Many noisy-label procedures reduce the influence of examples judged suspicious; some remove those examples entirely. This can also discard useful text, especially when data are limited. The central question of this paper is therefore:

> When a training label looks suspicious, should the corresponding input be removed as well?

We separate two questions: how much to trust a label, and whether to keep its text in training. OOF weighting uses residual ranks as a proxy for label reliability before student training and uses them only to reduce the example's loss contribution. Each example receives a score from a text–label predictor that was not trained on it. A large held-out residual signals a mismatch for this proxy, but does not by itself show that the text is useless. We convert residual ranks into continuous weights rather than a binary keep-or-remove decision.

We first show why OOF prediction prevents an example's own label from directly shrinking its residual, then bound the chance that a corrupted example receives a larger residual than a clean one. We compare OOF weighting with an oracle that knows which labels were corrupted and removes those examples. Finally, we shuffle weights across examples to test whether the gains depend on matching weights to examples.

Our contributions are threefold:

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.

1. We study whether label reliability alone should determine whether its text is kept in training.
2. We use OOF residual ranks to set fixed example weights before student training, and analyze how label perturbations affect these residuals and their ranking.
3. On two related real multi-rater datasets, OOF weighting has higher mean performance than equal weighting and oracle deletion; shuffling weights across examples removes the gain. Synthetic-noise experiments test this pattern on additional tasks.

2 Related Work

Our work is related to robust statistics, noisy-label learning, and sample reweighting. Robust losses reduce the influence of atypical observations through the loss or influence function (Huber, 1964; Tukey, 1960; Hampel, 1968; Rousseeuw & Leroy, 1987). Noisy-label methods address risk correction, noise modeling, or the behavior of deep networks under corrupted supervision (Natarajan et al., 2013; Frénay & Verleysen, 2014; Patrini et al., 2017; Goldberger & Ben-Reuven, 2017; Rolnick et al., 2017).

Another line of work selects or reweights examples during training, including curriculum learning, MentorNet, Co-teaching, learning-to-reweight, and Meta-Weight-Net (Bengio et al., 2009; Jiang et al., 2018; Han et al., 2018; Ren et al., 2018; Shu et al., 2019). These methods are close in spirit to our use of example weights. Our focus is whether estimated label reliability should also determine whether its text is removed. We compute the reliability score before student training with an OOF text–label proxy and compare against hard deletion.

The OOF construction connects cross-validation, Ridge regression, and cross-fitting (Stone, 1974; Hoerl & Kennard, 1970; Chernozhukov et al., 2018). TF–IDF and linear text models provide a transparent, low-cost proxy for text–label compatibility (Salton & McGill, 1983; Joachims, 1998; Wang & Manning, 2012). A supplementary check also evaluates a frozen MiniLM representation for the OOF proxy (Wang et al., 2020).

3 Estimating Label Reliability While Keeping Text

3.1 Problem setup

Let $\mathcal{D}_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is a text input and y_i is an observed continuous or ordinal label. There is a conceptual clean target y_i^*, but training observes y_i, which may be affected by rater disagreement or other corruption. A student f_θ is trained with

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} w_i \big(f_\theta(x_i) - y_i\big)^2. \tag{1}$$

The important question is what w_i means. If it is repeatedly updated from the student's own per-example loss, it can lose its ability to identify unreliable labels after the student memorizes them. In OOF weighting, w_i is a fixed estimate of label reliability computed before student training, while the student remains free to use the corresponding text.

3.2 OOF proxy for label reliability

We split the training data into $K = 5$ folds. For an example i in fold $k(i)$, a Ridge predictor is fit on all other folds using TF–IDF features:

$$\widehat{g}_{-k(i)} = \mathrm{Ridge}_\alpha(\mathrm{TFIDF}(x), y), \qquad \widehat{y}_i^{\mathrm{OOF}} = \widehat{g}_{-k(i)}(x_i). \tag{2}$$

The held-out residual is

$$r_i = \big|y_i - \widehat{y}_i^{\mathrm{OOF}}\big|. \tag{3}$$

We rank $-r_i$ in ascending order, using ranks $1, \ldots, N$, and normalize the resulting weights to have mean one:

$$\widetilde{w}_i = \frac{\mathrm{rank}(-r_i)}{N}, \qquad w_i = \frac{\widetilde{w}_i}{N^{-1} \sum_j \widetilde{w}_j}. \tag{4}$$

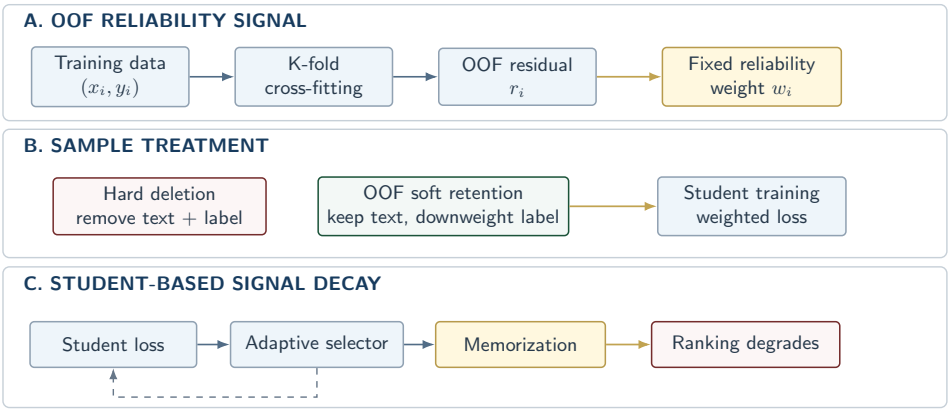


Figure 1: (A) OOF residuals yield fixed example weights. (B) OOF weighting reduces a label's loss contribution while keeping its text; hard deletion removes both. (C) Student-loss rankings can degrade during memorization.

A large residual indicates that the observed label is difficult for the held-out text–label proxy to explain. It therefore signals a potentially unreliable label while leaving the corresponding text available to the student. The proxy is independent of student parameters and training dynamics; it still uses observed training labels, but OOF removes the direct own-label fitting path. Consequently, the score measures how well the observed label agrees with this text proxy; it does not estimate whether the text is useful to the student.

3.3 Oracle hard deletion

The oracle receives the true corruption mask and sets the weights of substituted labels to zero, removing those examples from training. It is a privileged baseline because the corruption mask is unavailable in practice. Comparing against this baseline tests whether OOF weighting can usefully reduce label influence while keeping the text. A higher score is consistent with the text carrying useful information, although the supervised objective does not isolate text value from label effects.

3.4 Theory

Effect of a label on its own residual. For a centered design matrix $X \in \mathbb{R}^{n\times d}$, an in-sample Ridge predictor has hat matrix

$$H_\lambda = X(X^\top X + \lambda I)^{-1}X^\top. \tag{5}$$

Proposition 1 (Effect of a label perturbation on its Ridge residual). Fix X and $\lambda > 0$, and define the signed residuals $e_i^{\text{in}} = y_i - \widehat{y}_i^{\text{in}}$ and $e_i^{\text{OOF}} = y_i - \widehat{y}_i^{\text{OOF}}$. If only y_i is perturbed by Δ, the in-sample signed residual changes by $(1 - H_{\lambda,ii})\Delta$. If the fold containing i is excluded from the OOF fit, the OOF signed residual changes by Δ. The corresponding absolute residuals are $|e_i|$; their changes are not generally linear in Δ.

Thus, in-sample fitting absorbs part of a label perturbation according to leverage, while OOF prediction leaves the perturbation on the signed residual before taking its absolute value. A proof is given in Appendix A.

A residual-ranking condition. Suppose $y_i^* = x_i^\top \beta^* + u_i$, $y_i = y_i^* + z_i\varepsilon_i$, and $z_i \in \{0, 1\}$. Assume $\|x_i\| \le B$, $|u_i| \le \tau$, and $\|\widehat{\beta}_{-k(i)} - \beta^*\| \le \eta$. Let $A = \tau + B\eta$. The corruption ε_i may have any distribution, including a discrete or biased rater shift.

Theorem 1 (OOF residual ranking). For a random corrupted example i and a random clean example j,

$$\Pr\left(r_i^{\text{noisy}} > r_j^{\text{clean}}\right) \ge \Pr(|\varepsilon_i| > 2A). \tag{6}$$

If $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ is independent of (x_i, u_i) and the fold-external fit, the sharper bound $2[1 - \Phi(A/\sigma)]$ holds; it exceeds 0.5 when $\sigma > 1.48A$.

Because rank normalization is monotone, the same pairwise ordering is retained by the soft weights. The theorem identifies when a held-out residual provides a useful reliability ordering. Full proofs are in Appendix A.

When partial weighting can beat deletion. The ranking result alone does not imply that soft weighting improves the final student. We analyze the decision to retain a suspected group through a fixed feature map $\phi(x) \in \mathbb{R}^d$, which may be a frozen text encoder. Let X_C and X_N contain the clean and suspected feature rows, and assume

$$y_C = X_C\beta^* + \epsilon_C, \qquad y_N = X_N\beta^* + b + \epsilon_N. \tag{7}$$

Here b is an arbitrary fixed rater shift, ϵ_C has mean zero and covariance $\sigma_c^2 I$, and ϵ_N has mean zero and finite covariance Ω_N; the two error vectors are independent. Write $A_C = X_C^\top X_C \succ 0$, $B_N = X_N^\top X_N$, and $S_a = (A_C + aB_N)^{-1}$. Weighted least squares gives suspected examples a common weight $a \geq 0$, with oracle deletion at $a = 0$. For a positive semidefinite test-feature second-moment matrix Q, define $R_Q(a) = \mathbb{E}[(\widehat{\beta}_a - \beta^*)^\top Q(\widehat{\beta}_a - \beta^*) \mid X_C, X_N]$.

Proposition 2 (Risk under a systematic rater shift). Let $c = X_N^\top b$ and $M = X_N^\top \Omega_N X_N$. Then

$$R_Q(a) = \operatorname{tr}\big[QS_a\big(\sigma_c^2 A_C + a^2(M + cc^\top)\big)S_a\big]. \tag{8}$$

$$R_Q'(0) = -2\sigma_c^2 \operatorname{tr}\big(QA_C^{-1}B_N A_C^{-1}\big). \tag{9}$$

If $\sigma_c^2 > 0$ and this trace is positive, every sufficiently small $a > 0$ has $R_Q(a) < R_Q(0)$, even when $b \neq 0$.

The first-order gain comes from additional test-relevant feature directions; the effect of finite rater shift and noisy-label variance enters at second order in a. This result concerns a fixed representation and sufficiently small group weights. It neither guarantees an improvement for the actual OOF rank weights nor covers end-to-end Transformer training; those questions are tested empirically.

4 Experiments

4.1 Real multi-rater benchmarks and protocol

We focus the main paper on two real multi-rater text datasets. The first is the UC Berkeley Measuring Hate Speech corpus (Kennedy et al., 2020). We retain comments with at least three raters, obtaining 17,352 comments and 101,207 rater records; the target is the per-rater ordinal hatespeech label in $\{0, 1, 2\}$. The second is HateXplain (Mathew et al., 2021), where each post has three original labels: normal, offensive, or hatespeech, mapped to $0, 1, 2$.

For each seed, we split by comment or post identifier so that the same text cannot cross splits. On the training split, 30% of comments use one real rater's label in place of the multi-rater mean. Validation and test targets remain the multi-rater mean. This creates controlled rater substitution from real disagreement rather than synthetic label noise. We use five paired seeds, identical student initialization across methods, and paired tests against Equal.

For each dataset and seed, we use 5,000 examples: 3,000 for training, 1,000 for validation, and 1,000 for testing. Thus, the main experiments use a fixed-size subset of the 17,352 eligible Measuring Hate Speech comments.

The student is a four-layer Transformer regressor trained from scratch, with model dimension 256, four attention heads, feed-forward dimension 512, and maximum length 128. The vocabulary is built from training text; no pretrained text parameters are used. We compare equal weighting, Huber loss, Co-teaching, Meta-Weight-Net, OOF weightingwith a TF–IDF Ridge proxy, oracle hard deletion, an oracle based on true label deviations, and shuffled

Table 1: Main result and robust baselines on Measuring Hate Speech. Five paired seeds, 5,000 comments, and 30% rater substitution. The true-deviation oracle ranks examples using each substituted label's deviation from the all-rater mean.

Method	R^2	ΔR^2	Paired p vs Equal
Equal	0.3602 ± 0.0450	–	–
Huber loss	0.3471 ± 0.0455	-0.0131	0.0930
Co-teaching	0.3675 ± 0.0246	$+0.0073$	0.7042
Meta-Weight-Net	0.3690 ± 0.0523	$+0.0087$	0.4294
Oracle hard deletion	0.3697 ± 0.0277	$+0.0095$	0.3605
Oracle true-deviation rank	0.3876 ± 0.0343	$+0.0274$	0.0581
OOF weighting(Ridge proxy)	$\mathbf{0.4143 \pm 0.0451}$	$\mathbf{+0.0540}$	0.0098

Table 2: OOF-weight strength on the first real multi-rater benchmark.

Treatment	R^2	ΔR^2	Paired p
Equal	0.3602 ± 0.0450	–	–
Random hard deletion	0.3310 ± 0.0565	-0.0293	0.0827
OOF blend ($\lambda = 0.25$)	0.3654 ± 0.0381	$+0.0052$	0.6538
OOF blend ($\lambda = 0.50$)	0.3714 ± 0.0383	$+0.0112$	0.0973
OOF blend ($\lambda = 0.75$)	0.3864 ± 0.0358	$+0.0262$	0.0288
OOF ($\lambda = 1.00$)	0.4143 ± 0.0451	$+0.0540$	0.0098
Oracle hard deletion	0.3697 ± 0.0277	$+0.0095$	0.3605

OOF weights. All weights are computed before student training and held fixed, except for the adaptive baselines.

The primary protocol uses a clean validation target for checkpoint selection; audits using noisy validation and a fixed epoch are reported in Appendix B. Thus, the main comparisons use the same student, split, initialization, and training budget, while the only intended difference is how the observed training labels are treated.

For the privileged true-deviation reference, the training-side all-rater mean is used only to compute each substituted label's absolute deviation and rank the resulting weights. Validation and test targets do not enter this reference.

4.2 Main results

Table 1 compares OOF weightingwith robust-training baselines and two privileged references on the first benchmark. OOF weightinghas the highest mean R^2. The true-deviation oracle uses each substituted label's known deviation from the all-rater mean to set rank weights, information unavailable in ordinary training. OOF weightingalso has a higher mean than this oracle on both benchmarks; paired results are reported below. These comparisons support OOF weighting, but the supervised objective does not isolate text value from label effects.

To characterize sensitivity to attenuation strength, we interpolate between Equal and the OOF weights:

$$w_i(\lambda) = (1 - \lambda) + \lambda w_i^{\mathrm{OOF}}, \qquad \lambda \in \{0.25, 0.50, 0.75, 1.00\}. \tag{10}$$

Table 2 and Figure 2 show a monotone increase in mean performance across the tested strengths. Random hard deletion has a lower mean than Equal, but the paired difference is not conclusive ($p = 0.0827$); oracle deletion also has a lower mean than full OOF weighting.

The effective sample size is $\mathrm{ESS} = (\sum_i w_i)^2 / \sum_i w_i^2$. For the 3,000 training comments, the rank weights give $\mathrm{ESS} = 2250.4 \approx 0.750N$, whereas 30% oracle deletion gives $0.700N$.

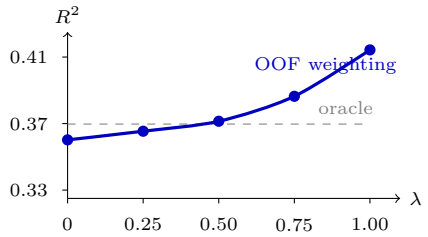


Figure 2: OOF-weighting curve on the first real multi-rater benchmark.

Table 3: Fixed-test small-sample scaling under 30% real-rater substitution. Values are five-seed means. The tuple is (R^2, ρ, MAE). OOF−Equal is a paired difference; lower MAE is better.

n_{train}	Equal	OOF Ridge	OOF−Equal
128	$(-.110, .117, .602)$	$(-.089, .100, .595)$	$(+.021, -.017, -.007)$
256	$(-.062, .197, .574)$	$(-.049, .178, .575)$	$(+.013, -.019, +.000)$
512	$(+.056, .313, .536)$	$(+.093, .327, .523)$	$(+.038, +.014, -.014)$
1000	$(+.181, .420, .489)$	$(+.238, .447, .471)$	$(+.058, +.027, -.019)$
2000	$(+.301, .486, .443)$	$(+.359, .528, .414)$	$(+.058, +.043, -.029)$
3000	$(+.373, .549, .408)$	$(+.432, .593, .378)$	$(+.059, +.044, -.030)$

Shuffled OOF has the same marginal weight distribution and ESS as OOF, but returns to Equal, showing that the ESS difference alone does not explain the result.

These results show that example-specific continuous weighting can outperform hard zeroing under this protocol. Oracle deletion has a lower mean R^2 than the full OOF weights in this experiment; the reported test does not directly assess this pairwise difference. Because all methods use a supervised loss, the comparison does not independently measure how much information the text contributes apart from its label.

4.3 Small-sample scaling under real rater substitution

The main benchmark uses 3,000 training comments, so we additionally vary the training size while keeping a fixed outer set of 5,000 comments and the same seed-specific validation/test split. For each size, 30% of the training comments receive a randomly selected real rater label, and all methods use the same nested training subset, student, and 40-epoch budget. Table 3 reports the three most interpretable metrics: R^2, Spearman rank correlation ρ, and MAE. The complete RMSE results and all per-seed values are in Appendix B.

The pattern is informative rather than uniformly positive. At 128 and 256 examples, the paired differences are noisy and Spearman correlation does not improve. From 512 examples onward, the OOF method improves all three metrics in aggregate; at 3,000 examples, the paired differences are $+0.059$ in R^2 ($p = 0.0011$), $+0.044$ in ρ ($p = 0.0020$), and -0.030 in MAE ($p = 0.0002$). Thus, the evidence supports a limited-data regime with a practical lower operating range, rather than a claim that OOF weighting is reliably beneficial at every extreme few-shot size.

4.4 Shuffled correspondence and second benchmark

Table 4 shows the shuffled control. Keeping the weight values but breaking their correspondence with examples removes the gain. The result therefore depends on a reliability ordering tied to the examples, not merely on using a continuous weight distribution.

We repeat the same protocol on HateXplain, and Table 5 adds the same baselines and oracle references. OOF weighting has the highest mean R^2 on this benchmark too. In the five-seed paired comparison, its difference from Co-teaching is not conclusive ($p = 0.0746$); differences against Huber, Meta-Weight-Net, and the true-deviation oracle are more consistent. These

Table 4: Reliability and shuffled-weight controls on the first real multi-rater benchmark.

Method	R^2	ΔR^2	Paired p	ESS $/N$
Equal	0.3602 ± 0.0450	–	–	1.000
Oracle hard deletion	0.3697 ± 0.0277	$+0.0095$	0.3605	0.700
OOF weighting(Ridge proxy)	0.4143 ± 0.0451	$+0.0540$	0.0098	0.750
Shuffled OOF weights	0.3594 ± 0.0465	-0.0008	0.8571	0.750

Table 5: Second real multi-rater benchmark: HateXplain. Five paired seeds and 30% rater substitution. The true-deviation oracle ranks examples using the known training-label deviation.

Method	R^2	ΔR^2	Paired p vs Equal
Equal	0.3365 ± 0.0262	–	–
Huber loss	0.3392 ± 0.0406	$+0.0028$	0.7234
Co-teaching	0.3523 ± 0.0351	$+0.0158$	0.0768
Meta-Weight-Net	0.3287 ± 0.0321	-0.0078	0.3225
Oracle hard deletion	0.3340 ± 0.0292	-0.0025	0.5167
Oracle true-deviation rank	0.3501 ± 0.0316	$+0.0136$	0.0204
OOF weighting(Ridge proxy)	$\mathbf{0.3749 \pm 0.0186}$	$\mathbf{+0.0384}$	0.0006

unadjusted comparisons support the result but do not establish a definitive ranking of methods.

Against the true-deviation oracle, the paired R^2 gain of OOF weightingis $+0.0266$ on Measuring Hate Speech (95% paired-t CI $[0.0111, 0.0421]$, two-sided $p = 0.0088$) and $+0.0248$ on HateXplain (95% CI $[0.0053, 0.0442]$, $p = 0.0241$). These are five-seed paired intervals and tests; they compare the OOF proxy with a privileged rank based on known training-label deviations.

We also compare with an oracle soft mask that knows which training labels were substituted while keeping every text input. At each tested blend strength $\lambda \in \{0.25, 0.50, 0.75\}$, OOF weightinghas higher mean R^2 on both benchmarks; the six paired OOF-minus-mask differences range from $+0.0295$ to $+0.0569$. Their exploratory 95% paired intervals exclude zero without multiple-comparison adjustment. Full results are in Table 12 of Appendix B.

4.5 Cross-dataset checks under synthetic noise

The two primary benchmarks are related hate-speech datasets, so they test replication across corpora rather than generalization across annotation domains. We therefore report two secondary checks, while keeping their limitations explicit. Under synthetic label corruption, OOF improves over Equal on Wine, Yelp, and Amazon by $+0.0490$, $+0.0870$, and $+0.0672$ respectively (three seeds per dataset; paired $p = 0.0182$, 0.0026, and 0.0737). These results broaden the task types but do not establish generalization to naturally occurring multi-rater disagreement.

An additional proxy-representation diagnostic with frozen MiniLM is reported in Appendix B.

5 Conclusion

We studied how a student-independent OOF text–label compatibility score can set example weights under controlled rater substitution. OOF has the highest mean R^2 among the compared methods on both real multi-rater benchmarks and exceeds Equal on both; paired evidence against the privileged true-deviation rank is also positive. With five seeds, compar-

isons against some alternative training methods are less conclusive, so we treat the results as evidence for this protocol rather than a definitive ranking. Shuffling the weights removes the gain, indicating that example-level correspondence matters. Since all compared students use a weighted supervised loss, these experiments do not independently establish how useful the text is apart from its label.

The small-sample scaling experiment identifies a practical boundary: under the present student and rater-substitution protocol, gains are inconsistent at 128 and 256 examples and become positive in aggregate from 512 examples onward. The theory gives a distribution-free proxy-ranking bound and a fixed-feature risk comparison that permits systematic rater shift. The latter still requires homoskedastic clean-label error and does not analyze the trained Transformer. The two primary datasets both concern hate speech, and the broader task checks use synthetic corruption. Evaluation on additional real multi-rater domains remains important.

AI use statement

We used generative AI tools to assist with translating the Chinese manuscript into English, improving English phrasing, reorganizing exposition, and migrating the manuscript to the ICLR 2027 LaTeX template. Generative AI tools were not used to generate the experimental code, datasets, numerical results, statistical tests, or the underlying theoretical assumptions and mathematical argument; they assisted with the English exposition of material already developed by the authors. The experiments were run with the authors' existing scripts and computing environment. The authors manually reviewed the AI-assisted text, equations, tables, numerical values, citations, and formatting, checked the cited works against the bibliography and source records, and take responsibility for the final content of the paper, including all text, claims, and artifacts produced with AI assistance.

Ethics statement

This work uses publicly available text datasets containing offensive and hateful language. We do not collect new human-subject data or infer identities, and we use comment-level identifiers only to prevent the same text from crossing data splits. Because annotation disagreement and model error may reflect demographic or annotator-specific bias, the method should not be interpreted as a fairness or moderation guarantee. We report the evaluation protocol and its limitations explicitly and recommend that any downstream deployment include domain-specific bias, privacy, and safety review.

Reproducibility statement

The main paper specifies the OOF construction, weighting rule, student architecture, data-splitting protocol, rater-substitution protocol, and statistical comparison. Appendix A gives complete proofs, and Appendix B gives the data-construction and leakage-prevention procedure, per-seed results, supplementary controls, and protocol audits. The accompanying scripts and materialized-data instructions provide the execution path for reproducing the reported tables and figures.

Author Contributions

All authors contributed to the problem formulation, experimental design, analysis, and manuscript preparation.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback.

References

Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, 2009.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal, 21(1):C1–C68, 2018.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems, 25(5):845–869, 2014.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In International Conference on Learning Representations, 2017.

Frank R. Hampel. Contributions to the theory of robust estimation. Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete, 4:73–101, 1968.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Advances in Neural Information Processing Systems, 2018.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970.

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

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Proceedings of the 35th International Conference on Machine Learning, pp. 2304–2313, 2018.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In European Conference on Machine Learning, pp. 137–142, 1998.

Chris J. Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. Measuring a hate speech spectrum with faceted rasch item response theory and perspective-aware, explainable-by-design deep learning. arXiv preprint arXiv:2009.10277, 2020. doi: 10.48550/arXiv.2009.10277.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 14867–14875, 2021. doi: 10.1609/aaai.v35i17.17745.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In Advances in Neural Information Processing Systems, pp. 1196–1204, 2013.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 3982–3992, 2019.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In Proceedings of the 35th International Conference on Machine Learning, pp. 4334–4343, 2018.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694, 2017.

Peter J. Rousseeuw and Annick M. Leroy. Robust Regression and Outlier Detection. Wiley, 1987.

Gerard Salton and Michael J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019. doi: 10.48550/arXiv.1910.01108.

Jun Shu, Qi Xie, Liyang Yi, Qian Zhao, Sanjun Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In Advances in Neural Information Processing Systems, 2019.

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society: Series B (Methodological), 36(2):111–147, 1974.

John W. Tukey. A survey of sampling from contaminated distributions. In Ingram Olkin, S. G. Ghurye, Wassily Hoeffding, William G. Madow, and Henry B. Mann (eds.), Contributions to Probability and Statistics, pp. 448–485. Stanford University Press, 1960.

Sida Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pp. 90–94, 2012.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In Advances in Neural Information Processing Systems, volume 33, pp. 5776–5788, 2020.

A Proofs

A.1 Proof of Proposition 1

Let $\boldsymbol{e}_i$ be the ith standard basis vector and write the perturbation as $\boldsymbol{y}' = \boldsymbol{y} + \Delta \boldsymbol{e}_i$. For fixed X, the in-sample Ridge fitted values are a linear map,

$$\widehat{\boldsymbol{y}}' = H_\lambda \boldsymbol{y}' = \widehat{\boldsymbol{y}} + \Delta H_\lambda \boldsymbol{e}_i. \tag{11}$$

Thus the ith fitted value changes by $H_{\lambda,ii}\Delta$, and

$$e_i^{\text{in}\prime} = (y_i + \Delta) - \big(\widehat{y}_i + H_{\lambda,ii}\Delta\big) = e_i^{\text{in}} + (1 - H_{\lambda,ii})\Delta. \tag{12}$$

The eigenvalues of H_λ are $s_q^2/(s_q^2 + \lambda)$ (and zeros) under a singular-value decomposition of X, so $0 \le H_{\lambda,ii} < 1$. The in-sample residual therefore mechanically attenuates an own-label perturbation according to leverage. If i belongs to fold I_k, the OOF coefficient $\widehat{\beta}_{-k}$ is fitted only on labels outside I_k. Changing y_i leaves $x_i^\top \widehat{\beta}_{-k}$ unchanged, and hence

$$e_i^{\text{OOF}\prime} = (y_i + \Delta) - x_i^\top \widehat{\beta}_{-k} = e_i^{\text{OOF}} + \Delta. \tag{13}$$

This proves the proposition. □

A.2 Proof of Theorem 1

For each sample q, define

$$a_q = u_q + x_q^\top(\beta^* - \widehat{\beta}_{-k(q)}). \tag{14}$$

By Cauchy–Schwarz and the assumptions,

$$|a_q| \le |u_q| + \|x_q\|\,\|\widehat{\beta}_{-k(q)} - \beta^*\| \le \tau + B\eta = A. \tag{15}$$

Consequently, every clean sample j has $r_j^{\text{clean}} = |a_j| \le A$, while a corrupted sample satisfies $r_i^{\text{noisy}} = |a_i + \varepsilon_i| \ge |\varepsilon_i| - A$. Thus $|\varepsilon_i| > 2A$ implies that the corrupted residual exceeds every clean residual, without a distributional assumption. This proves the first bound. Under the additional Gaussian assumption, condition on (x_i, u_i) and the fold-external training data so that a_i is fixed. For any $a_i \in [-A, A]$, the event $|a_i + \varepsilon_i| \le A$ is the interval $[-A - a_i, A - a_i]$, whose length is $2A$. A centered Gaussian density is symmetric and decreases with distance from zero, so among all intervals of this length, $[-A, A]$ has maximal probability mass. Therefore

$$\Pr(|a_i + \varepsilon_i| > A) \ge \Pr(|\varepsilon_i| > A) = 2\,(1 - \Phi(A/\sigma))\,. \tag{16}$$

Whenever this event holds, the noisy residual exceeds the residual of every clean sample. Taking expectations over the fold-external fit gives the stated lower bound. Since ties have probability zero under continuous Gaussian noise, this pairwise probability is also the residual-ranking AUC between a random corrupted and a random clean example. Finally, rank normalization is monotone and therefore preserves the ordering. This proves the Gaussian refinement and the theorem. □

A.3 Proof of Proposition 2

The normal equations and the observation model give

$$\widehat{\beta}_a - \beta^* = S_a\big(X_C^\top \epsilon_C + ac + aX_N^\top \epsilon_N\big). \tag{17}$$

Its conditional mean is $aS_a c$, and independence gives conditional covariance $S_a(\sigma_c^2 A_C + a^2 M)S_a$. Adding squared bias to variance and taking the quadratic test risk yields the displayed expression for $R_Q(a)$. Since $S_a' = -S_a B_N S_a$, $S_0 = A_C^{-1}$, and the derivative of the term multiplied by a^2 vanishes at zero,

$$R_Q'(0) = \sigma_c^2 \left.\frac{\mathrm{d}}{\mathrm{d}a} \operatorname{tr}(Q S_a A_C S_a)\right|_{a=0} \tag{18}$$

$$= -2\sigma_c^2 \operatorname{tr}(Q A_C^{-1} B_N A_C^{-1}). \tag{19}$$

Both Q and $A_C^{-1} B_N A_C^{-1}$ are positive semidefinite, so the trace is nonnegative. When it is positive, differentiability and a strictly negative derivative imply $R_Q(a) < R_Q(0)$ for all sufficiently small $a > 0$. Finite b and Ω_N ensure that their terms remain second order as $a \downarrow 0$. □

B Experimental details and supplementary results

B.1 Data construction and leakage prevention

The Measuring Hate Speech records are grouped by comment_id and retain the per-rater ordinal hatespeech labels when at least three raters are available. The clean target is the mean over raters for the same comment. HateXplain posts are grouped by post identifier and retain the three original labels. For every seed, splitting happens before rater substitution; the OOF proxy sees only training comments and training observed labels; checkpoint selection uses validation data; and the test multi-rater mean is used only for final evaluation.

More explicitly, for each seed we (i) shuffle and split comment identifiers into train, validation, and test; (ii) perform rater substitution only on training comments; (iii) fit every OOF proxy using training text and training observed labels only; (iv) select a checkpoint on the validation split or use the prespecified fixed epoch; and (v) evaluate once on held-out comments whose multi-rater means were never used to construct a weight. Thus, different rater records for one comment cannot leak across splits, and test targets do not enter the weight function, hyperparameter selection, or checkpoint selection.

The main benchmark uses seeds 42–46 and paired comparisons on the same split and initialization:

All methods use the same splits, batch size, learning rate, epoch budget, and student architecture. The supplementary scans use seeds 42–44 and 20 epochs. The full execution

Table 6: Per-seed test R^2 on the first real multi-rater benchmark.

Seed	Equal	Oracle hard deletion	OOF weighting
42	0.4064	0.3935	0.4682
43	0.3020	0.3445	0.3636
44	0.3235	0.3357	0.3956
45	0.3885	0.3937	0.4551
46	0.3808	0.3813	0.3888

order is: split by comment identifier, substitute a training rater, compute OOF weights using training data only, select a checkpoint on validation data, and evaluate once on the held-out multi-rater mean.

All primary comparisons use the same student initialization, optimizer budget, and training data order within a seed. The real multi-rater main experiment uses seeds 42–46; the stronger-student and supplementary baseline checks use the same paired seeds when available. The JSON results, data-materialization script, training scripts, and aggregation scripts are the artifacts used to produce the reported tables and figures.

B.2 Small-sample curve details

The fixed-test scaling experiment uses the same 5,000-comment outer sample for all sizes and takes nested subsets of each seed's training pool. The complete mean ± standard deviation summary for RMSE is:

Table 7: RMSE in the fixed-test small-sample experiment.

n_{train}	Equal	OOF Ridge	Oracle deletion
128	0.7229 ± 0.0254	0.7163 ± 0.0123	0.7194 ± 0.0214
256	0.7074 ± 0.0224	0.7029 ± 0.0227	0.7111 ± 0.0267
512	0.6665 ± 0.0353	0.6532 ± 0.0284	0.6727 ± 0.0300
1000	0.6212 ± 0.0155	0.5989 ± 0.0062	0.6291 ± 0.0172
2000	0.5739 ± 0.0145	0.5493 ± 0.0099	0.5728 ± 0.0172
3000	0.5439 ± 0.0241	0.5177 ± 0.0186	0.5369 ± 0.0211

For transparency, the $n_{\text{train}} = 3000$ per-seed audit is shown below; each entry is R^2/ρ:

Table 8: Per-seed R^2/Spearman audit for the fixed-test $n = 3000$ curve.

Seed	Equal	OOF Ridge	Oracle deletion
42	0.3314/0.5055	0.4058/0.5589	0.3723/0.5373
43	0.3978/0.5755	0.4367/0.5966	0.3961/0.5561
44	0.4119/0.5680	0.4575/0.6223	0.4567/0.6078
45	0.3547/0.5322	0.4225/0.5812	0.3373/0.5312
46	0.3679/0.5622	0.4364/0.6048	0.3774/0.5451

B.3 OOF proxy representation diagnostic

Table 9 compares proxy representations on the primary real multi-rater benchmark. Both TF–IDF and frozen MiniLM produce positive mean gains over Equal; the MiniLM comparison is less conclusive with five paired seeds.

The TF–IDF OOF weights have AUC 0.6232 against the known corruption mask on the first real benchmark; frozen MiniLM has AUC 0.6186. Both proxies therefore capture a useful, but imperfect, reliability ordering.

Table 9: Proxy representation diagnostic on the primary real multi-rater benchmark. Values are paired ΔR^2 over Equal.

Benchmark	Proxy or treatment	ΔR^2	Paired p
Real multi-rater	TF–IDF OOF	+0.0540	0.0098
Real multi-rater	Frozen MiniLM OOF	+0.0432	0.0543

B.4 Synthetic and cross-dataset results

These experiments extend the mechanism across synthetic and additional text datasets. On Wine with Gaussian corruption, OOF improves over Equal by +0.0467 ($p = 0.0142$). Across Wine, Yelp, and Amazon, the paired improvements are +0.0490, +0.0870, and +0.0672, with three seeds per dataset and paired p values 0.0182, 0.0026, and 0.0737, respectively.

With the student architecture and training budget fixed on Wine, OOF remains above Equal as the independent Gaussian corruption rate varies from 15% to 70%, providing a direct operating-range check for the mechanism. The full values and corresponding curve are reported below.

Table 10: Cross-dataset OOF results.

Dataset	Seeds	ΔR^2	Paired p
Wine	3	+0.0490	0.0182
Yelp	3	+0.0870	0.0026
Amazon	3	+0.0672	0.0737

On Wine, OOF remains above Equal at Gaussian corruption rates from 15% to 70%:

Table 11: Noise-rate sweep on Wine.

Noise rate	Equal	OOF	ΔR^2	p
15%	0.3039	0.3374	+0.0335	0.0003
30%	0.2567	0.3034	+0.0467	0.0142
50%	0.2066	0.2845	+0.0780	0.0017
70%	0.1800	0.2443	+0.0643	0.0013

B.5 Oracle-mask soft-weight controls

To separate knowing which labels were substituted from estimating a graded text–label signal, we keep every training text and set the oracle base weight to one for unmodified examples and zero for substituted examples. We blend this mask with Equal using the same λ values as the OOF strength scan, then normalize weights to mean one. The substituted-label weight relative to the unmodified-label weight is $1 - \lambda$. Table 12 reports five-seed means and paired differences from OOF. The intervals are unadjusted exploratory paired-t intervals across the three λ values and two benchmarks.

B.6 Training-dynamics and baseline controls

The main comparison in Table 1 includes Huber, Co-teaching, and Meta-Weight-Net on the first benchmark; none shows a reliable paired gain over Equal in this five-seed run. Current-student loss rankings also evolve during memorization, whereas the OOF proxy is fixed; the corresponding diagnostic is shown in Figure 4.

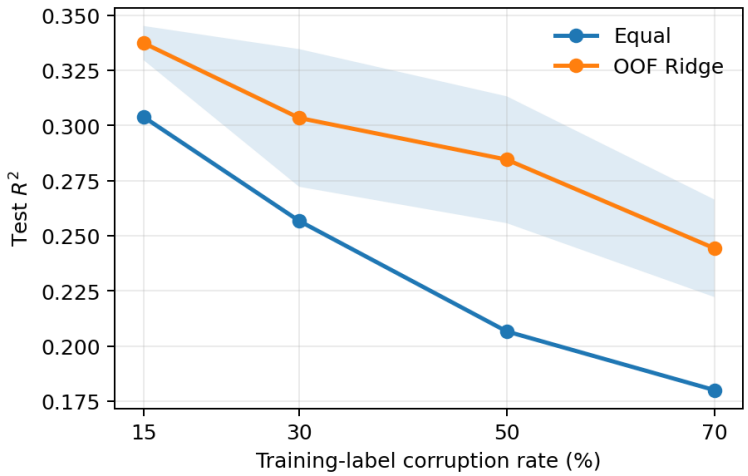


Figure 3: OOF remains above Equal across Gaussian corruption rates on Wine.

Table 12: Oracle-mask weighting on both real multi-rater benchmarks. All text inputs remain in training; only the known substituted-label mask determines the weights.

Benchmark	λ	Corrupt-label weight ratio	R^2	Paired ΔR^2 vs OOF (95% CI)
Measuring Hate Speech	0.25	0.75	0.3645 ± 0.0399	-0.0498 $[-0.0921, -0.0075]$
Measuring Hate Speech	0.50	0.50	0.3574 ± 0.0367	-0.0569 $[-0.1076, -0.0061]$
Measuring Hate Speech	0.75	0.25	0.3781 ± 0.0365	-0.0362 $[-0.0714, -0.0010]$
HateXplain	0.25	0.75	0.3428 ± 0.0314	-0.0321 $[-0.0525, -0.0116]$
HateXplain	0.50	0.50	0.3439 ± 0.0321	-0.0310 $[-0.0532, -0.0088]$
HateXplain	0.75	0.25	0.3454 ± 0.0302	-0.0295 $[-0.0473, -0.0118]$

The rank-weighting scan gives mean R^2 values 0.3758, 0.4060, and 0.3878 for alpha 0.1, 1, and 10. Under alpha 1, inverse and exponential weights give 0.3645 and 0.4081. Clean-validation, noisy-validation, and fixed-epoch protocols give three-seed mean R^2 values 0.4091, 0.4058, and 0.4151.

These checks distinguish the fixed OOF signal from a particular weighting parameterization or checkpoint rule. The current-student loss ranking changes over training as memorization develops, whereas the OOF signal is computed once before student training and remains fixed.

Table 13: Checkpoint-selection protocol audit.

Protocol	Mean R^2	Seeds
Clean validation	0.4091	3
Noisy validation	0.4058	3
Fixed epoch	0.4151	3

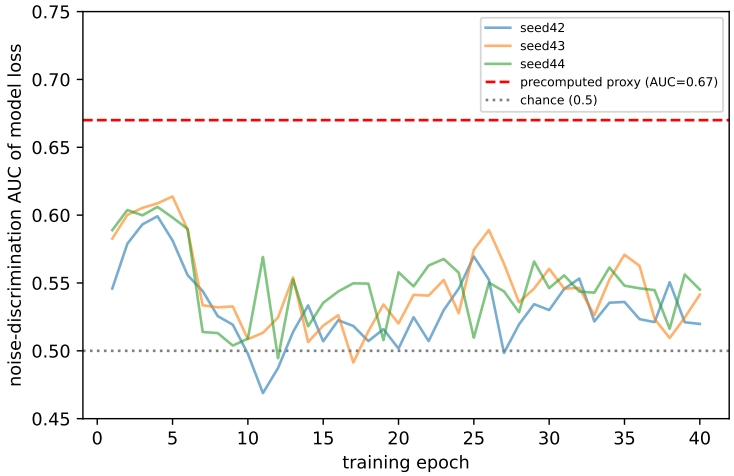


Figure 4: Training-loss identification of corruption changes during memorization, while the precomputed OOF proxy remains fixed.