WHAT CAN THE REPRESENTATION SPACE REVEAL ABOUT MEMORIZATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Memorization plays an important role in how neural networks fit rare and atypical training examples, but measuring per-example memorization is computationally expensive: estimating the Feldman–Zhang memorization score requires training many models. Existing practical proxies primarily rely on losses, gradients, or loss-surface geometry. We investigate whether memorization is directly reflected in the geometry of learned representations. We propose a single-training-run proxy that tracks, throughout training, how strongly an example's position relative to class centroids in representation space disagrees with its observed label. Under explicit assumptions, we relate this trajectory-averaged disagreement to Feldman–Zhang memorization through a sandwich bound, linking our score to memorization weighted by the fraction of training before the sample's representation becomes consistently aligned with its assigned class. Empirically, our score achieves the strongest agreement with Feldman–Zhang memorization among the evaluated proxies on ImageNet, while remaining competitive on CIFAR-100. The same signal is highly effective for mislabel detection and true-label recovery. For mislabel detection on CIFAR-100 under within-superclass label flips (asymmetric noise), our method achieves the highest AUROC across nearly all evaluated noise rates. Its AUROC decreases by less than one percentage point as the noise rate increases from 5% to 30%, compared with drops of up to approximately twelve percentage points for competing methods. These results suggest that memorization leaves a persistent and measurable geometric signature in learned representations.

1 INTRODUCTION

Deep neural networks are known for their strong generalization ability, yet memorization can play an important role in achieving this performance. In particular, Feldman & Zhang (2020) argue that for long-tailed distributions, where rare and atypical examples constitute a substantial portion of the data, memorizing such examples can be necessary for achieving near-optimal generalization. Identifying memorized examples is therefore important: these examples can be influential and valuable to the model, and removing them may degrade predictive performance. At the same time, Membership Inference Attacks (MIAs) (Shokri et al., 2017) are particularly effective on memorized examples (Carlini et al., 2022a;b; Kim et al., 2026), and Ravikumar et al. (2026) report a monotonic increase in MIA Area Under the Receiver Operating Characteristic curve (AUROC) with the Feldman–Zhang memorization score. Identifying which examples a model has memorized therefore matters both for understanding generalization and for assessing privacy risk.

Feldman & Zhang (2020) and Feldman (2020) provide empirical evidence that memorization is associated with atypicality. Here, we use atypicality to refer to examples that deviate from the dominant patterns of their class or data distribution, including rare, ambiguous, outlier, and mislabeled samples. In particular, high-memorization samples are frequently found among atypical, ambiguous, outlier, and mislabeled examples, whereas low-memorization samples are typically more prototypical. This pattern supports the hypothesis that memorization plays an important role in fitting examples from the long tail of the data distribution.

Feldman & Zhang (2020) define the memorization of a training example $(x_i, y_i) \in S$ under a learning algorithm $\mathcal{A}$ as

$$\mathrm{mem}(\mathcal{A}, S, i) := \Pr_{f \leftarrow \mathcal{A}(S)}[f(x_i) = y_i] - \Pr_{f \leftarrow \mathcal{A}(S \setminus i)}[f(x_i) = y_i]\,. \tag{1}$$

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.

Here, $S \setminus i$ denotes the dataset obtained by removing (x_i, y_i) from S, and the probabilities are taken over the randomness of the training algorithm. Directly estimating this quantity is computationally prohibitive. Feldman & Zhang (2020) therefore introduce a subsampled estimator, which is still expensive and requires training thousands of models. This computational cost has motivated substantially cheaper *proxies*, many of which can be obtained from a single training run. There are different kinds of proxies. Some methods are based on learning dynamics and can be used as memorization proxies. Toneva et al. (2019) characterize examples using forgetting events. VoG measures example difficulty using gradient variability over training (Agarwal et al., 2022), whereas TracIn estimates the influence of individual training examples by tracing gradient interactions across training checkpoints (Pruthi et al., 2020).

Explicit memorization proxies predominantly rely on loss-derived signals, their derivatives, or loss-surface geometry. Arpit et al. (2017) propose loss sensitivity as a gradient-based proxy for memorization, Agiollo et al. (2024) exploit the discrepancy between sharpness-aware minimization and stochastic gradient descent through loss-surface geometry, Garg et al. (2024) use input-loss curvature, Ravikumar et al. (2025) introduce cumulative sample loss (CSL), and Ravikumar et al. (2026) propose cumulative sample gradient and sample gradient-assisted loss (CSG/SGAL). This raises a natural question that motivates our work: Can the memorization signal live in other spaces?

Our idea is inspired by Neural Collapse (Papyan et al., 2020). That work shows that, in the terminal phase of training, the penultimate representations of samples in a class collapse toward their class mean, so within-class variability shrinks as representations approach their class means. Since highly memorized points are often atypical, we hypothesize that their representation trajectories may deviate from this dominant within-class behavior, remaining farther from and less aligned with their class centroid. This intuition motivates our method. We propose a memorization proxy that operates entirely in representation space: for each training sample, we form a distance-induced class probability from its distances to the class centroids and average the resulting label-disagreement probability over training. We evaluate our approach on CIFAR-10, CIFAR-100, and ImageNet.

In summary, our contributions are threefold. First, we show that per-example memorization is strongly reflected in learned representation geometry and introduce a single-training-run proxy based on trajectory-averaged representation–label disagreement. Under explicit assumptions, we relate this score to Feldman–Zhang memorization through a sandwich bound. Empirically, it achieves the strongest agreement with Feldman–Zhang memorization among the evaluated proxies on ImageNet, while remaining competitive on CIFAR-100. Representation–label disagreement remains informative after instantaneous training loss has largely collapsed. Second, we show that the same representation-space signal is effective for identifying mislabeled examples. To the best of our knowledge, we provide the first evaluation of per-example memorization proxies for mislabeled-sample detection under structured *asymmetric* label noise, where corrupted labels are restricted to classes within the same superclass as the original label. Our score remains particularly stable as the corruption rate increases. Third, our class-resolved geometry enables true-label recovery of mislabeled samples, a capability not directly provided by existing scalar memorization scores.

2 Related Work and Background

Estimating memorization. The per-example memorization notion we target was introduced theoretically by Feldman (2020) and studied empirically at scale by Feldman & Zhang (2020), who developed a subsampling-based estimator and linked high memorization to atypical long-tail examples. We use the Feldman–Zhang memorization estimates as our reference target, but computing them faithfully requires training thousands of models. Jiang et al. (2021) introduce the C-score, which measures structural regularity through expected held-out accuracy across training-set sizes, and study efficient approximations based on learning speed and representation-space distances. Although related to the held-out component of Feldman–Zhang memorization, C-score estimates the difficulty of learning a specific example rather than directly estimating memorization.

Because estimating the Feldman–Zhang score requires thousands of training runs, subsequent work has developed substantially cheaper proxies, many of which rely on the loss, its derivatives, or loss-surface geometry. Arpit et al. (2017) show that networks fit simple patterns before noise and introduce loss sensitivity as a proxy measure of memorization, while Agiollo et al. (2024) approxi-

mate memorization using discrepancies between models trained with sharpness-aware minimization (SAM) and SGD, motivated by loss-surface geometry.

Several recent memorization proxies use per-sample loss trajectories or input-loss derivatives. Garg et al. (2024) use the curvature of the loss with respect to the input as a memorization signal, averaged across training epochs. They show that high-curvature samples are often long-tailed, mislabeled, or conflicting. Cumulative Sample Loss (CSL; Ravikumar et al. (2025)) accumulates each sample's loss over training, motivated by the observation that harder examples maintain higher loss for longer. They also provide a theoretical connection to stability-based memorization. Ravikumar et al. (2026) characterize memorization using gradients of the loss with respect to individual inputs, accumulated over training as Cumulative Sample Gradient (CSG). Motivated by the observation that highly memorized samples tend to be learned later, they establish a theoretical link between CSG, learning time, and memorization. They also propose sample Gradient-Assisted early stopping with accumulated Loss (SGAL), which, like CSL (Ravikumar et al., 2025), accumulates per-sample loss but stops at a point determined by the sample-gradient dynamics; its theoretical justification follows from the CSL analysis. The evaluated proxies are compared in Section 4.1.

Detecting mislabeled examples. Mislabeled examples are inconsistent with the patterns of their assigned class, so fitting their assigned labels often requires memorization. Both Feldman & Zhang (2020) and Garg et al. (2024) report mislabeled or corrupted examples among high-scoring samples. Several methods target mislabeled examples using training dynamics, confidence, or margin-based statistics. Toneva et al. (2019) count forgetting events and find that frequently forgotten examples are hard, atypical, or noisy; Maini et al. (2022) extend this with second-split forgetting time (SSFT), which separates mislabeled from rare examples. Pleiss et al. (2020) aggregate over training the logit margin between the assigned label and the strongest competing class (AUM), Northcutt et al. (2021) estimate the joint distribution of noisy and true labels (Confident Learning), Carlini et al. (2019) study several measures of example representativeness, including model confidence and ensemble agreement, and Agarwal et al. (2022) measure the variance of gradients across checkpoints (VoG). Our mislabel indicator similarly exploits training-time information, but derives its signal from class-centroid geometry in representation space rather than from confidence, margin, or loss-derived quantities.

Representation geometry. Our score is inspired by Neural Collapse (Papyan et al., 2020), which describes a symmetric structure that emerges in the representations of deep classifiers during the terminal phase of training, after the training error has become negligible and optimization continues to reduce the loss. Papyan et al. (2020) characterize neural collapse through four interconnected phenomena (called NC1, NC2, NC3, and NC4). Most relevant to our work are NC1, under which within-class variability of last-layer features vanishes as individual representations concentrate around their corresponding class means, and NC4, under which the neural collapse classifier becomes functionally equivalent to a nearest-class-center classifier in the learned representation space. NC1 motivates tracking the sample-wise evolution toward class centroids, while NC4 provides a connection between class-centroid geometry and the network's classification behavior. Nguyen et al. (2023) demonstrate that mislabeled examples are more resistant to representational collapse.

3 METHODOLOGY

3.1 OUR SCORE

To set up the notation for our score, we consider a C-class classification problem and write $[C] = \{1, \dots, C\}$. The training data are N labeled examples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, $x_i \in \mathbb{R}^d$, $y_i \in [C]$, with inputs x_i and labels y_i. A deep classifier typically consists of a sequence of layers followed by a linear classifier. The layers preceding the classifier form a feature map $h_\theta : \mathbb{R}^d \to \mathbb{R}^p$ with parameters θ, sending an input x to a p-dimensional representation $h_\theta(x)$. The linear classifier, with weights $W \in \mathbb{R}^{C \times p}$ and bias $b \in \mathbb{R}^C$, reads this representation to produce the class scores, as $z_\theta(x) = W\, h_\theta(x) + b$, and the predicted label is $f_\theta(x) = \arg\max_{c \in [C]} \big(z_\theta(x)\big)_c$. The method works throughout with $h_\theta(x)$, the penultimate feature vector that the classifier reads.

Training runs for T epochs; let $\theta^{(t)}$ be the parameters at the end of epoch $t \in [T]$. The objects the method acts on are the per-epoch representations $h^{(t)}(x_i) \coloneqq h_{\theta^{(t)}}(x_i) \in \mathbb{R}^p$, $i \in [N]$, $t \in [T]$. For

each class $c \in [C]$ and epoch t, the class centroid is the empirical mean representation of the class,

$$\mu_c^{(t)} \coloneqq \frac{1}{|I_c|} \sum_{i \in I_c} h^{(t)}(x_i), \qquad I_c \coloneqq \{\, i \in [N] : y_i = c \,\}. \tag{2}$$

Centroids are recomputed at every epoch, so they evolve with the representation space during training. The Neural Collapse phenomenon NC4 establishes a connection between class-centroid geometry and classification in the terminal phase of training, where the classifier approaches a nearest-class-center rule (Papyan et al., 2020). Inspired by this connection, and without assuming that this rule holds before the terminal phase, we track how each example's representation relates to the class centroids throughout training. To remove sensitivity to representation magnitude across examples and epochs, we use cosine distance, which is invariant to feature scale. For sample x_i at epoch t, we define its cosine distance to the centroid of class c as:

$$D_c^{(t)}(x_i) \coloneqq 1 - \frac{\left\langle h^{(t)}(x_i), \mu_c^{(t)} \right\rangle}{\left\| h^{(t)}(x_i) \right\| \left\| \mu_c^{(t)} \right\|}. \tag{3}$$

We convert the distances to probabilities by applying the softmax function, yielding a continuous measure of how strongly the representation is associated with its observed class relative to competing classes. For sample (x_i, y_i) at epoch t, the probability assigned to the observed class y_i is

$$p_{\text{own}}^{(t)}(x_i) = \frac{\exp\left(-D_{y_i}^{(t)}(x_i)/\tau\right)}{\sum_c \exp\left(-D_c^{(t)}(x_i)/\tau\right)}, \tag{4}$$

where τ is the softmax temperature. Its complement,

$$s_i^{(t)} = 1 - p_{\text{own}}^{(t)}(x_i), \tag{5}$$

measures representation–label disagreement at epoch t. We interpret this disagreement as a representation-space notion of atypicality. Finally, we average it over the training trajectory to obtain a trajectory-level summary of this atypicality, defining our score, Representation Geometry Disagreement (RGD):

$$RGD_i = \frac{1}{T} \sum_{t=1}^{T} s_i^{(t)}. \tag{6}$$

3.2 Relation to the Feldman–Zhang score

We introduced the definition of memorization in Equation 1. In order to relate our metric to this score, we characterize how the representation geometry changes throughout training. We start by defining the epoch at which the representation becomes consistently aligned with its observed class, i.e., its disagreement stays below a threshold ε. We say that epoch T_i is the *representation ε-fitted epoch* if:

$$T_i \;:=\; \min\left\{\, t \in [T] : s_i^{(t')} \le \varepsilon \;\text{ for all } t' \ge t \,\right\}, \qquad T_i := T+1 \;\text{ if no such } t \text{ exists},$$

where tolerance $\varepsilon \in [0, 1)$ is fixed. The *unfit fraction*, the fraction of epochs that the sample has not been learned in the representation sense, is $\rho_i := (T_i - 1)/T \in [0, 1]$. T_i is not the loss-based learning epoch, which is when the *prediction* becomes correct, or when loss-based measurements fall below the threshold. The classifier head can fit a point before its representation moves, so T_i is typically later (Figure 1a, Section 4.1). To state the relation, we need some assumptions. The first assumes that if a model has a sample in its training data, the model will predict the label correctly. The second one considers a distribution for representations and connects our score to the distribution. The last assumption is a bridge between the representation space and model prediction. It concerns $\Pr_{f \leftarrow A(S \setminus i)}[f(x_i) = y_i]$, the probability that a model trained without x_i predicts its label, which reflects the prediction supported by the remaining training data.

Assumption 1 (Interpolation) $\Pr_{f \leftarrow A(S)}\big[f(x_i) = y_i\big] = 1$ *for all* $i \in [N]$.

Assumption 2 (Distribution of representations) *At each epoch* t*, the normalized representation of class* c *follows a von Mises-Fisher (vMF) distribution with mean direction* $\mu_c^{(t)}/\|\mu_c^{(t)}\|$ *and common concentration* κ_t*. With equal class priors, the class posterior given the representation is* $p_{\text{own}}^{(t)}(x_i)$*:*

$$\Pr\Big[Y = y_i \mid h^{(t)}(x_i)\Big] = \frac{\exp\Big(\kappa_t\big(1 - D_{y_i}^{(t)}(x_i)\big)\Big)}{\sum_c \exp\Big(\kappa_t\big(1 - D_c^{(t)}(x_i)\big)\Big)} = p_{\text{own}}^{(t)}(x_i)\Big|_{\tau = 1/\kappa_t}. \tag{7}$$

Remark 1 (Fixed temperature) *Estimating a concentration* κ_t *for every epoch is costly and noisy, so for simplicity we use a single fixed temperature. This does not change which centroid is nearest, only how sharply the softmax reports it, and it measures every epoch with the same criterion so that the values* $s_i^{(t)}$ *are comparable across* t*. A small temperature also makes* $s_i^{(t)}$ *close to* 0 *once* x_i *has joined its cluster, which is what makes* ε *small in Remark 2.*

Assumption 3 (Leave-one-out proxy) *During training, before* x_i *becomes representation-fitted, we assume that training on* x_i *has not yet induced a strong example-specific association in the model, which makes* $h^{(t)}(x_i)$ *primarily a reflection of what the rest of the data implies about* x_i*. Thus, we use* $\Pr\big[Y = y_i \mid h^{(t)}(x_i)\big]$ *as a proxy for the leave-one-out prediction probability, with per-example error* δ_i*: for each example* i*, there exists* $\delta_i \geq 0$ *such that, for every unfit epoch* $t < T_i$*,*

$$\Big|\Pr_{f \leftarrow A(S \setminus i)}\big[f(x_i) = y_i\big] - \Pr\big[Y = y_i \mid h^{(t)}(x_i)\big]\Big| \;\leq\; \delta_i.$$

For a fixed model, the prediction is obtained from the representation through the classifier head; the representation carries the information the model uses to predict x_i's label. Thus, we can reason about the model's label prediction through the representation. Based on these assumptions, we now show the relation between our score and the Feldman–Zhang memorization score in the following proposition.

Proposition 1 (Sandwich bound) *Under Assumptions 1–3,*

$$\rho_i\big(\,\mathrm{mem}(A, S, i) - \delta_i\big) \;\leq\; RGD_i \;\leq\; \rho_i\big(\,\mathrm{mem}(A, S, i) + \delta_i\big) + (1 - \rho_i)\,\varepsilon.$$

Proof. Write $\mathrm{mem} := \mathrm{mem}(A, S, i)$. By Assumption 1, the first term of $\mathrm{mem}(A, S, i)$ equals one, so $\mathrm{mem} = 1 - \Pr_{f \leftarrow A(S \setminus i)}[f(x_i) = y_i]$. At *unfit epochs* $t < T_i$, by Assumption 2, $s_i^{(t)} = 1 - \Pr[Y = y_i \mid h^{(t)}(x_i)]$. By subtracting mem from $s_i^{(t)}$, and using Assumption 3:

$$\big|s_i^{(t)} - \mathrm{mem}\,\big| = \Big|\Pr_{f \leftarrow A(S \setminus i)}\big[f(x_i) = y_i\big] - \Pr\big[Y = y_i \mid h^{(t)}(x_i)\big]\Big| \leq \delta_i$$

Hence,

$$\mathrm{mem} - \delta_i \;\leq\; s_i^{(t)} \;\leq\; \mathrm{mem} + \delta_i. \tag{8}$$

At *fitted epochs* $t \geq T_i$, by the definition of T_i, $0 \leq s_i^{(t)} \leq \varepsilon$.

Split the average at T_i*.*

$$RGD_i = \frac{1}{T}\sum_{t=1}^{T} s_i^{(t)} = \underbrace{\frac{1}{T}\sum_{t < T_i} s_i^{(t)}}_{T_i - 1 = \rho_i T \text{ terms}} + \underbrace{\frac{1}{T}\sum_{t \geq T_i} s_i^{(t)}}_{\in\,[0,\,(1-\rho_i)\varepsilon]}.$$

Applying inequality 8 to the first sum gives $\rho_i(\mathrm{mem} - \delta_i) \leq \frac{1}{T}\sum_{t<T_i} s_i^{(t)} \leq \rho_i(\mathrm{mem} + \delta_i)$. The second sum lies in $[0, (1 - \rho_i)\varepsilon]$. Adding the two lower bounds and the two upper bounds gives the claim. □

Remark 2 *The bound has width $2\rho_i\delta_i + (1-\rho_i)\varepsilon$. With $\tau = 0.02$ the softmax is sharp, so ε is small. For $\rho_i\delta_i$, there are two regimes. If x_i is fit fast, ρ_i is small, so $\rho_i\delta_i$ is small even if δ_i is large, and $RGD_i \approx 0$, which is the correct answer since such an example is implied by the rest of the data. If x_i is fit slowly, the network has by then been trained on all other examples, so $\Pr\left[Y = y_i \mid h^{(t)}(x_i)\right]$ is a more plausible leave-one-out proxy and motivates a smaller δ_i. Thus, $\rho_i\delta_i$ is small. Hence:*

$$RGD_i \;\approx\; \rho_i \cdot \mathrm{mem}(A, S, i).$$

In this regime, the bound becomes tight, and our proposed score approximates the true Feldman–Zhang memorization score weighted by the fraction of training during which the example was not yet fit. The unfit fraction ρ_i is itself related to memorization: examples whose label follows from the rest of the data are fit early, while atypical and mislabeled examples are fit late. Both factors in $\rho_i \cdot \mathrm{mem}(A, S, i)$ therefore tend to increase with memorization. We verify the agreement between RGD_i and the Feldman–Zhang score in Section 4.1.

These results establish a connection between the representation geometry of data point i (relative to the C class centroids) and the memorization of said data point, under assumptions consistent with the Neural Collapse phenomenon (Papyan et al., 2020). Next we show the practical upside of working with the representation geometry as a proxy for memorization.

4 Experiments

We evaluate RGD in three settings: memorization evaluation, with a particular focus on agreement with the FZ memorization score (Feldman & Zhang, 2020), mislabel detection, and label recovery. The experimental setup is provided in Appendix A. We also study whether the score can be computed from shorter training windows to reduce computational cost. The results show that shorter trajectory windows retain most of the memorization signal; the corresponding experiments are provided in Appendix B. In addition to our principled metric, we evaluate several alternative class-relative quantities derived from the same representation space. These quantities also contain memorization-relevant information, suggesting that the memorization signal is broadly present in the representation space rather than being specific to our metric. These results are reported in Appendix C. We use a fixed temperature of $\tau = 0.02$ in all experiments. A temperature ablation study is provided in Appendix D, showing that agreement with the FZ score remains stable over $\tau \in [0.01, 0.03]$ on both ImageNet and CIFAR-100.

4.1 Memorization evaluation

To assess agreement with the FZ memorization score (Feldman & Zhang, 2020), we compute Cosine Similarity (CS) and Pearson correlation between the sample-wise score vectors of RGD and FZ. The experimental setup is provided in Appendix A.1. Table 1 reports results on the ImageNet and CIFAR-100 datasets. For ImageNet, because our experimental setup matches that of Ravikumar et al. (2026), we use their reported baseline values. All CIFAR-100 baselines are from our own runs under a common protocol (Appendix A.1). CIFAR-100 results and our method on ImageNet are reported as mean and standard deviation (std) over three seeds. On ImageNet, our representation-based score achieves the highest agreement with the Feldman–Zhang scores (Feldman & Zhang, 2020) among the evaluated proxies, reaching a mean cosine similarity of 0.8394 and a mean Pearson correlation of 0.7361, compared with 0.79 and 0.64 for CSL, the strongest baseline. On CIFAR-100, RGD remains competitive, achieving a mean of 0.8577 on cosine similarity and 0.7584 on Pearson correlation.

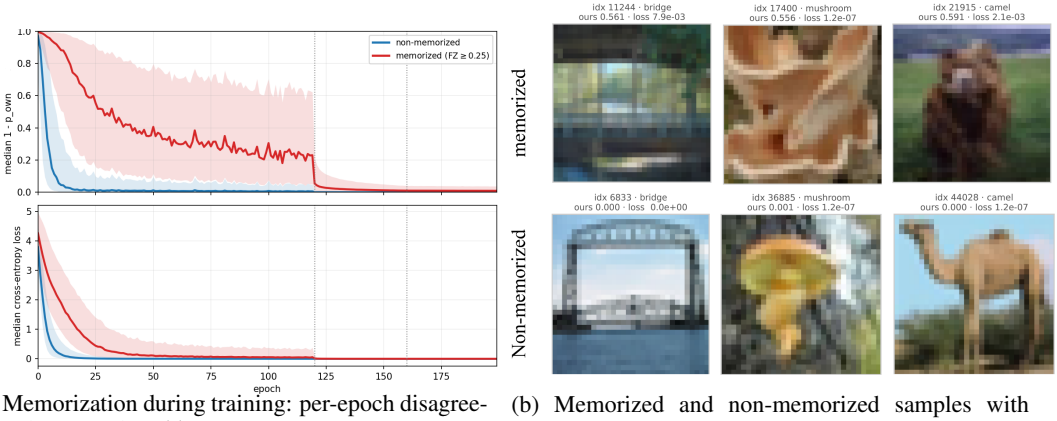
An advantage of our representation-based score is that its memorization signal persists up to the terminal phase of training. Our score is based on relative distances to class centroids. Thus, our signal remains available late into training, until within-class representations collapse toward their class means, as described by NC1 in Papyan et al. (2020). In contrast, the instantaneous loss naturally decreases as a sample becomes fitted. Many existing memorization proxies aggregate information derived from these per-epoch loss signals, whereas our representation-space score can remain discriminative even after the instantaneous loss has become very small.

Figure 1a illustrates this behavior by the per-epoch median and InterQuartile Range (IQR) of the loss and our score for memorized and non-memorized examples. Following Feldman & Zhang

(2020), we consider a sample memorized when its FZ score is greater than 0.25. The representation-space score remains clearly separated between memorized and non-memorized examples until approximately epoch 120, where the learning rate drops for the first time, whereas the instantaneous loss of memorized examples has largely collapsed before epoch 50. Figure 1b shows examples at epoch 75: memorized samples already have near-zero loss while retaining high representation-space scores. The images also qualitatively illustrate the atypicality of memorized examples relative to non-memorized examples with the same labels.

Table 1: Memorization alignment with Feldman–Zhang scores (Feldman & Zhang, 2020) on ImageNet and CIFAR-100. For CIFAR-100, and for RGD on ImageNet, we report the mean and std of metrics over three seeds.

	ImageNet		CIFAR-100	
Method	CS ↑	Pearson ↑	CS ↑	Pearson ↑
Final Sample Loss	0.63	0.49	0.1230 ± 0.0064	0.0843 ± 0.0036
Loss Sensitivity (Arpit et al., 2017)	0.49	0.17	0.8331 ± 0.0016	0.7754 ± 0.0034
Forget Freq. (Toneva et al., 2019)	0.49	0.04	0.8329 ± 0.0005	0.7293 ± 0.0010
Curvature (Garg et al., 2024)	0.62	0.33	0.8136 ± 0.0093	0.6776 ± 0.0165
CSL (Ravikumar et al., 2025)	0.79	0.64	0.8872 ± 0.0005	0.8188 ± 0.0009
CSG (Ravikumar et al., 2026)	0.71	0.52	0.8341 ± 0.0005	0.7148 ± 0.0009
SGAL (Ravikumar et al., 2026)	0.78	0.62	$\mathbf{0.8878} \pm \mathbf{0.0023}$	$\mathbf{0.8217} \pm \mathbf{0.0037}$
RGD (OURS)	$\mathbf{0.8394} \pm \mathbf{0.0010}$	$\mathbf{0.7361} \pm \mathbf{0.0011}$	0.8577 ± 0.0010	0.7584 ± 0.0018



(a) Memorization during training: per-epoch disagreement $(1 - p_{\text{own}})$ and loss

(b) Memorized and non-memorized samples with near-zero loss at epoch 75

Figure 1: Memorization case on CIFAR-100.

4.2 MISLABEL DETECTION

Here, the goal is to assess whether the score can distinguish mislabeled examples from correctly labeled ones. Because mislabeled samples are inconsistent with the patterns of their assigned class, fitting their labels often requires the model to memorize these examples. Mislabel detection therefore provides a controlled and useful setting for evaluating a memorization proxy.

We consider two label-corruption settings: symmetric and asymmetric. Under symmetric corruption, a specified fraction of training labels is replaced uniformly at random with one of the remaining classes. Under asymmetric corruption, the replacement class is restricted to another class within the same superclass as the original label, producing a more structured and semantically plausible corruption pattern than uniform random label flips. To the best of our knowledge, this is the first evaluation of per-example memorization proxies for mislabel detection under asymmetric label noise. We evaluate several corruption rates (noise ratios). The experimental setup is provided in Appendix A.2.

We report the mean and standard deviation of the AUROC over three seeds. Table 2 reports the results for asymmetric corruption on CIFAR-100; we do not evaluate this setting on CIFAR-10

because it does not provide superclass labels. Tables 3 and 4 report the results for symmetric corruption on CIFAR-10 and CIFAR-100, respectively. For the symmetric case, baseline values are from Ravikumar et al. (2026) except CL (Northcutt et al., 2021) and AUM (Pleiss et al., 2020), which we evaluate separately.

Under asymmetric CIFAR-100 corruption, RGD achieves the highest AUROC among the evaluated methods from 10% noise onward and remains stable, decreasing only from 0.9353 at 5% noise to 0.9281 at 30%, compared with drops of up to approximately twelve percentage points for competing methods. Under symmetric CIFAR-10 corruption, it is also among the strongest methods, with AUROC changing only from 0.9957 to 0.9920 over the same noise range. On symmetric CIFAR-100, the score remains stable, although below the strongest baselines. Overall, these results indicate that representation–label disagreement remains robust as label corruption increases.

The representation-space signal also persists in mislabel detection. Under 20% asymmetric label noise, Figure 2 shows that mislabeled examples remain separated after their instantaneous losses collapse, including at epoch 75 (Figure 2b). The nearest non-assigned class often matches the original true label, motivating Section 4.3. The trajectories also reveal a distinction between correctly labeled memorized and mislabeled examples: disagreement for memorized examples eventually approaches zero, whereas mislabeled examples retain nonzero disagreement late in training. This is consistent with correctly labeled memorized examples being atypical within their true class, while mislabeled examples additionally mismatch their semantic content and assigned label.

Table 2: AUROC of mislabeled-sample detection under *asymmetric* (within-superclass) label noise on CIFAR-100, for various levels of noise (% of training data with noisy labels).

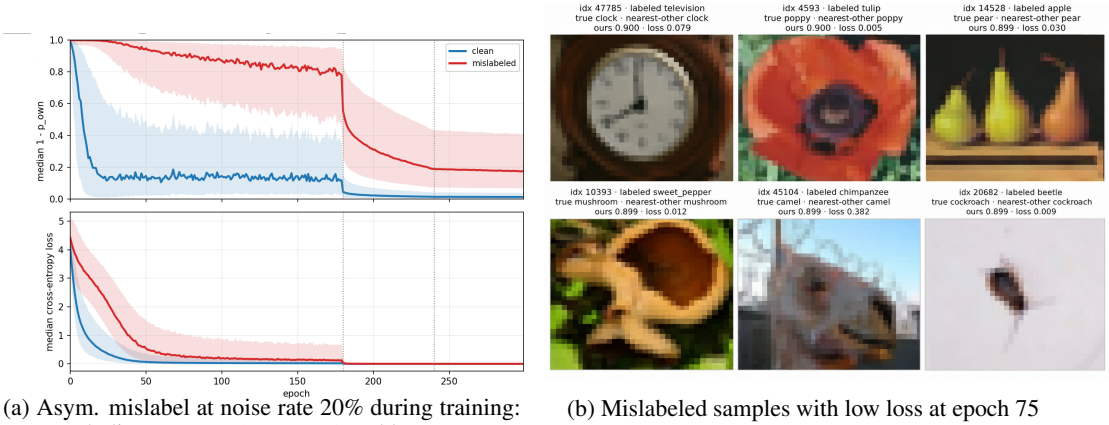
Method	5%	10%	20%	25%	30%
In Conf. (Carlini et al., 2019)	0.7418 ± 0.0025	0.7056 ± 0.0035	0.6609 ± 0.0049	0.6426 ± 0.0010	0.6189 ± 0.0028
AUM (Pleiss et al., 2020)	$\mathbf{0.9364 \pm 0.0024}$	0.9251 ± 0.0038	0.8901 ± 0.0012	0.8678 ± 0.0013	0.8459 ± 0.0017
CL (Northcutt et al., 2021)	0.8449 ± 0.0067	0.8387 ± 0.0009	0.8226 ± 0.0029	0.8107 ± 0.0024	0.7963 ± 0.0029
Curvature (Garg et al., 2024)	0.8264 ± 0.0053	0.7996 ± 0.0031	0.7563 ± 0.0020	0.7320 ± 0.0008	0.7159 ± 0.0025
CSL (Ravikumar et al., 2025)	0.9300 ± 0.0043	0.9151 ± 0.0009	0.8880 ± 0.0031	0.8696 ± 0.0029	0.8533 ± 0.0007
CSG (Ravikumar et al., 2026)	0.8392 ± 0.0019	0.8078 ± 0.0031	0.7650 ± 0.0054	0.7415 ± 0.0045	0.7192 ± 0.0053
SGAL (Ravikumar et al., 2026)	0.9360 ± 0.0049	0.9241 ± 0.0006	0.9007 ± 0.0018	0.8859 ± 0.0036	0.8693 ± 0.0017
RGD (OURS)	0.9353 ± 0.0037	$\mathbf{0.9347 \pm 0.0019}$	$\mathbf{0.9337 \pm 0.0004}$	$\mathbf{0.9314 \pm 0.0009}$	$\mathbf{0.9281 \pm 0.0015}$

Table 3: AUROC of mislabeled-sample detection under *symmetric* label noise on CIFAR-10.

Method	5%	10%	20%	25%	30%
In Conf. (Carlini et al., 2019)	0.7254 ± 0.0214	0.6528 ± 0.0042	0.5978 ± 0.0131	0.5800 ± 0.0051	0.5669 ± 0.0106
AUM (Pleiss et al., 2020)	0.9914 ± 0.0009	0.9816 ± 0.0024	0.9578 ± 0.0038	0.9455 ± 0.0002	0.9269 ± 0.0059
CL (Northcutt et al., 2021)	0.9482 ± 0.0023	0.9422 ± 0.0011	0.9261 ± 0.0018	0.9189 ± 0.0013	0.9092 ± 0.0009
Curvature (Garg et al., 2024)	0.9800 ± 0.0003	0.9819 ± 0.0006	0.9934 ± 0.0002	0.9932 ± 0.0001	0.9932 ± 0.0006
CSL (Ravikumar et al., 2025)	0.9845 ± 0.0026	0.9864 ± 0.0004	0.9870 ± 0.0003	0.9904 ± 0.0005	0.9906 ± 0.0003
CSG (Ravikumar et al., 2026)	0.9783 ± 0.0009	0.9809 ± 0.0011	0.9936 ± 0.0002	$\mathbf{0.9934 \pm 0.0001}$	$\mathbf{0.9935 \pm 0.0006}$
SGAL (Ravikumar et al., 2026)	0.9115 ± 0.0057	0.8835 ± 0.0128	0.8664 ± 0.0070	0.8553 ± 0.0145	0.8470 ± 0.0283
RGD (OURS)	$\mathbf{0.9957 \pm 0.0001}$	$\mathbf{0.9952 \pm 0.0003}$	$\mathbf{0.9943 \pm 0.0003}$	0.9930 ± 0.0005	0.9920 ± 0.0007

Table 4: AUROC of mislabeled-sample detection under *symmetric* label noise on CIFAR-100.

Method	5%	10%	20%	25%	30%
In Conf. (Carlini et al., 2019)	0.7069 ± 0.0069	0.6884 ± 0.0053	0.6493 ± 0.0075	0.6324 ± 0.0051	0.6257 ± 0.0044
AUM (Pleiss et al., 2020)	0.9478 ± 0.0004	0.9409 ± 0.0012	0.9190 ± 0.0026	0.9113 ± 0.0015	0.9007 ± 0.0014
CL (Northcutt et al., 2021)	0.8533 ± 0.0021	0.8457 ± 0.0063	0.8303 ± 0.0023	0.8237 ± 0.0031	0.8116 ± 0.0023
Curvature (Garg et al., 2024)	0.9876 ± 0.0021	0.9892 ± 0.0012	0.9931 ± 0.0004	0.9931 ± 0.0004	0.9932 ± 0.0002
CSL (Ravikumar et al., 2025)	0.9891 ± 0.0003	0.9895 ± 0.0002	0.9902 ± 0.0002	0.9904 ± 0.0002	0.9903 ± 0.0002
CSG (Ravikumar et al., 2026)	$\mathbf{0.9896 \pm 0.0008}$	$\mathbf{0.9904 \pm 0.0006}$	$\mathbf{0.9934 \pm 0.0003}$	$\mathbf{0.9936 \pm 0.0002}$	$\mathbf{0.9936 \pm 0.0001}$
SGAL (Ravikumar et al., 2026)	0.9895 ± 0.0006	0.9897 ± 0.0008	0.9856 ± 0.0008	0.9863 ± 0.0009	0.9861 ± 0.0004
RGD (OURS)	0.9715 ± 0.0014	0.9740 ± 0.0008	0.9733 ± 0.0010	0.9722 ± 0.0008	0.9704 ± 0.0001



(a) Asym. mislabel at noise rate 20% during training: per-epoch disagreement ($1 - p_{\text{own}}$) and loss

(b) Mislabeled samples with low loss at epoch 75

Figure 2: Asymmetric mislabel case on CIFAR-100.

4.3 Recovering the true label of mislabeled examples

Having shown that RGD is effective at identifying *whether* a data point is mislabeled, we now go further to examine whether the *true label* of mislabeled samples can also be recovered from representation-space geometry. This is possible because each example can be compared with every class centroid, rather than being summarized only by a single memorization score. Therefore, for a mislabeled example, we identify the non-assigned class with the highest average similarity (lowest distance) over the training trajectory and use it as an estimate of the true label.

To the best of our knowledge, existing explicit per-example memorization scores are scalar and do not directly provide information about plausible alternative labels. For comparison, we use two confidence-based baselines previously used for mislabel detection, In-Confidence (Carlini et al., 2019) and Confident Learning (Northcutt et al., 2021). For both, we use the most likely non-assigned class as the recovered label. Table 5 reports label-recovery accuracy in three settings at 20% noise, with experimental details in Appendix A.3. Our representation-geometry recovery rule outperforms both baselines and the corresponding chance level in all three settings.

Table 5: Mean and std of true-label recovery accuracy at 20% label noise over three seeds.

Case	Chance	In Conf.	CL	Ours (Rep. Geometry)
Sym. CIFAR-10	0.111	0.6098 ± 0.0083	0.8268 ± 0.0057	$\mathbf{0.8927 \pm 0.0005}$
Sym. CIFAR-100	0.010	0.3221 ± 0.0040	0.5571 ± 0.0024	$\mathbf{0.6101 \pm 0.0076}$
Asym. CIFAR-100	0.250	0.5155 ± 0.0034	0.6020 ± 0.0041	$\mathbf{0.7820 \pm 0.0044}$

5 Conclusion

We investigated whether memorization can be identified directly from the geometry of learned representations and proposed a single-training-run proxy based on trajectory-averaged representation–label disagreement relative to class centroids. Under explicit assumptions, we relate this score to Feldman–Zhang memorization through a sandwich bound. The resulting score closely aligns with the Feldman–Zhang memorization score on ImageNet and CIFAR-100, performs strongly for mislabeled-sample detection, and supports true-label recovery through class-resolved geometry. We further show that the representation-space signal can remain informative after instantaneous loss has largely collapsed. Taken together, our results indicate that memorization leaves a measurable geometric signature in learned representations. Our study is limited to image classification with ResNet architectures; extending this analysis across architectures, modalities, and training regimes, and establishing a deeper theoretical connection to the counterfactual definition of memorization, are promising directions for future work.

Reproducibility Statement

All experimental settings needed to reproduce our results are stated in Appendix A: architectures (ResNet-18 for CIFAR-10/CIFAR-100, ResNet-50 for ImageNet), epoch budgets, learning-rate schedules, weight decay, and the softmax temperature τ. The label-noise protocol, including the noise rates and the symmetric and asymmetric relabeling procedures, is described in the mislabel-detection subsection. All reported AUROC values are means and sample standard deviations over three seeds. The proxy itself requires only penultimate-layer features and per-class centroids from a standard training run and involves no additional hyperparameters beyond τ.

The Use of Large Language Models (LLMs)

In this work, we used generative AI tools for implementing methods, providing feedback on research methodology or experiments, assisting in the writing of proofs, and interpreting results. We did not use generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, or assist with translation. Generating synthetic datasets, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis were not applicable to this work.

Additionally, we used generative AI tools for suggesting experimental parameters, creating and editing code, summarizing or analyzing existing literature, discovering research topics, sourcing and searching for information, editing the manuscript to improve readability, identifying relevant literature, formatting references, and drafting parts of the manuscript.

All AI-assisted mathematical arguments and proofs were reviewed and verified by the authors. AI-assisted code and method implementations were checked against the intended experimental methodology, and AI-assisted literature searches, summaries, citations, and references were verified against the original sources. We reviewed all AI-assisted manuscript content and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10368–10378, 2022.

Andrea Agiollo, Young In Kim, and Rajiv Khanna. Approximating memorization using loss surface geometry for dataset pruning and summarization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 17–28. ACM, 2024. doi: 10.1145/3637528.3671985.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, pp. 233–242, 2017.

Nicholas Carlini, Úlfar Erlingsson, and Nicolas Papernot. Distribution density, tails, and outliers in machine learning: Metrics and applications. *arXiv preprint arXiv:1910.13427*, 2019.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, 2022a. doi: 10.1109/SP46214.2022.9833649.

Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramèr. The privacy onion effect: Memorization is relative. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 13263–13276, 2022b.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 954–959, 2020.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 2881–2891, 2020.

Isha Garg, Deepak Ravikumar, and Kaushik Roy. Memorization through the lens of curvature of loss function around samples. In *International Conference on Machine Learning (ICML)*, 2024.

Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C. Mozer. Characterizing structural regularities of labeled data in overparameterized models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 5034–5044, 2021.

Young In Kim, Andrea Agiollo, Pratiksha Agrawal, Johannes O. Royset, and Rajiv Khanna. Membership privacy risks of sharpness aware minimization. In *International Conference on Learning Representations (ICLR)*, 2026.

Pratyush Maini, Saurabh Garg, Zachary Lipton, and J. Zico Kolter. Characterizing datapoints via second-split forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Duc Anh Nguyen, Ron Levie, Julian Lienen, Eyke Hüllermeier, and Gitta Kutyniok. Memorization-dilation: Modeling neural collapse under label noise. In *International Conference on Learning Representations (ICLR)*, 2023.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 19920–19930, 2020.

Deepak Ravikumar, Efstathia Soufleri, Abolfazl Hashemi, and Kaushik Roy. Towards memorization estimation: Fast, formal and free. In *International Conference on Machine Learning (ICML)*, 2025.

Deepak Ravikumar, Efstathia Soufleri, Abolfazl Hashemi, and Kaushik Roy. Memorization through the lens of sample gradients. In *International Conference on Learning Representations (ICLR)*, 2026.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017. doi: 10.1109/SP.2017.41.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations (ICLR)*, 2019.

APPENDIX

A EXPERIMENT SETUP

All models were trained with weight decay 1e-4 and SGD momentum 0.9. The learning rate (LR) is 0.1 at the beginning and is decayed by 0.1 at two milestones. Also, the softmax temperature (τ) is 0.02 in all experiments.

Data preprocessing and trajectory extraction. For CIFAR-10/100, training uses random 32×32 crops with 4-pixel padding and random horizontal flips, followed by tensor conversion and per-channel normalization using the corresponding dataset mean and standard deviation; training uses mixed precision. For ImageNet, training uses $\mathrm{RandomResizedCrop}(224)$ and random horizontal flips, followed by tensor conversion and normalization with the standard ImageNet statistics; training also uses mixed precision.

For trajectory extraction, no data augmentation is used. For CIFAR-10/100, images undergo only tensor conversion and normalization, and representations are collected at the end of each epoch in a full-precision evaluation-mode pass with a fixed dataset order. For ImageNet, images are resized to 256 pixels and center-cropped to 224×224; the cropped uint8 images are cached and reused across checkpoints so that the scoring inputs are byte-identical across the trajectory. Representations are then extracted in evaluation mode from the saved training checkpoints.

We use the penultimate representation after global average pooling, which is 512-dimensional for ResNet-18 and 2048-dimensional for ResNet-50. Class centroids (Eq. 2) and the representation-space scores (Eqs. 3–6) are computed exclusively from these representations. In the label-noise experiments, the assigned (possibly corrupted) label is used as the observed label y_i both in p_{own} and in constructing the class centroids; the original clean labels are used only for evaluation. Trajectory extraction requires one additional evaluation-mode pass over the training set for each recorded checkpoint, but no additional model training.

A.1 Setup of memorization experiments

In the memorization task, we used the CIFAR-100 and ImageNet datasets. For CIFAR-100, a ResNet-18 model is trained for 200 epochs. LR decreases at epochs 120 and 160, and the batch size is 128. For the ImageNet dataset, ResNet-50 is trained for 90 epochs. LR decreases at epochs 29 and 59, and the batch size is 256. Loss sensitivity (Arpit et al., 2017) is computed as the standard deviation of per-example loss across training epochs, following the implementation of Ravikumar et al. (2026); the original unrolled-gradient definition is infeasible at this scale.

A.2 Setup of mislabeled experiments

In the mislabeled task, we trained ResNet-18 for all the datasets. The total number of epochs is 300, and the LR decreases at epochs 180 and 240. The batch size is 128. For CL (Northcutt et al., 2021), we use cleanlab[1]'s confident-joint filter on 3-fold out-of-sample predicted probabilities. Flagged examples are ranked by normalized margin, while unflagged examples are tied at zero. We use this run rather than the value reported by Ravikumar et al. (2026) so that the same held-out probabilities serve both mislabel detection and label recovery (Section 4.3).

A.3 Setup of true label recovery experiments

All settings for our method are the same as the mislabel settings (Appendix A.2). Regarding In-Confidence (Carlini et al., 2019) and Confident Learning (Northcutt et al., 2021), both are confidence-based, but they read the model's probabilities from different places. In-Confidence (Carlini et al., 2019) uses the model's in-sample probabilities from the model that already memorized the training labels, while Confident Learning (Northcutt et al., 2021) uses out-of-sample (held-out) probabilities. Held-out probabilities point at the true class much more reliably, so Confident Learning (Northcutt et al., 2021) recovers the correct label more often. The recovery rule is identical; only the probability source differs, which is why the two numbers differ.

B Shorter trajectory windows retain most of the signal

Our final score is obtained by averaging the per-epoch scores, $s_i^{(t)}$, over the training trajectory. To reduce cost, we can instead average over a shorter window starting from the first epoch, with the endpoint defined as a fraction of the total training epochs. Table 6 reports the CIFAR-100 results

[1] https://github.com/cleanlab/cleanlab

Table 6: Memorization on CIFAR-100 across trajectory windows.

Window size	Cosine similarity ↑	Pearson Correlation ↑
0.1	0.831	0.732
0.2	0.859	0.770
0.3	0.865	0.777
0.4	0.868	0.779
0.5	0.869	0.780
0.6	0.869	0.781
0.7	0.869	0.777
0.8	0.865	0.771
0.9	0.862	0.766
1.0	0.860	0.760

Table 7: ImageNet memorization across trajectory windows.

Window	Cosine Similarity	Pearson Correlation
$w = 0.1$	0.730	0.535
$w = 0.2$	0.756	0.586
$w = 0.3$	0.768	0.608
$w = 0.4$	0.788	0.651
$w = 0.5$	0.803	0.678
$w = 0.6$	0.813	0.695
$w = 0.7$	0.822	0.710
$w = 0.8$	0.830	0.723
$w = 0.9$	0.835	0.731
$w = 1.0$	0.840	0.736

in terms of agreement with the FZ score (Feldman & Zhang, 2020). Table 7 reports the ImageNet memorization results, and Tables 8, 9, and 10 report the mislabel detection results.

On CIFAR-100, performance improves rapidly over the early trajectory, plateaus over intermediate windows, and remains comparable when the full trajectory is used. ImageNet benefits more consistently from longer windows, although even $w = 0.1$ retains substantial alignment with the FZ score with cosine similarity 0.73, above most of the baselines. These results indicate that trajectory aggregation is useful while also allowing the computation to be truncated when budget constraints require it. SGAL (Ravikumar et al., 2026) has a similar early-stopping feature due to the early signal. In our case, our score can also be truncated, but for a different reason: representation–label disagreement is present from early training and persists, so early windows already contain most of it, while the late post-collapse epochs add little (Section 4.1). RGD requires a single training run and no retraining; if training is terminated at window w, only the first w fraction of the full training trajectory is required, together with one forward-only pass per retained checkpoint.

C The memorization signal belongs to the representation space, not to our statistic

Section 3.2 relates our score to the FZ score. A natural question is whether its agreement with memorization is specific to this statistic or whether other quantities derived from the same representation space also contain memorization-relevant information. We therefore compare our score with several alternative geometric quantities.

Specifically, we consider d_{own}, the cosine distance to the observed-class centroid; $d_{\text{nearest}-\text{other}}$, the cosine distance to the nearest alternative-class centroid; the ratio $d_{\text{own}}/d_{\text{nearest}-\text{other}}$; and the entropy of the centroid-induced class-probability distribution. The class-probability distribution is obtained analogously to Eq. 4, but over all classes. As with our main score, each quantity is aggregated over the training trajectory. These alternatives capture different aspects of the representation geometry. The motivation is again related to atypicality. Memorized examples may exhibit a larger

Table 8: Window analysis of the mislabeled symmetric task on CIFAR-100. Values are AUROC (mean ± std over seeds 0/1/2).

Window	5%	10%	20%	25%	30%
$w = 0.1$	0.9807 ± 0.0015	0.9817 ± 0.0006	0.9807 ± 0.0009	0.9799 ± 0.0008	0.9785 ± 0.0004
$w = 0.2$	0.9776 ± 0.0014	0.9796 ± 0.0007	0.9793 ± 0.0009	0.9791 ± 0.0009	0.9779 ± 0.0004
$w = 0.3$	0.9768 ± 0.0013	0.9790 ± 0.0007	0.9789 ± 0.0009	0.9787 ± 0.0008	0.9774 ± 0.0002
$w = 0.4$	0.9766 ± 0.0012	0.9789 ± 0.0008	0.9787 ± 0.0009	0.9784 ± 0.0008	0.9771 ± 0.0002
$w = 0.5$	0.9765 ± 0.0013	0.9788 ± 0.0007	0.9786 ± 0.0010	0.9783 ± 0.0007	0.9769 ± 0.0003
$w = 0.6$	0.9765 ± 0.0013	0.9789 ± 0.0007	0.9785 ± 0.0009	0.9782 ± 0.0007	0.9767 ± 0.0002
$w = 0.7$	0.9748 ± 0.0014	0.9774 ± 0.0008	0.9772 ± 0.0010	0.9766 ± 0.0008	0.9751 ± 0.0001
$w = 0.8$	0.9735 ± 0.0014	0.9761 ± 0.0008	0.9757 ± 0.0010	0.9749 ± 0.0008	0.9734 ± 0.0001
$w = 0.9$	0.9724 ± 0.0014	0.9750 ± 0.0008	0.9744 ± 0.0010	0.9735 ± 0.0008	0.9718 ± 0.0001
$w = 1.0$	0.9715 ± 0.0014	0.9740 ± 0.0008	0.9733 ± 0.0010	0.9722 ± 0.0008	0.9704 ± 0.0001

Table 9: Window analysis of the mislabeled symmetric task on CIFAR-10. Values are AUROC (mean ± std over seeds 0/1/2).

Window	5%	10%	20%	25%	30%
$w = 0.1$	0.9935 ± 0.0004	0.9931 ± 0.0005	0.9921 ± 0.0006	0.9891 ± 0.0008	0.9875 ± 0.0002
$w = 0.2$	0.9952 ± 0.0002	0.9950 ± 0.0003	0.9945 ± 0.0003	0.9933 ± 0.0004	0.9925 ± 0.0006
$w = 0.3$	0.9955 ± 0.0001	0.9952 ± 0.0002	0.9946 ± 0.0003	0.9935 ± 0.0005	0.9928 ± 0.0006
$w = 0.4$	0.9956 ± 0.0002	0.9953 ± 0.0003	0.9946 ± 0.0003	0.9934 ± 0.0005	0.9926 ± 0.0006
$w = 0.5$	0.9957 ± 0.0001	0.9953 ± 0.0003	0.9945 ± 0.0003	0.9933 ± 0.0005	0.9925 ± 0.0006
$w = 0.6$	0.9958 ± 0.0001	0.9953 ± 0.0003	0.9944 ± 0.0003	0.9932 ± 0.0005	0.9923 ± 0.0006
$w = 0.7$	0.9958 ± 0.0001	0.9952 ± 0.0003	0.9944 ± 0.0003	0.9931 ± 0.0005	0.9922 ± 0.0007
$w = 0.8$	0.9958 ± 0.0001	0.9952 ± 0.0003	0.9943 ± 0.0003	0.9931 ± 0.0005	0.9921 ± 0.0007
$w = 0.9$	0.9958 ± 0.0001	0.9952 ± 0.0003	0.9943 ± 0.0003	0.9931 ± 0.0005	0.9920 ± 0.0007
$w = 1.0$	0.9957 ± 0.0001	0.9952 ± 0.0003	0.9943 ± 0.0003	0.9930 ± 0.0005	0.9920 ± 0.0007

distance to the assigned-class centroid, greater proximity to another class, or greater uncertainty in the centroid-induced class distribution.

Tables 11 and 12 compare these quantities with the Feldman–Zhang memorization score on CIFAR-100 and ImageNet. Four of the five considered geometric quantities show meaningful agreement. The presence of substantial memorization signal across several distinct quantities shows that the result is not specific to our particular statistic. Rather, memorization-relevant information is broadly present in the learned representation geometry. We observe the same general pattern in the mislabel-detection setting. Results are in Tables 13, 14, and 15. In mislabel detection, the ratio remains close to our main score in all settings and d_{own} in the symmetric settings; entropy is weaker than in the memorization setting but still informative, and $d_{\text{nearest-other}}$ is weak. Based on Tables 13 and 15, the centroid-distance ratio slightly outperforms our score in some mislabel settings, and we keep a single principled statistic across both tasks rather than selecting per setting. So, the representation space is a rich space for detecting mislabels and memorization.

D Sensitivity to the softmax temperature

All experiments use a fixed temperature $\tau = 0.02$ (Remark 1). Tables 16 and 17 report the agreement of RGD_i with the Feldman–Zhang score as τ varies, with all other settings unchanged and the full trajectory ($w = 1.0$). Agreement is roughly stable over $\tau \in [0.01, 0.03]$ on both datasets and degrades outside that range; at $\tau = 1.0$ it falls to about 0.6 cosine similarity on both datasets.

Table 10: Window analysis of the mislabeled asymmetric task on CIFAR-100. Values are AUROC (mean ± std over seeds 0/1/2).

Window	5%	10%	20%	25%	30%
$w = 0.1$	0.9355 ± 0.0033	0.9338 ± 0.0030	0.9309 ± 0.0009	0.9279 ± 0.0015	0.9244 ± 0.0015
$w = 0.2$	0.9370 ± 0.0030	0.9370 ± 0.0027	0.9366 ± 0.0010	0.9352 ± 0.0010	0.9333 ± 0.0013
$w = 0.3$	0.9369 ± 0.0033	0.9368 ± 0.0025	0.9364 ± 0.0009	0.9348 ± 0.0014	0.9331 ± 0.0013
$w = 0.4$	0.9368 ± 0.0036	0.9366 ± 0.0023	0.9359 ± 0.0007	0.9344 ± 0.0013	0.9323 ± 0.0012
$w = 0.5$	0.9369 ± 0.0039	0.9363 ± 0.0023	0.9356 ± 0.0007	0.9339 ± 0.0014	0.9316 ± 0.0013
$w = 0.6$	0.9370 ± 0.0040	0.9364 ± 0.0023	0.9355 ± 0.0006	0.9335 ± 0.0012	0.9309 ± 0.0013
$w = 0.7$	0.9366 ± 0.0039	0.9362 ± 0.0021	0.9354 ± 0.0004	0.9335 ± 0.0011	0.9308 ± 0.0013
$w = 0.8$	0.9362 ± 0.0038	0.9357 ± 0.0020	0.9349 ± 0.0004	0.9329 ± 0.0010	0.9300 ± 0.0014
$w = 0.9$	0.9357 ± 0.0038	0.9352 ± 0.0019	0.9343 ± 0.0004	0.9322 ± 0.0010	0.9291 ± 0.0014
$w = 1.0$	0.9353 ± 0.0037	0.9347 ± 0.0019	0.9337 ± 0.0004	0.9314 ± 0.0009	0.9281 ± 0.0015

Table 11: Memorization alignment with FZ scores (Feldman & Zhang, 2020) on CIFAR-100.

Base quantity	CS ↑	Pearson ↑	Spearman ↑
ratio	0.7400	0.7040	0.8260
d_{own}	0.7190	0.5930	0.6782
$d_{nearest-other}$	0.5880	-0.1890	-0.2590
entropy	0.8387	0.7248	0.8229
1-p_{own}	0.8600	0.7604	0.8443

Table 12: Memorization alignment with FZ scores (Feldman & Zhang, 2020) on ImageNet dataset.

Base quantity	CS ↑	Pearson ↑	Spearman ↑
ratio	0.7499	0.6337	0.7884
d_{own}	0.7390	0.5819	0.6268
$d_{nearest-other}$	0.5873	-0.0669	-0.1236
entropy	0.7920	0.6329	0.7180
1-p_{own}	0.8396	0.7366	0.7901

Table 13: AUROC of different metrics in representation space for asymmetric CIFAR-100 mislabeling task over three seeds

Base quantity	5%	10%	20%	25%	30%
ratio	0.9362 ± 0.0034	0.9360 ± 0.0020	0.9348 ± 0.0007	0.9323 ± 0.0009	0.9283 ± 0.0016
1-p_{own}	0.9353 ± 0.0037	0.9347 ± 0.0019	0.9337 ± 0.0004	0.9314 ± 0.0009	0.9281 ± 0.0015
d_{own}	0.8359 ± 0.0051	0.8233 ± 0.0006	0.8001 ± 0.0013	0.7838 ± 0.0020	0.7637 ± 0.0064
$d_{nearest-other}$	0.6642 ± 0.0056	0.6607 ± 0.0019	0.6475 ± 0.0022	0.6424 ± 0.0031	0.6288 ± 0.0004
entropy	0.8339 ± 0.0045	0.8239 ± 0.0028	0.8206 ± 0.0020	0.8182 ± 0.0002	0.8217 ± 0.0024

Table 14: AUROC of different metrics in representation space for symmetric CIFAR-10 mislabeling task in three seeds

Base quantity	5%	10%	20%	25%	30%
ratio	0.9956 ± 0.0002	0.9946 ± 0.0007	0.9934 ± 0.0007	0.9925 ± 0.0006	0.9913 ± 0.0007
1-p_{own}	0.9957 ± 0.0001	0.9952 ± 0.0003	0.9943 ± 0.0003	0.9930 ± 0.0005	0.9920 ± 0.0007
d_{own}	0.9862 ± 0.0001	0.9814 ± 0.0031	0.9727 ± 0.0041	0.9709 ± 0.0018	0.9623 ± 0.0058
$d_{nearest-other}$	0.7328 ± 0.0207	0.7600 ± 0.0602	0.7748 ± 0.0443	0.7992 ± 0.0302	0.7671 ± 0.0271
entropy	0.9570 ± 0.0009	0.9554 ± 0.0060	0.9449 ± 0.0026	0.9377 ± 0.0004	0.9381 ± 0.0026

Table 15: AUROC of different metrics in representation space for symmetric CIFAR-100 mislabeling task in three seeds

Base quantity	5%	10%	20%	25%	30%
ratio	0.9761 ± 0.0017	0.9785 ± 0.0007	0.9777 ± 0.0013	0.9775 ± 0.0002	0.9761 ± 0.0004
1-p_{own}	0.9715 ± 0.0014	0.9740 ± 0.0008	0.9733 ± 0.0010	0.9722 ± 0.0008	0.9704 ± 0.0001
d_{own}	0.9681 ± 0.0026	0.9669 ± 0.0009	0.9587 ± 0.0014	0.9562 ± 0.0014	0.9525 ± 0.0008
$d_{nearest-other}$	0.5412 ± 0.0028	0.5005 ± 0.0055	0.5451 ± 0.0060	0.5582 ± 0.0047	0.5622 ± 0.0057
entropy	0.8788 ± 0.0015	0.8726 ± 0.0020	0.8617 ± 0.0067	0.8629 ± 0.0034	0.8606 ± 0.0031

Table 16: Temperature sweep on CIFAR-100. Agreement of our score with FZ scores. $\tau = 0.02$ is the setting used in the main text.

τ	CS ↑	Pearson ↑	Spearman ↑
0.003	0.831	0.724	0.817
0.01	0.841	0.737	0.829
0.02	0.859	0.760	0.844
0.03	0.859	0.766	0.844
0.05	0.806	0.713	0.811
1.0	0.615	0.574	0.678

Table 17: Temperature sweep on ImageNet. Agreement of our score with FZ scores. $\tau = 0.02$ is the setting used in the main text.

τ	CS ↑	Pearson ↑	Spearman ↑
0.003	0.830	0.708	0.745
0.01	0.840	0.725	0.767
0.02	0.840	0.737	0.790
0.03	0.808	0.709	0.793
0.05	0.711	0.583	0.741
1.0	0.616	0.577	0.618