DeDD: Decontaminating Dataset Distillation under Label Noise

Anonymous authors
Paper under double-blind review

Abstract

Dataset distillation (DD) compresses a large training set into a small synthetic set on which models can be trained to achieve accuracy comparable to that obtained on the original data. Most existing DD methods, however, assume that accurate gold labels are readily available, whereas annotations of large-scale datasets are inevitably corrupted by noise. Under such label noise, existing DD methods degrade substantially, even when preceded by an additional label-cleaning stage. To address this problem, we propose *Decontaminating Dataset Distillation* (DeDD), a framework that distills a clean synthetic set directly from label-corrupted training data. By modeling the observed class-conditional distributions as mixtures of their latent clean counterparts and analytically inverting this mixture, our method recovers unbiased estimates of the clean class prototypes and matches the synthetic set to them in a single stage, without explicit label cleaning or sample selection. Theoretically, our objective yields, in expectation, the same optimization gradients as distillation on clean data. Experiments with simulated and real-world label noise demonstrate that DeDD outperforms state-of-the-art DD methods.

1 Introduction

The recent success of deep learning has relied heavily on the availability of large labeled datasets. However, training state-of-the-art models on such datasets is financially costly and environmentally demanding, and this cost is paid repeatedly during model selection and hyperparameter optimization. To address this challenge, *dataset distillation* (DD) (Wang et al., 2018) aims to synthesize a compact set of informative samples whose training utility approximates that of the full dataset. Subsequent approaches, including matching-based algorithms (Zhao et al., 2021; Lee & Chung, 2024), decoupling-based methods (Yin et al., 2023; Shao et al., 2024; Wang et al., 2026), selection-based methods (Sun et al., 2024; Hu et al., 2026), generation-based models (Su et al., 2024; 2026), and versatile plug-and-play schemes (Shang et al., 2025; Cui et al., 2025), have improved compression efficiency while maintaining competitive accuracy.

Despite substantial progress, an important limitation has received little attention: most DD methods implicitly assume access to perfect *gold labels* for the training data. However, this assumption is often violated in real-world settings (Zhou, 2018), especially for the large-scale datasets that most need distillation and whose labels are rarely verified manually. In settings such as web-crawled data (Niu et al., 2018), crowdsourced labeling (Raykar et al., 2010; Rodrigues & Pereira, 2018), and medical imaging (Tajbakhsh et al., 2020), annotations are often inaccurate, and the resulting corrupted supervision gives rise to the problem of DD under label noise.

When existing DD methods are applied to label-corrupted data, their performance can degrade substantially. A simple countermeasure is a two-stage pipeline: first filter or relabel the data using a label-cleaning algorithm (Han et al., 2018; Li et al., 2020), and then distill the resulting dataset. This strategy, however, has two limitations. First, noise elimination is rarely complete. The obtained dataset may still contain inaccurate labels, which brings the task back to the original label-noise setting. Second, cleaning alters the data distribution itself. Filtering discards hard yet informative samples, and relabeling introduces the errors of a proxy network, so distillation targets a distorted version of the clean data distribution. These limitations stem from decoupling the denoising objective from the distillation objective, allowing errors to propagate across stages rather than being corrected under a unified training objective. A recent one-stage method, robust dataset condensation (RDC) (Kim

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.

& Song, 2025), mitigates label noise through contrastive learning with golden MixUp (Zhang et al., 2018), but offers no guarantee of recovering the clean distribution and is computationally expensive.

In this paper, we introduce *Decontaminating Dataset Distillation* (DeDD), a framework that directly distills a synthetic dataset from label-corrupted training data. Our method models the observed class-conditional densities as linear mixtures of latent clean densities with class-specific mixture parameters, which follows from the class-wise instance-independent noise model in noisy-label learning (Natarajan et al., 2013; Patrini et al., 2017). Under a mild mutual irreducibility condition, which is reasonable for semantically distinct classes, the mixture parameters are identifiable and can be estimated via mixture proportion estimation (MPE) (Scott, 2015), after which inverting the mixture yields the clean class means in closed form. Plugging these recovered statistics into the distribution matching (DM) objective (Zhao & Bilen, 2023) gives a one-stage decontaminating distillation loss that targets the clean distribution directly, without explicit label cleaning or sample selection. We further add a prototype margin loss that repels synthetic class representations from recovered negative class prototypes, improving inter-class separation beyond distribution alignment alone. Our contributions can be summarized as follows:

- *Decontaminating framework*: We cast DD under label noise as a decontamination problem and propose a method that recovers unbiased estimates of the clean class prototypes from label-corrupted data by closed-form mixture inversion and distills from them in one stage.
- *Theoretical guarantees*: We prove that the decontaminated matching loss yields, in expectation, the same gradients as distillation on clean data (Theorem 1) and upper-bounds the loss deviation between models trained on the distilled data and the clean data (Theorem 2).
- *Experiments*: Extensive experiments across multiple benchmark datasets show that our method achieves competitive performance against existing DD baselines while being more computationally efficient than the existing noise-robust DD method.

2 Preliminaries

In this section, we first review the standard formulation of DD and the distribution matching objective on which our method builds, and then formalize the problem of DD under label noise.

2.1 Dataset Distillation

Let $\mathcal{R} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ denote the original dataset of n samples, where each pair is drawn i.i.d. from a joint distribution with density $p(\boldsymbol{x}, y)$ over $\mathcal{X} \times \mathcal{Y}$. Here $\mathcal{X} \subseteq \mathbb{R}^d$ is the input space and $\mathcal{Y} = \{1, \ldots, K\}$ is the label space. Let $\Phi_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^K$ denote a model parameterized by $\boldsymbol{\theta}$. DD aims to construct a much smaller synthetic dataset $\mathcal{S} = \{(\boldsymbol{s}_j, y'_j)\}_{j=1}^{|\mathcal{S}|}$, with $|\mathcal{S}| \ll |\mathcal{R}|$, where $(\boldsymbol{s}_j, y'_j) \in \mathcal{X} \times \mathcal{Y}$ is the j-th synthetic sample–label pair (Sachdeva & McAuley, 2023; Lei & Tao, 2024). The goal is for a model $\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}$ trained on $\mathcal{S}$ to match the held-out performance of a model $\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}$ trained on the full dataset $\mathcal{R}$. Letting ℓ denote a loss function (e.g., cross-entropy), we quantify the fidelity of such a synthetic set through the notion of an ϵ-approximate data summary, defined below.

Definition 1 (ϵ-Approximate Data Summary (Sachdeva & McAuley, 2023))**.** Given a model $\Phi_{\boldsymbol{\theta}}$, let $\boldsymbol{\theta}^{\mathcal{R}}$ and $\boldsymbol{\theta}^{\mathcal{S}}$ denote the optimal parameters obtained by training $\Phi_{\boldsymbol{\theta}}$ on $\mathcal{R}$ and $\mathcal{S}$, respectively. For $\epsilon > 0$, we say that $\mathcal{S}$ is an ϵ-approximate data summary of $\mathcal{R}$ if

$$\sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)} \left|\ell(\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y) - \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}(\boldsymbol{x}), y)\right| \leq \epsilon. \tag{1}$$

To obtain such a high-quality data summary, DD therefore aims to seek a distilled set $\mathcal{S}$ that minimizes the worst-case loss discrepancy between the models trained on the original dataset and those trained on the distilled dataset, as formalized in the following definition.

Definition 2 (Dataset Distillation (Sachdeva & McAuley, 2023))**.** Under the same notation as above, the optimization problem of DD is formulated as

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)} \left|\ell(\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y) - \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}(\boldsymbol{x}), y)\right|. \tag{2}$$

DM (Zhao & Bilen, 2023) approximates this objective by aligning the class-wise feature representations of $\mathcal{S}$ and $\mathcal{R}$ via first-order moment matching. Specifically, DM obtains the distilled dataset by minimizing the following maximum mean discrepancy objective:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathbb{E}_{\boldsymbol{\vartheta}\sim P_{\boldsymbol{\vartheta}}} \sum_{c=1}^{K} \left\| \frac{1}{|\mathcal{R}^c|} \sum_{\boldsymbol{x}\in\mathcal{R}^c} \psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}) - \frac{1}{|\mathcal{S}^c|} \sum_{\boldsymbol{s}\in\mathcal{S}^c} \psi_{\boldsymbol{\vartheta}}(\boldsymbol{s}) \right\|_2^2, \tag{3}$$

where $\psi_{\boldsymbol{\vartheta}} : \mathcal{X} \to \mathbb{R}^m$ is a randomly initialized encoder with parameters $\boldsymbol{\vartheta}$ drawn from an initialization distribution $P_{\boldsymbol{\vartheta}}$, $\mathcal{R}^c$ and $\mathcal{S}^c$ denote the subsets of $\mathcal{R}$ and $\mathcal{S}$ with label c, respectively, and $\|\boldsymbol{v}\|_2 := \sqrt{\boldsymbol{v}^\top \boldsymbol{v}}$ denotes the Euclidean norm of a vector $\boldsymbol{v} \in \mathbb{R}^m$.

2.2 Dataset Distillation under Label Noise

Despite the rapid progress of DD, most existing methods assume that the training labels are accurate (Yu et al., 2024). This assumption is frequently violated in practice, where annotations are corrupted by sample ambiguity, automated labeling, or human error. Let $\tilde{\mathcal{R}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^n$ denote a noisy-label version of $\mathcal{R}$, in which the observed label $\tilde{y}_i$ may be corrupted, i.e., $\tilde{y}_i \neq y_i$ for some samples, while the inputs $\boldsymbol{x}_i$ remain unchanged. Following the definition of standard DD (Sachdeva & McAuley, 2023), we formalize DD under label noise as follows.

Definition 3 (Dataset Distillation under Label Noise)**.** Under the above notation, and given access only to $\tilde{\mathcal{R}}$ (and not to the clean dataset $\mathcal{R}$), DD under label noise seeks the distilled set $\mathcal{S}^*$ given by

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)} \left| \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y) - \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}(\boldsymbol{x}), y) \right|. \tag{4}$$

The key difficulty is that the objective in Eq. (4) is defined with respect to the clean dataset $\mathcal{R}$, which is inaccessible in our setting. Consequently, any practical surrogate of it, e.g., the class-wise real means in Eq. (3), requires the clean partitions $\mathcal{R}^c$ and cannot be computed from $\tilde{\mathcal{R}}$ alone. This constitutes the central challenge: we must approximate the behavior of a clean-data oracle while observing only the noisy-label dataset $\tilde{\mathcal{R}}$. Most existing DD methods do not address this challenge, as they implicitly assume clean supervision and thus degrade when labels are corrupted.

3 Methodology

In this section, we first present the class-wise density decomposition, then derive the decontaminating dataset distillation loss and analyze it theoretically, and finally describe the practical implementation, including the estimation of the mixture parameters and the prototype margin loss.

3.1 Class-Wise Density Decomposition

In our setting, the clean dataset $\mathcal{R}$ is unavailable, and we observe only a noisy-label dataset $\tilde{\mathcal{R}}$, whose class-conditional densities are mixtures of the true class-conditional densities across classes. Let $p(\boldsymbol{x} \mid y = c)$ and $p(\boldsymbol{x} \mid y \neq c)$ denote the densities of inputs whose *true* label is c and is not c, respectively, and let $p(\boldsymbol{x} \mid \tilde{y} = c)$ and $p(\boldsymbol{x} \mid \tilde{y} \neq c)$ denote the densities of inputs whose *observed* label is c and is not c, respectively. The observed and true class-conditional densities are then related by the following lemma, whose detailed proof is given in Appendix A.1.

Lemma 1 (Class-Wise Density Decomposition)**.** *Under the instance-independent assumption that* $p(\tilde{y} = c \mid \boldsymbol{x}, y = c) = p(\tilde{y} = c \mid y = c)$ *and* $p(\tilde{y} = c \mid \boldsymbol{x}, y \neq c) = p(\tilde{y} = c \mid y \neq c)$*, we have*

$$\begin{cases} p(\boldsymbol{x} \mid \tilde{y} = c) = \alpha_c\, p(\boldsymbol{x} \mid y = c) + (1 - \alpha_c)\, p(\boldsymbol{x} \mid y \neq c), \\ p(\boldsymbol{x} \mid \tilde{y} \neq c) = (1 - \beta_c)\, p(\boldsymbol{x} \mid y = c) + \beta_c\, p(\boldsymbol{x} \mid y \neq c), \end{cases} \tag{5}$$

where $\alpha_c = p(y = c \mid \tilde{y} = c)$ *and* $\beta_c = p(y \neq c \mid \tilde{y} \neq c)$ *are class-specific mixture parameters.*

Remark 1. In Lemma 1, we adopt a class-wise instance-independent noise assumption, which is closely related to assumptions commonly used in noisy-label learning and enables theoretically grounded loss correction and statistically consistent learning (Natarajan et al., 2013; Patrini et al., 2017; Yao et al., 2020). Estimating an instance-dependent transition matrix is substantially more challenging and, without additional assumptions or supervision, may not be identifiable (Liu et al., 2023). We leave extensions to the instance-dependent setting for future work.

Both observed partitions in Eq. (5) are generally contaminated. For instance, under crowdsourced annotation, some images labeled "cat" in fact show other classes ($\alpha_c < 1$), while some true cats receive other labels and end up in the non-cat partition ($\beta_c < 1$). Following the common assumption that correct labeling is more likely than incorrect labeling (Natarajan et al., 2013; Cheng et al., 2021), we assume $\alpha_c + \beta_c > 1$, which ensures that the linear system in Eq. (5) is non-degenerate. Hence, given α_c and β_c, the latent clean densities are uniquely determined by the observed ones. When $\alpha_c = \beta_c = 1$ for every class c, the labels are clean and the problem reduces to standard DD.

3.2 Decontaminating Dataset Distillation

Although the clean class-conditional density is unobservable, the linear system in Eq. (5) can be inverted in closed form, expressing the former in terms of the two observed class-conditional densities and the mixture parameters, as formalized in the following proposition.

Proposition 1 (Recovery of Clean Class-Conditional Densities)**.** *Suppose $\alpha_c + \beta_c > 1$. Then the clean class-conditional density admits the closed-form expression:*

$$p(\boldsymbol{x} \,|\, y = c) = \frac{\beta_c \, p(\boldsymbol{x} \,|\, \tilde{y} = c) - (1 - \alpha_c) \, p(\boldsymbol{x} \,|\, \tilde{y} \neq c)}{\alpha_c + \beta_c - 1}. \tag{6}$$

The proof is given in Appendix A.2. Notably, the recovery requires neither explicit anchor points nor curated clean samples, but only knowledge of the mixture parameters α_c and β_c. Multiplying both sides of Eq. (6) by $\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})$ and integrating over $\boldsymbol{x}$ yields the corresponding identity for class-conditional feature means, which is all that distribution matching requires:

$$\mu_c^{\mathcal{R}} := \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|y=c)}[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})] = \frac{\beta_c \, \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\tilde{y}=c)}[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})] - (1 - \alpha_c) \, \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\tilde{y} \neq c)}[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})]}{\alpha_c + \beta_c - 1}. \tag{7}$$

That is, the clean class mean is a linear combination of the two observed class means, even though neither is clean on its own. Replacing the expectations in Eq. (7) by their empirical counterparts computed on the noisy-label dataset $\tilde{\mathcal{R}}$ gives the plug-in estimator:

$$\hat{\mu}_c = \frac{\beta_c \, \tilde{\mu}_c^{+} - (1 - \alpha_c) \, \tilde{\mu}_c^{-}}{\alpha_c + \beta_c - 1}, \qquad \tilde{\mu}_c^{+} = \frac{1}{|\tilde{\mathcal{R}}^c|} \sum_{\boldsymbol{x} \in \tilde{\mathcal{R}}^c} \psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}), \qquad \tilde{\mu}_c^{-} = \frac{1}{|\tilde{\mathcal{R}}^{\neg c}|} \sum_{\boldsymbol{x} \in \tilde{\mathcal{R}}^{\neg c}} \psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}), \tag{8}$$

where $\tilde{\mathcal{R}}^c$ denotes the subset of $\tilde{\mathcal{R}}$ with observed label c and $\tilde{\mathcal{R}}^{\neg c} := \tilde{\mathcal{R}} \setminus \tilde{\mathcal{R}}^c$ denotes its complement, so that $\tilde{\mu}_c^{+}$ and $\tilde{\mu}_c^{-}$ are the empirical feature means of the two observed partitions.

Optimization Objective. Substituting $\hat{\mu}_c$ for the real-data class mean in Eq. (3) yields our decontaminating dataset distillation loss, denoted $\mathcal{L}_{\mathrm{DeDD}}$:

$$\mathcal{L}_{\mathrm{DeDD}} = \mathbb{E}_{\boldsymbol{\vartheta} \sim P_{\boldsymbol{\vartheta}}} \sum_{c=1}^{K} \left\| \hat{\mu}_c - \frac{1}{|\mathcal{S}^c|} \sum\nolimits_{\boldsymbol{s} \in \mathcal{S}^c} \psi_{\boldsymbol{\vartheta}}(\boldsymbol{s}) \right\|_2^2. \tag{9}$$

Unlike two-stage pipelines that first denoise the supervision and then perform dataset distillation, the objective in Eq. (9) incorporates the correction directly into the matching objective, requiring no explicit label cleaning or sample selection and hence discarding no data.

3.3 Theoretical Analysis

Let $\mathcal{L}_{\mathrm{DM}} := \mathbb{E}_{\boldsymbol{\vartheta}} \sum_{c=1}^{K} \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2$ denote the population form of the standard DM objective in Eq. (3), where $\mu_c^{\mathcal{R}}$ is the expected clean class mean defined in Eq. (7) and $\mu_c^{\mathcal{S}} := \frac{1}{|\mathcal{S}^c|} \sum_{\boldsymbol{s} \in \mathcal{S}^c} \psi_{\boldsymbol{\vartheta}}(\boldsymbol{s})$ is the synthetic class mean. Theorem 1 shows that $\mathcal{L}_{\mathrm{DeDD}}$, although computed solely from label-corrupted data, is a statistically consistent surrogate for $\mathcal{L}_{\mathrm{DM}}$.

Theorem 1 (Decontamination Consistency)**.** *Suppose that $\alpha_c + \beta_c > 1$ for every class c and that $\tilde{\mathcal{R}}^c$ and $\tilde{\mathcal{R}}^{\neg c}$ are i.i.d. samples from $p(\boldsymbol{x} \mid \tilde{y} = c)$ and from $p(\boldsymbol{x} \mid \tilde{y} \neq c)$, respectively. Assume further that $\psi_{\boldsymbol{\vartheta}}$ is differentiable in its input with $\|\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|_2 \leq B$ and $\|\nabla_{\boldsymbol{x}} \psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|_{\mathrm{op}} \leq L$ for all $\boldsymbol{x}$, where $\|\cdot\|_{\mathrm{op}}$ denotes the operator norm induced by $\|\cdot\|_2$. Throughout, $\mathbb{E}_{\tilde{\mathcal{R}}}[\,\cdot\,]$ denotes expectation over the sampling of $\tilde{\mathcal{R}}$ and $\mathbb{E}_{\tilde{\mathcal{R}}}[\,\cdot \mid \boldsymbol{\vartheta}]$ the same expectation for a fixed encoder. Then:*

(a) ***(Gradient consistency)*** $\mathbb{E}_{\tilde{\mathcal{R}}}[\nabla_{\mathcal{S}}\mathcal{L}_{\mathrm{DeDD}}] = \nabla_{\mathcal{S}}\mathcal{L}_{\mathrm{DM}}$.

(b) ***(Expected-loss consistency)*** $\mathbb{E}_{\tilde{\mathcal{R}}}[\mathcal{L}_{\mathrm{DeDD}}] = \mathcal{L}_{\mathrm{DM}} + V$ *with* $V := \sum_{c=1}^{K} \mathbb{E}_{\boldsymbol{\vartheta}}[\mathrm{tr}\,\mathrm{Cov}[\hat{\mu}_c \mid \boldsymbol{\vartheta}]]$, *where the variance term* V *is* $O\left(1/|\tilde{\mathcal{R}}|\right)$ *and vanishes as* $|\tilde{\mathcal{R}}| \to \infty$.

Remark 2. Part (a) shows that, in expectation, the gradient of the DeDD loss coincides with that of the clean-data DM objective, so that optimizing $\mathcal{L}_{\mathrm{DeDD}}$ drives $\mathcal{S}$ toward the same solution as would be obtained with access to clean labels. Part (b) shows that the two objectives differ only by a variance term V that is independent of $\mathcal{S}$ and decays with $|\tilde{\mathcal{R}}|$. Hence they share the same minimizers in expectation, and the realized $\mathcal{L}_{\mathrm{DeDD}}$ converges to $\mathcal{L}_{\mathrm{DM}}$ as the number of training samples grows.

Theorem 2 builds on decontamination consistency to show that $\mathcal{L}_{\mathrm{DeDD}}$ controls the ideal objective in Eq. (2), with the slack vanishing as $O(|\tilde{\mathcal{R}}|^{-1/2})$.

Theorem 2 (Loss Deviation Bound)**.** *Fix a distilled set* $\mathcal{S}$. *For a fixed encoder* $\psi_{\boldsymbol{\vartheta}}$ *and* $\mathcal{A} \in \{\mathcal{R}, \mathcal{S}\}$, *let* $\Phi_{\boldsymbol{\theta}^{\mathcal{A}}}(\boldsymbol{x}) = (W^{\mathcal{A}})^{\top}\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})$ *with* $\boldsymbol{\theta}^{\mathcal{A}} = (\boldsymbol{\vartheta}, W^{\mathcal{A}})$ *be the linear probe trained on* $\mathcal{A}$ *by ridge regression, i.e.,* $W^{\mathcal{A}} = \arg\min_W \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\|W^{\top}\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}) - e_y\|_2^2 + \lambda\|W\|_{\mathrm{F}}^2$ *with* $\lambda > 0$, *where* $e_y \in \{0,1\}^K$ *is the one-hot encoding of* y, $\|W\|_{\mathrm{F}} := \sqrt{\mathrm{tr}(W^{\top}W)}$ *is the Frobenius norm, and the expectation is over* $p(\boldsymbol{x}, y)$ *for* $\mathcal{A} = \mathcal{R}$. *Assume that* $\ell(\cdot, y)$ *is* L_ℓ*-Lipschitz in its logit argument w.r.t.* $\|\cdot\|_2$ *and that the conditions of Theorem 1 and Assumptions 1 and 2 (Appendix A.4) hold. Then, with* B *as in Theorem 1 and* $\pi_{\max} := \max_c p(y = c)$, *we have*

$$\mathbb{E}_{\boldsymbol{\vartheta}}\Big[\sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)} \big|\ell(\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y) - \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}(\boldsymbol{x}), y)\big|\Big] \;\le\; \frac{L_\ell B\,\pi_{\max}}{\lambda}\sqrt{\mathbb{E}_{\tilde{\mathcal{R}}}[\mathcal{L}_{\mathrm{DeDD}}]} \;=:\; \bar{\epsilon}. \tag{10}$$

Corollary 1 (Data-Summary Guarantee)**.** *Under the conditions of Theorem 2, for any* $\eta \in (0,1)$, *with probability at least* $1-\eta$ *over the draw of* $\boldsymbol{\vartheta} \sim P_{\boldsymbol{\vartheta}}$, *the distilled set* $\mathcal{S}$ *is an* ϵ*-approximate data summary of* $\mathcal{R}$ *in the sense of Definition 1, with* $\epsilon = \bar{\epsilon}/\eta$.

Remark 3. Corollary 1 closes the loop opened by Definition 3. Although $\mathcal{L}_{\mathrm{DeDD}}$ is computed entirely from label-corrupted data, it inflates $\bar{\epsilon}$ over its clean-label counterpart by at most $L_\ell B\pi_{\max}\sqrt{V}/\lambda$, a term that vanishes as $|\tilde{\mathcal{R}}| \to \infty$ by Theorem 1(b), without ever accessing the clean dataset $\mathcal{R}$.

3.4 Practical Implementations

Estimation of Mixture Parameters (α_c, β_c)**.** The decontaminated target $\hat{\mu}_c$ in Eq. (8) depends on the unknown (α_c, β_c). We estimate them by a reduction to MPE. Let $\mathcal{U} = \kappa\mathcal{P} + (1-\kappa)\mathcal{N}$ be a mixture of two distributions with proportion $\kappa \in (0,1)$. Under the identifiability conditions, MPE estimates κ from i.i.d. samples drawn from $\mathcal{U}$ and $\mathcal{P}$ without requiring samples from $\mathcal{N}$. Several consistent estimators are available (Ramaswamy et al., 2016; Jain et al., 2016; Garg et al., 2021).

Rearranging Eq. (5) recasts the estimation of unknown parameters as two MPE instances:

$$p(\boldsymbol{x} \mid \tilde{y} = c) = \kappa_c^{+}\, p(\boldsymbol{x} \mid \tilde{y} \neq c) + (1 - \kappa_c^{+})\, p(\boldsymbol{x} \mid y = c), \quad \kappa_c^{+} = \frac{1-\alpha_c}{\beta_c}, \tag{11}$$

$$p(\boldsymbol{x} \mid \tilde{y} \neq c) = \kappa_c^{-}\, p(\boldsymbol{x} \mid \tilde{y} = c) + (1 - \kappa_c^{-})\, p(\boldsymbol{x} \mid y \neq c), \quad \kappa_c^{-} = \frac{1-\beta_c}{\alpha_c}. \tag{12}$$

In Eq. (11), $p(\boldsymbol{x} \mid \tilde{y} = c)$ plays the role of $\mathcal{U}$, and $p(\boldsymbol{x} \mid \tilde{y} \neq c)$ plays the role of $\mathcal{P}$, while the clean class-conditional density $p(\boldsymbol{x} \mid y = c)$ is the unobserved $\mathcal{N}$. Eq. (12) is a symmetric instance with the observed-label distributions interchanged. Note that although the known component, namely $p(\boldsymbol{x} \mid \tilde{y} \neq c)$ in Eq. (11) and $p(\boldsymbol{x} \mid \tilde{y} = c)$ in Eq. (12), is itself a corrupted mixture rather than a clean density, MPE requires only empirical access to $\mathcal{P}$, not its purity. The reduction is therefore valid because both observed-label distributions are accessible from $\tilde{\mathcal{R}}^{c}$ and its complement $\tilde{\mathcal{R}}^{\neg c}$.

At the population level, MPE identifies the maximal κ for which the implied residual remains a valid density (Scott, 2015). This maximum coincides with the true κ precisely when the latent residual is irreducible with respect to the known component:

Definition 4 (Mutual Irreducibility)**.** Two probability distributions $\mathcal{P}$ and $\mathcal{N}$ are mutually irreducible if neither contains the other as a mixture component: there exist no probability distribution $\mathcal{H}$ and $\gamma \in (0,1)$ such that $\mathcal{P} = (1-\gamma)\mathcal{N} + \gamma\mathcal{H}$, and no probability distribution $\mathcal{H}'$ and $\gamma' \in (0,1)$ such that $\mathcal{N} = (1-\gamma')\mathcal{P} + \gamma'\mathcal{H}'$.

Table 1: Clean-test accuracy (%) as mean ± std over 3 seeds. One-stage methods distill directly from the label-corrupted training set, whereas two-stage methods (†) first relabel it with DivideMix and then distill. In each column, the best result is in **bold** and the second best is underlined.

	CIFAR-10		CIFAR-100		Tiny-ImageNet		CIFAR-10N		CIFAR-100N	
IPC	**10**	**50**	**10**	**50**	**1**	**10**	**10**	**50**	**10**	**50**
Ratio (%)	0.2	1	2	10	0.2	2	0.2	1	2	10
Random	23.26±0.23	32.05±0.27	9.52±0.23	18.51±0.32	1.14±0.01	4.20±0.11	24.04±0.24	37.62±0.42	13.99±0.27	24.95±0.19
Herding	25.70±0.87	34.65±0.10	12.43±0.16	19.43±0.13	2.21±0.06	4.54±0.11	27.35±0.28	39.67±0.27	16.53±0.17	26.55±0.26
K-Center	12.90±0.42	13.64±0.35	2.31±0.07	2.86±0.09	1.08±0.09	1.17±0.05	13.68±0.38	16.77±0.27	5.02±0.04	9.94±0.33
DC	47.07±0.32	45.34±0.26	23.58±0.10	20.58±0.05	4.04±0.10	10.65±0.33	48.02±0.69	47.18±0.61	23.97±0.24	23.25±0.25
DSA	46.14±0.43	48.35±0.77	27.43±0.38	25.83±0.47	4.13±0.04	11.66±0.23	46.29±0.33	49.89±0.16	27.64±0.08	29.83±0.30
DM	43.99±0.36	43.96±0.38	23.93±0.25	22.44±0.12	3.63±0.09	9.49±0.12	44.88±0.80	43.59±0.67	24.57±0.30	22.72±0.15
IDM	44.44±0.35	56.63±0.56	23.61±0.32	34.80±0.25	4.16±0.07	11.35±0.06	45.03±0.63	58.03±0.44	25.09±0.28	34.61±0.42
MTT	54.66±1.21	58.06±0.23	29.76±0.08	28.28±0.15	1.56±0.03	5.15±0.02	53.51±1.37	60.17±0.93	29.30±0.25	29.60±0.32
DATM	46.95±0.88	59.20±0.37	29.41±0.18	33.91±0.23	4.17±0.08	8.84±0.13	49.40±0.62	60.66±0.20	27.59±0.11	36.06±0.14
TGDD	42.43±0.35	58.10±0.78	20.79±0.45	35.50±0.37	2.73±0.05	9.34±0.03	42.63±0.50	58.25±0.34	23.22±0.30	34.62±0.13
AT-BPTT	43.35±0.63	55.18±0.13	21.54±0.18	32.15±0.12	1.67±0.05	8.42±0.24	44.15±0.72	46.52±0.72	24.54±0.53	31.35±0.91
NCFM	50.29±0.62	57.04±0.57	27.66±0.04	36.12±0.11	2.84±0.11	10.14±0.07	42.12±0.33	48.83±0.31	23.43±0.16	27.37±0.24
DC†	50.18±0.32	56.27±0.16	27.76±0.02	29.78±0.17	4.50±0.18	13.29±0.29	50.16±0.32	56.18±0.26	26.72±0.40	28.61±0.36
DSA†	50.87±0.72	58.68±0.17	31.59±0.53	37.64±0.07	4.51±0.07	13.82±0.38	51.40±0.14	59.18±0.40	30.33±0.22	36.02±0.31
DM†	46.72±0.22	46.83±0.07	28.21±0.18	25.17±0.38	3.94±0.11	11.13±0.12	48.18±0.69	45.80±0.12	26.84±0.32	24.44±0.05
IDM†	47.13±0.48	60.71±0.68	28.67±0.14	39.73±0.07	4.64±0.24	13.92±0.30	46.41±0.60	61.23±0.39	26.67±0.03	36.23±0.06
MTT†	57.30±0.61	65.47±0.37	33.18±0.21	37.53±0.09	1.82±0.07	11.89±0.46	54.93±0.86	66.16±0.44	30.76±0.15	36.52±0.17
DATM†	56.75±0.50	65.01±0.15	28.55±0.18	40.37±0.27	4.68±0.14	9.25±0.15	52.15±0.39	**67.65±0.36**	27.42±0.18	37.89±0.20
TGDD†	46.75±0.27	61.06±0.32	29.00±0.11	40.94±0.49	3.41±0.07	12.83±0.12	46.86±0.19	60.52±0.27	27.25±0.06	36.13±0.06
AT-BPTT†	44.06±0.81	58.32±0.51	22.21±0.17	29.45±0.12	2.27±0.08	9.02±0.05	46.24±0.82	49.62±0.62	27.52±0.81	32.54±0.19
NCFM†	54.48±0.17	60.20±0.37	28.79±0.24	38.60±0.22	4.24±0.24	12.24±0.30	49.36±0.48	60.19±0.27	27.38±0.17	34.37±0.17
RDC	48.99±0.66	59.86±0.72	31.54±0.06	37.01±0.12	4.03±0.11	13.53±0.13	48.96±0.12	60.38±0.32	27.10±0.27	35.54±0.22
Ours	**57.86±0.42**	**65.70±0.22**	**34.64±0.33**	**41.98±0.43**	**5.63±0.21**	**14.84±0.34**	**55.81±0.12**	66.83±0.25	**31.89±0.28**	**38.44±0.17**

A sufficient condition for both MPE instances to be identifiable is that the latent clean densities $p(\boldsymbol{x} \mid y = c)$ and $p(\boldsymbol{x} \mid y \neq c)$ are mutually irreducible, which may be reasonable for sufficiently distinct classes. Under this condition, a consistent MPE estimator yields consistent estimates $\hat{\kappa}_c^+$ and $\hat{\kappa}_c^-$. Inverting the relations between (κ_c^+, κ_c^-) and (α_c, β_c) then yields the plug-in estimates:

$$\hat{\alpha}_c = \frac{1 - \hat{\kappa}_c^+}{1 - \hat{\kappa}_c^+ \hat{\kappa}_c^-}, \qquad \hat{\beta}_c = \frac{1 - \hat{\kappa}_c^-}{1 - \hat{\kappa}_c^+ \hat{\kappa}_c^-}, \tag{13}$$

which we substitute into Eq. (8) to construct the decontaminated target without label cleaning. The full procedure is summarized in Algorithm 1 of Appendix C.

Prototype Margin Loss. The objective in Eq. (9) aligns the synthetic class mean with the positive prototype $\hat{\mu}_c$ but does not separate it from the negative one. Following Proposition 1, the same mixture inversion, solved for $p(\boldsymbol{x} \mid y \neq c)$, yields a prototype for samples not in class c:

$$\hat{\mu}_{\neg c} = \frac{\alpha_c \tilde{\mu}_c^- - (1 - \beta_c)\tilde{\mu}_c^+}{\alpha_c + \beta_c - 1}, \tag{14}$$

where $\tilde{\mu}_c^+$ and $\tilde{\mu}_c^-$ are defined in Eq. (8). We further introduce a *prototype margin loss* that encourages $\mu_c^{\mathcal{S}}$ to be closer to the positive prototype than to the negative one by a margin $\delta > 0$:

$$\mathcal{L}_{\text{margin}} = \mathbb{E}_{\boldsymbol{\vartheta} \sim P_{\boldsymbol{\vartheta}}} \sum_{c=1}^{K} \max\big(0,\, \delta + \cos\big(\mu_c^{\mathcal{S}}, \hat{\mu}_{\neg c}\big) - \cos\big(\mu_c^{\mathcal{S}}, \hat{\mu}_c\big)\big)\,, \tag{15}$$

where $\cos(\boldsymbol{u}, \boldsymbol{v}) := \boldsymbol{u}^\top \boldsymbol{v} / (\|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2)$ denotes the cosine similarity between vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. The cosine hinge loss pushes the synthetic class mean toward the recovered positive prototype and away from the negative one, enhancing inter-class discriminability beyond mean alignment alone. The final training objective combines the decontaminating distillation loss with the margin loss:

$$\mathcal{L} = \mathcal{L}_{\text{DeDD}} + \lambda_{\text{m}} \mathcal{L}_{\text{margin}}, \tag{16}$$

where $\lambda_{\text{m}} > 0$ is a weighting coefficient for the prototype margin loss.

4 Experiments

In this section, we evaluate our method on benchmarks with synthetic and real-world label noise, comparing accuracy, generalization, and efficiency with prior methods, followed by analyses of decontamination fidelity, ablations, hyperparameter sensitivity, and robustness to other noise structures.

Table 2: Clean-test accuracy (%) on five ImageNet subsets at 128×128. All methods are one-stage without relabeling. In each column, the best result is in **bold** and the second best is underlined.

	ImageNette		ImageFruit		ImageMeow		ImageSquawk		ImageYellow	
IPC	**1**	**10**	**1**	**10**	**1**	**10**	**1**	**10**	**1**	**10**
Ratio (%)	0.105	1.05	0.077	0.77	0.077	0.77	0.077	0.77	0.077	0.77
Random	7.87±0.50	21.27±2.02	8.80±0.71	11.13±1.20	9.53±0.19	12.53±0.57	10.07±0.25	14.33±1.09	10.67±0.25	21.47±1.36
DC	20.00±1.18	24.00±1.93	12.60±0.71	13.93±1.37	14.73±0.38	12.07±0.66	23.07±0.19	21.07±3.49	24.47±1.52	24.87±1.41
DSA	21.73±1.91	18.33±0.96	13.93±1.55	12.80±1.02	12.80±0.57	12.80±1.23	21.20±1.28	18.60±3.10	23.40±0.98	18.60±2.08
DM	20.27±1.32	26.13±1.86	14.00±0.43	13.73±0.98	13.93±1.37	12.93±2.46	19.27±1.23	21.73±2.25	22.27±1.31	31.33±2.22
IDM	16.47±0.09	30.07±3.09	11.00±1.73	17.27±0.90	12.67±0.50	16.60±2.29	14.13±0.77	20.33±1.91	14.87±1.11	31.40±1.42
MTT	14.07±0.90	15.67±0.84	9.87±0.96	11.00±0.28	10.80±0.59	11.47±0.41	14.07±1.46	14.13±0.68	12.40±1.23	16.33±0.82
DATM	14.13±0.66	22.53±1.79	9.73±0.66	16.00±1.73	9.73±1.00	15.73±0.41	10.20±1.07	20.53±1.18	14.93±1.73	26.27±0.47
TGDD	22.67±1.91	29.60±1.85	13.67±0.84	16.87±2.74	11.33±1.23	13.73±0.47	20.53±2.03	30.00±1.30	21.00±0.98	28.80±2.21
AT-BPTT	23.40±1.41	19.07±1.43	16.87±1.06	17.20±0.33	14.93±1.16	14.47±1.33	22.87±1.27	18.13±1.48	25.40±0.57	32.53±2.53
NCFM	20.00±0.75	22.07±0.98	18.27±1.00	19.27±0.41	9.87±0.96	12.33±0.81	20.80±0.65	29.13±0.47	28.67±0.19	37.87±0.41
RDC	20.25±0.15	33.15±1.19	15.52±0.63	16.46±0.93	9.67±0.57	14.13±0.82	24.49±0.34	32.56±0.59	28.73±0.28	40.73±0.49
Ours	**32.73**±1.79	**54.00**±2.99	**20.13**±2.23	**28.40**±1.48	**15.80**±1.03	**27.73**±2.45	**31.00**±1.23	**50.67**±2.04	**35.87**±0.25	**50.00**±2.16

4.1 Experimental Setup

Noisy-label Datasets. We evaluate on Fashion-MNIST (Xiao et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), Tiny-ImageNet (Le et al., 2015), and multiple ImageNet subsets (Cazenavette et al., 2022), together with the real-world CIFAR-10N and CIFAR-100N (Wei et al., 2022). On the former, we build noisy-label variants in which the samples labeled c consist of a fraction α_c drawn from class c and a fraction $1 - \alpha_c$ drawn uniformly from the other classes, leaving the test set clean. For simplicity we use the same value $\alpha_c = \alpha$ for every class, with $\alpha = 0.7$ (30% corruption) on Tiny-ImageNet and $\alpha = 0.6$ (40%) on all others. Appendix D.2 gives the full construction and Appendix D.5 the complete results.

Baselines. To isolate the effect of noisy supervision, we compare with baselines under the same backbone, excluding methods that use external priors such as pretrained teacher models, generative priors or teacher-generated soft labels (Yin et al., 2023; Gu et al., 2024; Dey et al., 2026). The baselines mainly split into two groups: (1) *Coreset selection methods*: Random, Herding (Welling, 2009), and K-Center (Sener & Savarese, 2018). (2) *Dataset distillation methods*: DC (Zhao et al., 2021), DSA (Zhao & Bilen, 2021), MTT (Cazenavette et al., 2022), DM (Zhao & Bilen, 2023), IDM (Zhao et al., 2023), DATM (Guo et al., 2024), AT-BPTT (Li et al., 2025), TGDD (Ran et al., 2026), NCFM (Wang et al., 2025), and RDC (Kim & Song, 2025). For each distillation method we also build a *two-stage* variant (†), in which DivideMix (Li et al., 2020) relabels the training set before the same method distills it.

Implementation Details. A 3-layer ConvNet serves as both the random encoder ψ_{ϑ} and the evaluation network. Instead of the common real-image initialization, we initialize the synthetic images from Gaussian noise to avoid seeding them with mislabeled examples, and distill for 20,000 iterations with 256 real images per class from each observed partition. The MPE step of Algorithm 1 is solved with TEDN (Garg et al., 2021), and the margin loss uses $\lambda_{\mathrm{m}} = 0.1$ and $\delta = 1.0$ throughout. Following the standard protocol (Zhao & Bilen, 2023), we train a classifier from scratch on the distilled set and report its clean-test accuracy over 3 seeds. Appendix D.4 gives the remaining details.

4.2 Experimental Results

Performance Comparison. Tables 1 and 2 report the results under label noise. Coreset methods inherit the corrupted labels and lag behind, while DivideMix relabeling improves the baselines at the cost of an extra stage. In one stage without label correction, our method is best in most settings, even surpassing the two-stage variants. On CIFAR-10N and CIFAR-100N, it beats every one-stage baseline and is first in three of four settings, trailing only a two-stage variant at IPC 50 on CIFAR-10N. The margin is largest on the five ImageNet subsets, with a 9.4% average lead in all 10 settings.

Computational Efficiency. Table 3 compares the per-iteration distillation cost of our method with that of the noise-robust RDC on a single NVIDIA A100 (40GB), where the reduction is the ratio of the cost of RDC to ours. Matching decontaminated class prototypes in a single stage, without inner-loop network training or label correction, our method is markedly faster and lighter on both datasets. The

Table 3: Per-iteration time (s) and peak GPU memory (GB) of RDC and our method.

		Time (s/iter)			GPU Memory (GB)		
Dataset	IPC	RDC	Ours	Reduction	RDC	Ours	Reduction
CIFAR-100	10	4.47	1.76	**2.5**×	1.59	0.90	**1.8**×
	50	5.06	1.89	**2.7**×	4.42	1.04	**4.3**×
Tiny-ImageNet	10	4.88	2.85	**1.7**×	7.28	4.61	**1.6**×
	50	11.13	3.18	**3.5**×	18.91	5.73	**3.3**×

advantage widens with IPC, from 1.7–2.5× in time and 1.6–1.8× in memory at IPC 10 to 2.7–3.5× and 3.3–4.3× at IPC 50, so it achieves higher accuracy at a lower cost.

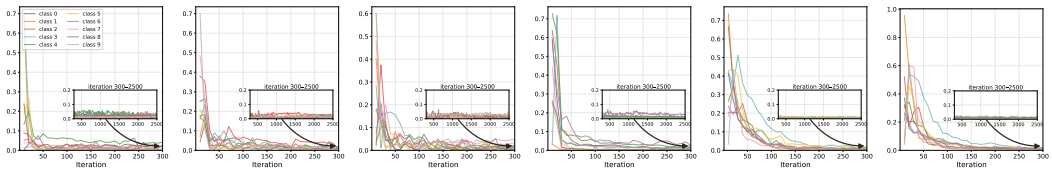
Cross-Architecture Evaluation. Table 4 evaluates data distilled with ConvNet-3 on unseen AlexNet (Krizhevsky et al., 2012), VGG-11 (Simonyan & Zisserman, 2015), and ResNet-18 (He et al., 2016). On CIFAR-10, TGDD drops by 25.9% from ConvNet-3 to AlexNet, whereas our method drops by at most 4.5% and ranks first on every architecture. The same holds on ImageNette (Table 10), where it leads by up to 22.9% on the unseen ones, so its distilled data capture class information rather than architecture-specific artifacts.

Table 4: Cross-architecture accuracy on CIFAR-10 at IPC 50.

	ConvNet-3	AlexNet	VGG-11	ResNet-18
DC	45.34±0.26	30.37±0.44	33.90±0.21	36.25±0.90
DSA	48.35±0.77	30.46±0.41	34.39±0.61	37.27±0.25
DM	43.96±0.38	31.71±0.54	34.07±0.94	35.37±1.40
IDM	56.63±0.56	38.47±0.33	43.53±0.16	44.32±1.29
MTT	58.06±0.23	48.43±0.63	54.19±0.92	50.88±0.58
DATM	59.20±0.37	51.86±0.39	56.76±0.83	55.40±0.79
TGDD	58.10±0.78	32.22±1.10	50.91±0.63	50.29±2.17
Ours	**65.70**±0.22	**61.23**±0.40	**62.32**±0.15	**61.57**±0.05

4.3 Experimental Analysis

Accuracy of Mixture Parameter Estimation. Decontamination hinges on accurate estimates of (α_c, β_c). As oracles, we take the empirical frequencies of $p(y = c \mid \tilde{y} = c)$ and $p(y \neq c \mid \tilde{y} \neq c)$ computed from the true labels, which are known by construction under simulated noise and released with CIFAR-10N. Figure 1 reports the per-class absolute error of the TEDN estimates against these oracles on three datasets. The error drops to a small value within a few iterations on all of them.

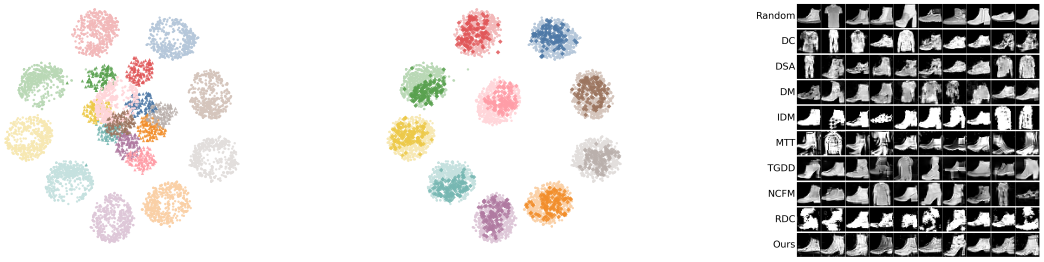


(a) α, F-MNIST (b) α, CIFAR-10 (c) α, CIFAR-10N (d) β, F-MNIST (e) β, CIFAR-10 (f) β, CIFAR-10N

Figure 1: Absolute error between the estimated and oracle mixture parameters over TEDN iterations for α (a–c) and β (d–f) on Fashion-MNIST, CIFAR-10, and CIFAR-10N. Each curve is one class.

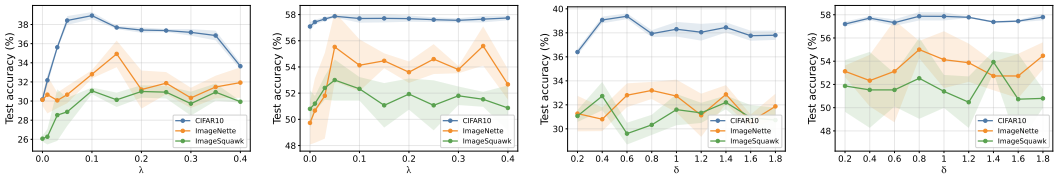
Recovery of Clean Class Prototypes. Figure 2(a,b) contrasts the matching targets of DM and our method on CIFAR-10 by t-SNE. The contaminated prototypes $\tilde{\mu}_c^+$ in (a) are dragged toward the other classes, away from their own clusters, so matching them would inject this bias into the distilled data. The decontaminated prototypes $\hat{\mu}_c$ in (b) instead fall back within the oracle prototypes of their classes, confirming that the mixture inversion of Eq. (8) removes the contamination.

Class Consistency of Distilled Images. Figure 2(c) shows the Fashion-MNIST "ankle boot" images randomly selected or distilled by different methods under label noise. Since the baselines rely directly on the observed partition of this class, which also contains other classes, most of them mix in images of shirts or trousers. Our method instead matches the decontaminated prototype, so all of its images remain ankle boots, consistent with the prototype recovery in Figure 2(b).



(a) Contaminated prototypes $\tilde{\mu}_c^+$ (b) Decontaminated prototypes $\hat{\mu}_c$ (c) Distilled "ankle boot" images

Figure 2: (a,b) t-SNE of class prototypes on CIFAR-10 before and after decontamination, showing oracle prototypes from clean labels (•), contaminated prototypes $\tilde{\mu}_c^+$ (▲) in (a), and decontaminated prototypes $\hat{\mu}_c$ (♦) in (b), each computed from one random batch and colored by class. (c) Fashion-MNIST "ankle boot" images selected randomly or distilled by different methods.



(a) λ_m, IPC=1 (fixed $\delta = 1.0$) (b) λ_m, IPC=10 (fixed $\delta = 1.0$) (c) δ, IPC=1 (fixed $\lambda_m = 0.1$) (d) δ, IPC=10 (fixed $\lambda_m = 0.1$)

Figure 3: Sensitivity to the weighting coefficient λ_m (a,b) and the margin δ (c,d) on CIFAR-10, ImageNette, and ImageSquawk, with the other hyperparameter fixed at its default value.

Table 5: Ablation of decontamination and prototype margin (clean-test accuracy, %).

Dataset	IPC	Enabled components –	Decontam.	Margin.	Both
Fashion-MNIST	10	82.76	83.59	83.59	**84.16**
	50	83.70	85.45	84.44	**85.87**
CIFAR-10	10	53.93	57.09	54.27	**57.69**
	50	62.57	65.48	61.34	**65.86**
CIFAR-100	10	30.65	34.33	33.32	**34.34**
	50	38.49	41.85	39.94	**42.69**
ImageNette	1	20.07	30.13	26.07	**32.20**
	10	41.73	49.73	42.20	**54.00**
ImageSquawk	1	22.47	26.07	21.60	**31.80**
	10	27.27	50.80	39.53	**50.87**
ImageYellow	1	27.67	35.40	28.13	**36.40**
	10	39.53	48.33	46.53	**51.33**
Average		44.24	50.69	46.75	**52.27**
Gain over baseline		–	+6.45	+2.51	**+8.03**

Table 6: Robustness to other noise structures (clean-test accuracy, %).

	Noise type	IPC	DC	DM	IDM	MTT	Ours
CIFAR-10	Shared α	10	47.1	44.0	44.4	54.7	**57.9**
		50	45.3	44.0	56.6	58.1	**65.7**
	Per-class α_c	10	46.1	42.6	44.0	53.1	**57.3**
		50	45.1	43.1	55.8	56.8	**64.9**
	Sym. flips	10	47.0	43.5	44.7	53.9	**56.9**
		50	45.9	43.3	56.4	58.5	**65.6**
	Asym. flips	10	46.7	44.0	45.4	43.6	**53.9**
		50	50.6	44.3	56.7	54.6	**62.3**
CIFAR-100	Shared α	10	23.6	23.9	23.6	29.8	**34.6**
		50	20.6	22.4	34.8	28.3	**42.0**
	Per-class α_c	10	23.6	23.6	23.2	29.0	**34.3**
		50	20.3	22.4	34.2	29.5	**41.4**
	Sym. flips	10	23.1	23.7	23.5	29.4	**34.5**
		50	20.5	22.6	34.1	28.6	**41.6**
	Asym. flips	10	18.4	18.4	17.2	22.2	**23.0**
		50	16.5	17.3	24.9	21.5	**27.0**

4.4 Ablation Studies

Effectiveness of Each Component. We ablate the two components in Table 5 to verify their effectiveness. The results show that decontamination is the primary driver, as it alone improves every setting and raises average accuracy by 6.45%. In contrast, the prototype margin alone yields smaller gains, as it targets inter-class separation rather than label noise. With both components, our method performs best in all 12 settings, indicating that the margin complements decontamination.

Sensitivity of Hyperparameters. In Figure 3, we vary the weighting coefficient λ_m and the margin δ of the prototype margin on CIFAR-10, ImageNette, and ImageSquawk, with the other fixed at its default. It can be observed that accuracy rises sharply once the margin loss is enabled and stays high over a wide range of λ_m peaking around 0.1, while varying δ has little effect. These trends hold on all datasets and IPCs, showing that our method is robust to its hyperparameters.

Robustness to Other Noise Structures. We further evaluate our method under other noise structures, detailed in Appendix D.3. Beyond the clean fraction α shared by all classes in our main setup, we consider per-class clean fractions α_c and symmetric or asymmetric label flips. As shown in Table 6, per-class α_c and symmetric flips change the accuracy of our method by at most 1%, and it remains the best in every setting, even under asymmetric flips that violate our assumed noise model.

5 Conclusion

In this paper, we studied dataset distillation under label noise and proposed a decontaminating framework that distills from label-corrupted data. By inverting the clean-class mixture behind each observed class, our method recovers unbiased estimates of the clean class prototypes and matches them in a single stage without label cleaning. We proved that the decontaminated matching loss is a consistent surrogate for clean-data distribution matching and upper-bounds the loss gap to models trained on clean data. Experiments on simulated and real-world label noise show that our method outperforms the baselines in most settings at a lower computational cost.

Reproducibility Statement

All assumptions are stated explicitly, and complete proofs are given in Appendices A.1–A.5. All experiments use publicly available datasets. Appendices D.2 and D.3 describe how the noisy-label benchmarks are constructed, and Appendix D.4 reports the distillation and evaluation recipes. An anonymized implementation is provided as supplementary material.

AI Use Statement

Generative AI tools were used to assist with language polishing. All research ideas, theoretical development, experimental design, and analyses are due to the authors, who checked and approved all AI-assisted text and take full responsibility for the manuscript.

Ethics Statement

This work involves no human subjects and no new data collection. All experiments use publicly available benchmarks, in accordance with their licenses. We declare no conflicts of interest.

References

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10708–10717, 2022.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021.

Lechao Cheng, Kaifeng Chen, Jiyang Li, Shengeng Tang, Shufei Zhang, and Meng Wang. Dataset distillers are good label denoisers in the wild. *arXiv preprint arXiv:2411.11924*, 2024.

Jiacheng Cui, Xinyue Bi, Yaxin Luo, Xiaohan Zhao, Jiacheng Liu, and Zhiqiang Shen. FADRM: Fast and accurate data residual matching for dataset distillation. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Priyam Dey, Aditya Sahdev, Sunny Bhati, Konda Reddy Mopuri, and Venkatesh Babu Radhakrishnan. Rethinking dataset distillation: Hard truths about soft labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 178–187, 2026.

Gerald B Folland. *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, 1999.

Saurabh Garg, Yifan Wu, Alexander J. Smola, Sivaraman Balakrishnan, and Zachary C. Lipton. Mixture proportion estimation and PU learning: A modern approach. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8532–8544, 2021.

Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15793–15803, 2024.

Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *International Conference on Learning Representations*, 2024.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, volume 31, pp. 8527–8537, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Youbing Hu, Yun Cheng, Olga Saukh, Firat Ozdemir, Anqi Lu, Zhiqiang Cao, Min Zhang, and Zhijun Li. FocusDC: Real-world scene infusion for robust dataset condensation. In *Proceedings of the Conference on Parsimony and Learning*, 2026.

Shantanu Jain, Martha White, Michael W. Trosset, and Predrag Radivojac. Nonparametric semi-supervised learning of class proportions. *arXiv preprint arXiv:1601.01944*, 2016.

Nicole Hee-Yeon Kim and Hwanjun Song. Robust dataset condensation using supervised contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2857–2866, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pp. 1106–1114, 2012.

Yann Le, Xuan Yang, et al. Tiny ImageNet visual recognition challenge. Technical Report 7, 2015.

Yongmin Lee and Hye Won Chung. SelMatch: Effectively scaling up dataset distillation via selection-based initialization and partial updates by trajectory matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 26546–26567, 2024.

Shiye Lei and Dacheng Tao. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32, 2024.

Junnan Li, Richard Socher, and Steven C. H. Hoi. DivideMix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

Muquan Li, Hang Gou, Dongyang Zhang, Shuang Liang, Xiurui Xie, Deqiang Ouyang, and Ke Qin. Beyond random: Automatic inner-loop optimization in dataset distillation. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yang Liu, Hao Cheng, and Kun Zhang. Identifiability of label noise transition matrix. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 21475–21496, 2023.

Wenshui Luo, Shuo Chen, Tongliang Liu, Bo Han, Gang Niu, Masashi Sugiyama, Dacheng Tao, and Chen Gong. Estimating per-class statistics for label noise learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(1):305–322, 2025.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, volume 26, pp. 1196–1204, 2013.

Li Niu, Qingtao Tang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Learning from noisy web data with category-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7689–7698, 2018.

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*, pp. 2233–2241, 2017.

Harish G. Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pp. 2052–2060, 2016.

Fengli Ran, Xiao Pu, Bo Liu, Xiuli Bi, and Bin Xiao. TGDD: Trajectory guided dataset distillation with balanced distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15662–15670, 2026.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of machine learning research*, 11(4), 2010.

Filipe Rodrigues and Francisco C. Pereira. Deep learning from crowds. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1611–1618, 2018.

Noveen Sachdeva and Julian J. McAuley. Data distillation: A survey. *Transactions on Machine Learning Research*, 2023.

Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pp. 838–846, 2015.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Xinyi Shang, Peng Sun, and Tao Lin. GIFT: Unlocking full potential of labels in distilled dataset at near-zero cost. In *International Conference on Learning Representations*, 2025.

Shitong Shao, Zikai Zhou, Huanran Chen, and Zhiqiang Shen. Elucidating the design space of dataset condensation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 99161–99201, 2024.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. D^4M: Dataset distillation via disentangled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5809–5818, 2024.

Duo Su, Huyu Wu, Huanran Chen, Yiming Shi, Yuzhu Wang, Xi Ye, and Jun Zhu. Diffusion models as dataset distillation priors. In *International Conference on Learning Representations*, volume 2026, pp. 26532–26552, 2026.

Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9390–9399, 2024.

Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N. Chiang, Zhihao Wu, and Xiaowei Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693, 2020.

Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25570–25580, 2025.

Shaobo Wang, Yantai Yang, Guo Chen, Peiru Li, Kaixin Li, Yufa Zhou, Zhaorun Chen, and Linfeng Zhang. Grounding and enhancing informativeness and utility in dataset distillation. In *International Conference on Learning Representations*, 2026.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022.

Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual T: Reducing estimation error for transition matrix in label-noise learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7260–7271, 2020.

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at ImageNet scale from a new perspective. In *Advances in Neural Information Processing Systems*, volume 36, pp. 73582–73603, 2023.

Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):150–170, 2024.

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

Bo Zhao and Hakan Bilen. Dataset condensation with differentiable Siamese augmentation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 12674–12685, 2021.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6503–6512, 2023.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021.

Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7856–7865, 2023.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1): 44–53, 2018.

APPENDIX CONTENTS

A Proofs **15**

A.1 Proof of Lemma 1 15

A.2 Proof of Proposition 1 15

A.3 Proof of Theorem 1 16

A.4 Proof of Theorem 2 17

A.5 Plug-in Estimation of the Mixture Parameters 19

B Related Work **21**

C Algorithm **21**

D Experimental Details **21**

D.1 Datasets and Baselines 21

D.2 Construction of the Noisy-Label Benchmarks 22

D.3 Noise Structures for Robustness Evaluation 22

D.4 Implementation Details 23

D.5 Additional Results 24

A PROOFS

A.1 PROOF OF LEMMA 1

Proof. Fix a class c. The assumption of Lemma 1 gives

$$p(\tilde{y}=c\,|\,\boldsymbol{x},y=c)=p(\tilde{y}=c\,|\,y=c),\qquad p(\tilde{y}=c\,|\,\boldsymbol{x},y\neq c)=p(\tilde{y}=c\,|\,y\neq c),$$

and, taking complements,

$$p(\tilde{y}\neq c\,|\,\boldsymbol{x},y=c)=p(\tilde{y}\neq c\,|\,y=c),\qquad p(\tilde{y}\neq c\,|\,\boldsymbol{x},y\neq c)=p(\tilde{y}\neq c\,|\,y\neq c).$$

By Bayes' rule and the first identity,

$$p(\boldsymbol{x}\,|\,\tilde{y}=c,y=c)=\frac{p(\boldsymbol{x}\,|\,y=c)\,p(\tilde{y}=c\,|\,\boldsymbol{x},y=c)}{p(\tilde{y}=c\,|\,y=c)}=p(\boldsymbol{x}\,|\,y=c).$$

Applying Bayes' rule with the remaining three identities in the same way, we have

$$p(\boldsymbol{x}\,|\,\tilde{y}=c,y\neq c)=\frac{p(\boldsymbol{x}\,|\,y\neq c)\,p(\tilde{y}=c\,|\,\boldsymbol{x},y\neq c)}{p(\tilde{y}=c\,|\,y\neq c)}=p(\boldsymbol{x}\,|\,y\neq c),$$

$$p(\boldsymbol{x}\,|\,\tilde{y}\neq c,y=c)=\frac{p(\boldsymbol{x}\,|\,y=c)\,p(\tilde{y}\neq c\,|\,\boldsymbol{x},y=c)}{p(\tilde{y}\neq c\,|\,y=c)}=p(\boldsymbol{x}\,|\,y=c),$$

$$p(\boldsymbol{x}\,|\,\tilde{y}\neq c,y\neq c)=\frac{p(\boldsymbol{x}\,|\,y\neq c)\,p(\tilde{y}\neq c\,|\,\boldsymbol{x},y\neq c)}{p(\tilde{y}\neq c\,|\,y\neq c)}=p(\boldsymbol{x}\,|\,y\neq c).$$

Decomposing each observed class-conditional density according to the true label and substituting the identities above, we obtain

$$\begin{aligned}p(\boldsymbol{x}\,|\,\tilde{y}=c)&=p(\boldsymbol{x},y=c\,|\,\tilde{y}=c)+p(\boldsymbol{x},y\neq c\,|\,\tilde{y}=c)\\&=p(y=c\,|\,\tilde{y}=c)\,p(\boldsymbol{x}\,|\,\tilde{y}=c,y=c)+p(y\neq c\,|\,\tilde{y}=c)\,p(\boldsymbol{x}\,|\,\tilde{y}=c,y\neq c)\\&=\alpha_c\,p(\boldsymbol{x}\,|\,y=c)+(1-\alpha_c)\,p(\boldsymbol{x}\,|\,y\neq c)\end{aligned}$$

and, likewise,

$$\begin{aligned}p(\boldsymbol{x}\,|\,\tilde{y}\neq c)&=p(\boldsymbol{x},y=c\,|\,\tilde{y}\neq c)+p(\boldsymbol{x},y\neq c\,|\,\tilde{y}\neq c)\\&=p(y=c\,|\,\tilde{y}\neq c)\,p(\boldsymbol{x}\,|\,\tilde{y}\neq c,y=c)+p(y\neq c\,|\,\tilde{y}\neq c)\,p(\boldsymbol{x}\,|\,\tilde{y}\neq c,y\neq c)\\&=(1-\beta_c)\,p(\boldsymbol{x}\,|\,y=c)+\beta_c\,p(\boldsymbol{x}\,|\,y\neq c),\end{aligned}$$

where $\alpha_c=p(y=c\,|\,\tilde{y}=c)$ and $\beta_c=p(y\neq c\,|\,\tilde{y}\neq c)$ by definition. These are exactly the two equations in Eq. (5), which concludes the proof. □

A.2 PROOF OF PROPOSITION 1

Proof. Writing Eq. (5) in matrix form gives

$$\begin{pmatrix}p(\boldsymbol{x}\,|\,\tilde{y}=c)\\p(\boldsymbol{x}\,|\,\tilde{y}\neq c)\end{pmatrix}=A_c\begin{pmatrix}p(\boldsymbol{x}\,|\,y=c)\\p(\boldsymbol{x}\,|\,y\neq c)\end{pmatrix},\qquad A_c:=\begin{pmatrix}\alpha_c & 1-\alpha_c\\1-\beta_c & \beta_c\end{pmatrix},$$

where $\det A_c=\alpha_c\beta_c-(1-\alpha_c)(1-\beta_c)=\alpha_c+\beta_c-1>0$ by assumption, so A_c is invertible with

$$A_c^{-1}=\frac{1}{\alpha_c+\beta_c-1}\begin{pmatrix}\beta_c & -(1-\alpha_c)\\-(1-\beta_c) & \alpha_c\end{pmatrix}.$$

Multiplying both sides by A_c^{-1}, the first row yields

$$p(\boldsymbol{x}\,|\,y=c)=\frac{\beta_c\,p(\boldsymbol{x}\,|\,\tilde{y}=c)-(1-\alpha_c)\,p(\boldsymbol{x}\,|\,\tilde{y}\neq c)}{\alpha_c+\beta_c-1},$$

which is Eq. (6). The second row gives the analogous expression for the complementary density, $p(\boldsymbol{x}\,|\,y\neq c)=\big[\alpha_c\,p(\boldsymbol{x}\,|\,\tilde{y}\neq c)-(1-\beta_c)\,p(\boldsymbol{x}\,|\,\tilde{y}=c)\big]/(\alpha_c+\beta_c-1)$, which underlies the negative prototype used in Section 3.4. □

A.3 Proof of Theorem 1

Proof. Fix an encoder ϑ, and let $\mu_c^{\mathcal{R}} := \mathbb{E}_{\boldsymbol{x}\sim p(\cdot|y=c)}[\psi_{\vartheta}(\boldsymbol{x})]$ and $\mu_{\neg c}^{\mathcal{R}} := \mathbb{E}_{\boldsymbol{x}\sim p(\cdot|y\neq c)}[\psi_{\vartheta}(\boldsymbol{x})]$.

Unbiasedness of the decontaminated estimator. Since the samples in $\tilde{\mathcal{R}}^c$ and $\tilde{\mathcal{R}}^{\neg c}$ are i.i.d. draws from $p(\boldsymbol{x} \mid \tilde{y} = c)$ and $p(\boldsymbol{x} \mid \tilde{y} \neq c)$, Lemma 1 gives

$$\mathbb{E}_{\tilde{\mathcal{R}}}\big[\tilde{\mu}_c^{+} \mid \vartheta\big] = \int \psi_{\vartheta}(\boldsymbol{x})\, p(\boldsymbol{x} \mid \tilde{y} = c)\, d\boldsymbol{x} = \alpha_c \mu_c^{\mathcal{R}} + (1-\alpha_c)\mu_{\neg c}^{\mathcal{R}},$$

$$\mathbb{E}_{\tilde{\mathcal{R}}}\big[\tilde{\mu}_c^{-} \mid \vartheta\big] = \int \psi_{\vartheta}(\boldsymbol{x})\, p(\boldsymbol{x} \mid \tilde{y} \neq c)\, d\boldsymbol{x} = (1-\beta_c)\mu_c^{\mathcal{R}} + \beta_c \mu_{\neg c}^{\mathcal{R}}.$$

Plugging into the definition of $\hat{\mu}_c$ in Eq. (8) and collecting coefficients:

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathcal{R}}}[\hat{\mu}_c \mid \vartheta] &= \frac{\beta_c\big[\alpha_c \mu_c^{\mathcal{R}} + (1-\alpha_c)\mu_{\neg c}^{\mathcal{R}}\big] - (1-\alpha_c)\big[(1-\beta_c)\mu_c^{\mathcal{R}} + \beta_c \mu_{\neg c}^{\mathcal{R}}\big]}{\alpha_c + \beta_c - 1} \\
&= \frac{\big[\alpha_c\beta_c - (1-\alpha_c)(1-\beta_c)\big]\mu_c^{\mathcal{R}} + \big[\beta_c(1-\alpha_c) - (1-\alpha_c)\beta_c\big]\mu_{\neg c}^{\mathcal{R}}}{\alpha_c + \beta_c - 1} \\
&= \frac{(\alpha_c + \beta_c - 1)\,\mu_c^{\mathcal{R}} + 0 \cdot \mu_{\neg c}^{\mathcal{R}}}{\alpha_c + \beta_c - 1} = \mu_c^{\mathcal{R}},
\end{aligned}$$

where we used $\alpha_c\beta_c - (1-\alpha_c)(1-\beta_c) = \alpha_c + \beta_c - 1 > 0$.

(a) Gradient consistency. Note that $\mu_c^{\mathcal{S}}(\vartheta) = \frac{1}{|\mathcal{S}^c|}\sum_{\boldsymbol{s}\in\mathcal{S}^c}\psi_{\vartheta}(\boldsymbol{s})$ depends only on $\mathcal{S}$ and ϑ, not on the random sample $\tilde{\mathcal{R}}$. Moreover, each summand $(\nabla_{\mathcal{S}}\mu_c^{\mathcal{S}})^{\top}(\mu_c^{\mathcal{S}} - \hat{\mu}_c)$ below is bounded uniformly in $(\vartheta, \tilde{\mathcal{R}})$, since $\|\psi_{\vartheta}(\boldsymbol{x})\|_2 \le B$ bounds $\mu_c^{\mathcal{S}} - \hat{\mu}_c$ and $\|\nabla_{\boldsymbol{x}}\psi_{\vartheta}(\boldsymbol{x})\|_{\mathrm{op}} \le L$ bounds $\nabla_{\mathcal{S}}\mu_c^{\mathcal{S}}$. Hence, by Fubini's theorem (Folland, 1999), we may differentiate under $\mathbb{E}_{\vartheta}$ and exchange $\mathbb{E}_{\tilde{\mathcal{R}}}$ with $\mathbb{E}_{\vartheta}$. Differentiating Eq. (9) term by term gives

$$\nabla_{\mathcal{S}}\,\mathcal{L}_{\mathrm{DeDD}} = 2\,\mathbb{E}_{\vartheta}\sum_{c=1}^{K}\big(\nabla_{\mathcal{S}}\mu_c^{\mathcal{S}}\big)^{\top}\big(\mu_c^{\mathcal{S}} - \hat{\mu}_c\big),$$

and taking $\mathbb{E}_{\tilde{\mathcal{R}}}$ on both sides yields

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathcal{R}}}[\nabla_{\mathcal{S}}\mathcal{L}_{\mathrm{DeDD}}] &= 2\,\mathbb{E}_{\vartheta}\sum_{c=1}^{K}\big(\nabla_{\mathcal{S}}\mu_c^{\mathcal{S}}\big)^{\top}\big(\mu_c^{\mathcal{S}} - \mathbb{E}_{\tilde{\mathcal{R}}}[\hat{\mu}_c \mid \vartheta]\big) \\
&= 2\,\mathbb{E}_{\vartheta}\sum_{c=1}^{K}\big(\nabla_{\mathcal{S}}\mu_c^{\mathcal{S}}\big)^{\top}\big(\mu_c^{\mathcal{S}} - \mu_c^{\mathcal{R}}\big) = \nabla_{\mathcal{S}}\mathcal{L}_{\mathrm{DM}},
\end{aligned}$$

where the second equality uses the unbiasedness established above.

(b) Expected-loss consistency. Fix ϑ and a class c. Conditional on ϑ, the synthetic mean $\mu_c^{\mathcal{S}}$ is deterministic, whereas $\hat{\mu}_c$ is random with mean $\mu_c^{\mathcal{R}}$ by the unbiasedness shown above. For a random vector Z with mean $\bar{Z}$ and a deterministic vector a, expanding the square gives $\mathbb{E}\|Z - a\|_2^2 = \|\bar{Z} - a\|_2^2 + \operatorname{tr}\operatorname{Cov}[Z]$, because the cross term $2\,\mathbb{E}\langle Z - \bar{Z}, \bar{Z} - a\rangle$ vanishes. Applying this with $Z = \hat{\mu}_c$ and $a = \mu_c^{\mathcal{S}}$ yields

$$\mathbb{E}_{\tilde{\mathcal{R}}}\big[\|\mu_c^{\mathcal{S}} - \hat{\mu}_c\|_2^2 \mid \vartheta\big] = \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2 + \operatorname{tr}\operatorname{Cov}[\hat{\mu}_c \mid \vartheta]. \tag{17}$$

Summing Eq. (17) over c and taking $\mathbb{E}_{\vartheta}$, which may be exchanged with $\mathbb{E}_{\tilde{\mathcal{R}}}$ by Fubini's theorem since all summands are bounded, the left-hand side becomes $\mathbb{E}_{\tilde{\mathcal{R}}}[\mathcal{L}_{\mathrm{DeDD}}]$ by the definition of $\mathcal{L}_{\mathrm{DeDD}}$ in Eq. (9), the first term on the right becomes $\mathbb{E}_{\vartheta}\sum_{c=1}^{K}\|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2 = \mathcal{L}_{\mathrm{DM}}$, and the second becomes $\sum_{c=1}^{K}\mathbb{E}_{\vartheta}[\operatorname{tr}\operatorname{Cov}[\hat{\mu}_c \mid \vartheta]] = V$. This proves $\mathbb{E}_{\tilde{\mathcal{R}}}[\mathcal{L}_{\mathrm{DeDD}}] = \mathcal{L}_{\mathrm{DM}} + V$.

It remains to bound V. Since $\tilde{\mu}_c^{+}$ and $\tilde{\mu}_c^{-}$ are computed on the independent partitions $\tilde{\mathcal{R}}^c$ and $\tilde{\mathcal{R}}^{\neg c}$, the definition of $\hat{\mu}_c$ in Eq. (8) gives

$$\operatorname{Cov}[\hat{\mu}_c \mid \vartheta] = \frac{\beta_c^2 \operatorname{Cov}[\tilde{\mu}_c^{+} \mid \vartheta] + (1-\alpha_c)^2 \operatorname{Cov}[\tilde{\mu}_c^{-} \mid \vartheta]}{(\alpha_c + \beta_c - 1)^2},$$

and since $\tilde{\mu}_c^+$ and $\tilde{\mu}_c^-$ are averages of i.i.d. vectors, $\mathrm{Cov}[\tilde{\mu}_c^+ \mid \boldsymbol{\vartheta}] = \frac{1}{|\tilde{\mathcal{R}}^c|}\mathrm{Cov}_{\boldsymbol{x}\sim p(\boldsymbol{x}\mid\tilde{y}=c)}[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})]$ and $\mathrm{Cov}[\tilde{\mu}_c^- \mid \boldsymbol{\vartheta}] = \frac{1}{|\tilde{\mathcal{R}}^{\neg c}|}\mathrm{Cov}_{\boldsymbol{x}\sim p(\boldsymbol{x}\mid\tilde{y}\neq c)}[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})]$. Finally, $\|\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|_2 \le B$ implies $\operatorname{tr}\mathrm{Cov}[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})] \le \mathbb{E}\|\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|_2^2 \le B^2$ under either distribution, so

$$\operatorname{tr}\mathrm{Cov}[\hat{\mu}_c \mid \boldsymbol{\vartheta}] \le \frac{B^2}{(\alpha_c+\beta_c-1)^2}\Big(\frac{\beta_c^2}{|\tilde{\mathcal{R}}^c|} + \frac{(1-\alpha_c)^2}{|\tilde{\mathcal{R}}^{\neg c}|}\Big) = O(1/|\tilde{\mathcal{R}}|),$$

uniformly in $\boldsymbol{\vartheta}$, where the rate uses that $|\tilde{\mathcal{R}}^c|$ and $|\tilde{\mathcal{R}}^{\neg c}|$ grow proportionally with $|\tilde{\mathcal{R}}|$. Hence $V = O(1/|\tilde{\mathcal{R}}|)$ and $V \to 0$ as $|\tilde{\mathcal{R}}| \to \infty$. □

A.4 Proof of Theorem 2

Let $e_c \in \{0,1\}^K$ denote the one-hot encoding of class c and I_m the $m \times m$ identity matrix. For $\mathcal{A} = \mathcal{R}$, $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}$ is the expectation under the clean distribution $p(\boldsymbol{x}, y)$, and for $\mathcal{A} = \mathcal{S}$ it is the average over the synthetic pairs $(\boldsymbol{s}_j, y'_j)$. Let $\pi_c^{\mathcal{R}} := p(y = c)$ and $\pi_c^{\mathcal{S}} := |\mathcal{S}^c|/|\mathcal{S}|$ denote the class proportions and $\mu_c^{\mathcal{A}} := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}) \mid y = c]$ the class means, so that $\mu_c^{\mathcal{R}}$ is the clean class mean of Eq. (7) and $\mu_c^{\mathcal{S}}$ the synthetic class mean. The ridge objective of Theorem 2 reads $J_{\mathcal{A}}(W) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\|W^\top\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}) - e_y\|_2^2 + \lambda\|W\|_{\mathrm{F}}^2$, with minimizer $W^{\mathcal{A}}$. Define the second-moment and cross-moment matrices

$$\Sigma_{\mathcal{A}} := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\big[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})^\top\big] \in \mathbb{R}^{m\times m}, \qquad N_{\mathcal{A}} := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\big[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\, e_y^\top\big] \in \mathbb{R}^{m\times K}. \quad (18)$$

Besides the conditions of Theorem 1, the proof uses the following two structural assumptions, which reflect standard dataset distillation practice and are stated here rather than in the main text to keep Theorem 2 readable. Remark 4 gives the corresponding bound when either of them is relaxed, so neither is essential to the result.

Assumption 1 (Matched Class Proportions)**.** $\mathcal{S}$ has the same class proportions as the clean distribution, i.e., $\pi_c^{\mathcal{S}} = \pi_c^{\mathcal{R}} =: \pi_c$ for all c. Let $\pi_{\max} = \max_c \pi_c$ denote the maximum class prior.

Assumption 2 (Matched Second Moments)**.** The second-moment matrix of the encoder features does not depend on the dataset, $\Sigma_{\mathcal{S}} = \Sigma_{\mathcal{R}} =: \Sigma_0$. For instance, $\Sigma_0 = \sigma^2 I_m$ when the encoder output is normalized to have fixed second moments.

Assumption 1 is easy to satisfy because the number of synthetic images per class is a design choice. The standard DD protocol allocates the same number to every class, which matches the class proportions of class-balanced clean data such as our benchmarks. Assumption 2 is a simplifying idealization that normalization layers in the encoder make reasonable, and Remark 4 gives the corresponding bound without it.

Lemma 2 (Closed Form of Linear Probe)**.** *For $\lambda > 0$, $J_{\mathcal{A}}$ has the unique minimizer*

$$W^{\mathcal{A}} := (\Sigma_{\mathcal{A}} + \lambda I_m)^{-1} N_{\mathcal{A}}, \qquad \textit{where} \qquad N_{\mathcal{A}} := \sum\nolimits_{c=1}^{K} \pi_c^{\mathcal{A}}\, \mu_c^{\mathcal{A}}\, e_c^\top. \quad (19)$$

Proof. First, we have

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\Big[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\, e_y^\top\Big] \\
=&\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\sum_{c=1}^{K}\Big[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\,\mathbf{1}\{y=c\}\, e_c^\top\Big] \\
=&\sum_{c=1}^{K}\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\big[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\,\mathbf{1}\{y=c\}\big]\, e_c^\top \\
=&\sum_{c=1}^{K}\pi_c^{\mathcal{A}}\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{A}}\big[\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}) \mid y=c\big]\, e_c^\top \\
=&\sum_{c=1}^{K}\pi_c^{\mathcal{A}}\,\mu_c^{\mathcal{A}}\, e_c^\top.
\end{aligned}$$

Using $\|W^\top\psi\|_2^2 = \mathrm{tr}(W^\top\psi\psi^\top W)$, $\langle W^\top\psi, e_y\rangle = \mathrm{tr}(W^\top\psi e_y^\top)$, $\|e_y\|_2^2 = 1$, and linearity of expectation (all moments are finite because $\|\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|_2 \le B$), the objective expands to

$$J_{\mathcal{A}}(W) = \mathrm{tr}\big(W^\top(\Sigma_{\mathcal{A}} + \lambda I_m)W\big) - 2\,\mathrm{tr}\big(W^\top N_{\mathcal{A}}\big) + 1.$$

Since $\Sigma_{\mathcal{A}} \succeq 0$, the matrix $\Sigma_{\mathcal{A}} + \lambda I_m$ is positive definite, so the quadratic part is strictly convex in W and $J_{\mathcal{A}}$ has a unique minimizer, characterized by the stationarity condition $\nabla_W J_{\mathcal{A}}(W) = 2(\Sigma_{\mathcal{A}} + \lambda I_m)W - 2N_{\mathcal{A}} = 0$. This gives the first equation in Eq. (19). □

Then, we give the proof of Theorem 2.

Proof of Theorem 2. Fix an encoder $\boldsymbol{\vartheta}$ for Steps 1 to 3, in which all quantities depend on it. Throughout, B is the feature bound of Theorem 1 and L_ℓ the Lipschitz constant of Theorem 2.

Step 1 (Pointwise loss deviation). For any $(\boldsymbol{x}, y)$, the Lipschitz condition on ℓ, the inequality $\|A^\top v\|_2 \le \|A\|_{\mathrm{op}}\|v\|_2 \le \|A\|_{\mathrm{F}}\|v\|_2$, and $\|\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|_2 \le B$ give

$$\big|\ell(\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y) - \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}(\boldsymbol{x}), y)\big| \le L_\ell\big\|(W^{\mathcal{S}} - W^{\mathcal{R}})^\top\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\big\|_2 \le L_\ell B\,\|W^{\mathcal{R}} - W^{\mathcal{S}}\|_{\mathrm{F}}. \tag{20}$$

The right-hand side does not depend on $(\boldsymbol{x}, y)$, so the same bound holds for the supremum over $(\boldsymbol{x}, y)$.

Step 2 (Reducing the head gap to a class-mean gap). By Assumption 2 the two probes share the same second-moment matrix Σ_0, so Lemma 2 gives $W^{\mathcal{A}} = Q_0^{-1}N_{\mathcal{A}}$ with the common matrix $Q_0 := \Sigma_0 + \lambda I_m$, and hence

$$W^{\mathcal{R}} - W^{\mathcal{S}} = Q_0^{-1}\big(N_{\mathcal{R}} - N_{\mathcal{S}}\big). \tag{21}$$

Step 3 (Norm bounds). (a) Since $\Sigma_0 \succeq 0$, we have $Q_0 \succeq \lambda I_m$ and hence $\|Q_0^{-1}\|_{\mathrm{op}} \le 1/\lambda$. (b) By Assumption 1 and Lemma 2, $N_{\mathcal{R}} - N_{\mathcal{S}} = \sum_c \pi_c\,(\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}})\,e_c^\top$, whose c-th column is $\pi_c(\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}})$, and therefore

$$\|N_{\mathcal{R}} - N_{\mathcal{S}}\|_{\mathrm{F}}^2 = \sum_{c=1}^K \pi_c^2\,\|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2 \le \pi_{\max}^2 \sum_{c=1}^K \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2.$$

Applying $\|XZ\|_{\mathrm{F}} \le \|X\|_{\mathrm{op}}\|Z\|_{\mathrm{F}}$ to Eq. (21) with (a) and (b) yields

$$\|W^{\mathcal{R}} - W^{\mathcal{S}}\|_{\mathrm{F}} \le \frac{\pi_{\max}}{\lambda}\sqrt{\sum_{c=1}^K \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2}, \tag{22}$$

and combining with Eq. (20) gives, for every $\boldsymbol{\vartheta}$,

$$\sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\big|\ell(\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y) - \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}(\boldsymbol{x}), y)\big| \le \frac{L_\ell B\,\pi_{\max}}{\lambda}\sqrt{\sum_{c=1}^K \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2}. \tag{23}$$

Step 4 (Integrating over the encoder). Both sides of Eq. (23) are bounded functions of $\boldsymbol{\vartheta}$ (since $\|\mu_c^{\mathcal{A}}\|_2 \le B$), so we may take $\mathbb{E}_{\boldsymbol{\vartheta}\sim P_{\boldsymbol{\vartheta}}}$ on both sides. By Jensen's inequality for the concave map $t \mapsto \sqrt{t}$ and the definition $\mathcal{L}_{\mathrm{DM}} = \mathbb{E}_{\boldsymbol{\vartheta}}\sum_{c=1}^K \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2$,

$$\mathbb{E}_{\boldsymbol{\vartheta}}\sqrt{\sum_{c=1}^K \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2} \le \sqrt{\mathbb{E}_{\boldsymbol{\vartheta}}\sum_{c=1}^K \|\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}}\|_2^2} = \sqrt{\mathcal{L}_{\mathrm{DM}}},$$

and therefore

$$\mathbb{E}_{\boldsymbol{\vartheta}}\Big[\sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}\big|\ell(\Phi_{\boldsymbol{\theta}^{\mathcal{S}}}(\boldsymbol{x}), y) - \ell(\Phi_{\boldsymbol{\theta}^{\mathcal{R}}}(\boldsymbol{x}), y)\big|\Big] \le \frac{L_\ell B\,\pi_{\max}}{\lambda}\sqrt{\mathcal{L}_{\mathrm{DM}}}. \tag{24}$$

Step 5 (From $\mathcal{L}_{\mathrm{DM}}$ to $\mathcal{L}_{\mathrm{DeDD}}$). By Theorem 1(b), $\mathbb{E}_{\tilde{\mathcal{R}}}[\mathcal{L}_{\mathrm{DeDD}}] = \mathcal{L}_{\mathrm{DM}} + V$ with $V \ge 0$, so $\mathcal{L}_{\mathrm{DM}} \le \mathbb{E}_{\tilde{\mathcal{R}}}[\mathcal{L}_{\mathrm{DeDD}}]$. Substituting this into Eq. (24) and using the monotonicity of $\sqrt{\cdot}$ yields Eq. (10). □

Finally, we provide the proof of Corollary 1.

Proof of Corollary 1. The supremum in Eq. (10) is a non-negative function of $\vartheta \sim P_\vartheta$ with expectation at most $\bar{\epsilon}$, so Markov's inequality gives $\Pr_\vartheta\left[\sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)} |\cdot| \le \bar{\epsilon}/\eta\right] \ge 1-\eta$, which is the statement of Definition 1 with $\epsilon = \bar{\epsilon}/\eta$ for the linear probes of Theorem 2. □

Remark 4 (On the assumptions). (1) *Matched second moments.* Without Assumption 2, the same argument goes through with the resolvent identity $Q_{\mathcal{R}}^{-1} - Q_{\mathcal{S}}^{-1} = Q_{\mathcal{R}}^{-1}(\Sigma_{\mathcal{S}} - \Sigma_{\mathcal{R}})Q_{\mathcal{S}}^{-1}$, where $Q_{\mathcal{A}} := \Sigma_{\mathcal{A}} + \lambda I_m$, and adds the term $L_\ell B^2\, \mathbb{E}_\vartheta \|\Sigma_{\mathcal{S}} - \Sigma_{\mathcal{R}}\|_{\mathrm{op}}/\lambda^2$ to the right-hand side of Eq. (10), using $\|N_{\mathcal{S}}\|_{\mathrm{F}} \le B$. This extra term does not depend on the labels, because $\Sigma_{\mathcal{R}}$ is determined by the input marginal $p(\boldsymbol{x})$, which is shared by $\mathcal{R}$ and $\tilde{\mathcal{R}}$. (2) *Balanced classes.* For balanced classes $\pi_{\max} = 1/K$, so $\bar{\epsilon} = L_\ell B\sqrt{\mathbb{E}_{\tilde{\mathcal{R}}}[\mathcal{L}_{\mathrm{DeDD}}]}/(K\lambda)$. (3) *Unequal class proportions.* If Assumption 1 is dropped, then $N_{\mathcal{R}} - N_{\mathcal{S}} = \sum_c \pi_c^{\mathcal{R}}(\mu_c^{\mathcal{R}} - \mu_c^{\mathcal{S}})e_c^\top + \sum_c(\pi_c^{\mathcal{R}} - \pi_c^{\mathcal{S}})\mu_c^{\mathcal{S}} e_c^\top$. The second sum has Frobenius norm at most $B\|\pi^{\mathcal{R}} - \pi^{\mathcal{S}}\|_2$, where $\pi^{\mathcal{A}} := (\pi_1^{\mathcal{A}}, \dots, \pi_K^{\mathcal{A}})$, since each $\mu_c^{\mathcal{S}}$ is an average of vectors of norm at most B, so the same argument adds the term $L_\ell B^2\|\pi^{\mathcal{R}} - \pi^{\mathcal{S}}\|_2/\lambda$ to the right-hand side of Eq. (10), with $\pi_{\max} := \max_c \pi_c^{\mathcal{R}}$. (4) *Finite-sample reference model.* Training on $\mathcal{R}$ is identified with the population objective $J_{\mathcal{R}}$, consistent with the population form of $\mathcal{L}_{\mathrm{DM}}$. Replacing $J_{\mathcal{R}}$ by its empirical version on the n clean samples can be handled by the same argument, at the cost of an additional term that vanishes as $n \to \infty$ and is independent of $\mathcal{S}$ and of the label noise.

A.5 Plug-in Estimation of the Mixture Parameters

Theorem 1 treats the mixture parameters (α_c, β_c) as known, whereas Algorithm 1 substitutes the MPE estimates $(\hat{\alpha}_c, \hat{\beta}_c)$ of Eq. (13). This subsection quantifies the perturbation caused by this substitution and shows that it is Lipschitz in the parameter error, so that the guarantees of Theorems 1 and 2 degrade gracefully and are recovered in the limit whenever the MPE estimator is consistent. The unbiasedness stated in Theorem 1 is therefore a property of the mixture inversion itself and carries over to the computed target up to an error proportional to the parameter error Δ_c. Figure 1 measures this error and finds it small on every dataset we consider.

For $a, b \in [0,1]$ with $a + b > 1$, write the decontaminated target of Eq. (8) as a function of the parameters,

$$\hat{\mu}_c(a,b) := \frac{b\,\tilde{\mu}_c^+ - (1-a)\,\tilde{\mu}_c^-}{a+b-1}, \tag{25}$$

so that $\hat{\mu}_c = \hat{\mu}_c(\alpha_c, \beta_c)$ is the target analyzed in Theorem 1, while $\hat{\mu}_c' := \hat{\mu}_c(\hat{\alpha}_c, \hat{\beta}_c)$ is the target actually computed by Algorithm 1. Let $\hat{\mathcal{L}}_{\mathrm{DeDD}}$ denote the *plug-in loss* obtained by replacing every $\hat{\mu}_c$ in Eq. (9) by $\hat{\mu}_c'$, and let

$$\Delta_c := |\hat{\alpha}_c - \alpha_c| + |\hat{\beta}_c - \beta_c|, \qquad \Delta := \sum\nolimits_{c=1}^{K} \Delta_c \tag{26}$$

denote the per-class and the total parameter error. Write $\Lambda_c := \alpha_c + \beta_c - 1$ and $\hat{\Lambda}_c := \hat{\alpha}_c + \hat{\beta}_c - 1$ for the determinant of the mixture matrix A_c of Proposition 1 at the true and at the estimated parameters. Throughout we assume a uniform margin

$$\min_c \min\{\Lambda_c,\ \hat{\Lambda}_c\} \ \ge\ \tau \ >\ 0, \tag{27}$$

which holds for the true parameters by the standing assumption of Theorem 1 and can be enforced for the estimates by clipping $\hat{\Lambda}_c$ at τ. Since $\Lambda_c \ge \tau$ and clipping is the projection onto $[\tau, \infty)$, this step never increases $|\hat{\Lambda}_c - \Lambda_c|$, so all bounds below remain valid for the clipped estimates. Note that $\tau \le 1$, because $a + b - 1 \le 1$ for $a, b \le 1$.

Proposition 2 (Plug-in stability)**.** *Under the conditions of Theorem 1 and Eq. (27), the following hold for every class c, every encoder ϑ, and every realization of $\tilde{\mathcal{R}}$:*

(a) ***(Prototype perturbation)*** $\left\|\hat{\mu}_c' - \hat{\mu}_c\right\|_2 \le 3B\Delta_c/\tau^2$.

(b) ***(Loss perturbation)*** $\left|\hat{\mathcal{L}}_{\mathrm{DeDD}} - \mathcal{L}_{\mathrm{DeDD}}\right| \le 18B^2\Delta/\tau^3$.

(c) ***(Gradient perturbation)*** *for every synthetic sample* $\boldsymbol{s}_j \in \mathcal{S}^c$, $\big\|\nabla_{\boldsymbol{s}_j}\hat{\mathcal{L}}_{\mathrm{DeDD}} - \nabla_{\boldsymbol{s}_j}\mathcal{L}_{\mathrm{DeDD}}\big\|_2 \le 6BL\Delta_c/(\tau^2|\mathcal{S}^c|)$.

Proof. Fix a class c and an encoder $\boldsymbol{\vartheta}$, and abbreviate the two numerators

$$g_c := \beta_c\,\tilde{\mu}_c^{+} - (1-\alpha_c)\,\tilde{\mu}_c^{-}, \qquad \hat{g}_c := \hat{\beta}_c\,\tilde{\mu}_c^{+} - (1-\hat{\alpha}_c)\,\tilde{\mu}_c^{-},$$

so that $\hat{\mu}_c = g_c/\Lambda_c$ and $\hat{\mu}'_c = \hat{g}_c/\hat{\Lambda}_c$ by Eq. (25). Because $\tilde{\mu}_c^{+}$ and $\tilde{\mu}_c^{-}$ are averages of feature vectors of norm at most B, we have $\|\tilde{\mu}_c^{\pm}\|_2 \le B$, whence

$$\|\hat{g}_c - g_c\|_2 = \big\|(\hat{\beta}_c-\beta_c)\,\tilde{\mu}_c^{+} + (\hat{\alpha}_c-\alpha_c)\,\tilde{\mu}_c^{-}\big\|_2 \le B\Delta_c, \qquad \|g_c\|_2 \le \big(\beta_c + 1 - \alpha_c\big)B \le 2B,$$

and $|\hat{\Lambda}_c - \Lambda_c| = \big|(\hat{\alpha}_c-\alpha_c) + (\hat{\beta}_c-\beta_c)\big| \le \Delta_c$.

(a) Decomposing the difference of the two quotients,

$$\hat{\mu}'_c - \hat{\mu}_c = \frac{\hat{g}_c}{\hat{\Lambda}_c} - \frac{g_c}{\Lambda_c} = \frac{\hat{g}_c - g_c}{\hat{\Lambda}_c} + \frac{g_c\,(\Lambda_c - \hat{\Lambda}_c)}{\Lambda_c\hat{\Lambda}_c},$$

and bounding each term with Eq. (27),

$$\big\|\hat{\mu}'_c - \hat{\mu}_c\big\|_2 \le \frac{B\Delta_c}{\tau} + \frac{2B\Delta_c}{\tau^2} = \frac{B\Delta_c(\tau+2)}{\tau^2} \le \frac{3B\Delta_c}{\tau^2},$$

where the last inequality uses $\tau \le 1$.

(b) From $\|g_c\|_2 \le 2B$, $\|\hat{g}_c\|_2 \le 2B$ and $\Lambda_c, \hat{\Lambda}_c \ge \tau$ we get $\|\hat{\mu}_c\|_2, \|\hat{\mu}'_c\|_2 \le 2B/\tau$, while $\|\mu_c^{\mathcal{S}}\|_2 \le B$ because $\mu_c^{\mathcal{S}}$ is an average of feature vectors. Applying the elementary bound $\big|\|r\|_2^2 - \|r'\|_2^2\big| \le \|r-r'\|_2\big(\|r\|_2 + \|r'\|_2\big)$ to the two residuals $r := \hat{\mu}'_c - \mu_c^{\mathcal{S}}$ and $r' := \hat{\mu}_c - \mu_c^{\mathcal{S}}$, whose difference is $r - r' = \hat{\mu}'_c - \hat{\mu}_c$, and using part (a) together with $\tau \le 1$,

$$\big|\,\|\hat{\mu}'_c - \mu_c^{\mathcal{S}}\|_2^2 - \|\hat{\mu}_c - \mu_c^{\mathcal{S}}\|_2^2\,\big| \le \frac{3B\Delta_c}{\tau^2}\Big(\frac{4B}{\tau} + 2B\Big) \le \frac{3B\Delta_c}{\tau^2}\cdot\frac{6B}{\tau} = \frac{18B^2\Delta_c}{\tau^3}.$$

Summing over c and taking $\mathbb{E}_{\boldsymbol{\vartheta}}$, which leaves the deterministic right-hand side unchanged, gives (b).

(c) Only the class-c term of Eq. (9) depends on $\boldsymbol{s}_j \in \mathcal{S}^c$, and since $\mu_c^{\mathcal{S}} = \frac{1}{|\mathcal{S}^c|}\sum_{\boldsymbol{s}\in\mathcal{S}^c}\psi_{\boldsymbol{\vartheta}}(\boldsymbol{s})$,

$$\nabla_{\boldsymbol{s}_j}\mathcal{L}_{\mathrm{DeDD}} = \frac{2}{|\mathcal{S}^c|}\,\mathbb{E}_{\boldsymbol{\vartheta}}\Big[\nabla_{\boldsymbol{x}}\psi_{\boldsymbol{\vartheta}}(\boldsymbol{s}_j)^{\top}\big(\mu_c^{\mathcal{S}} - \hat{\mu}_c\big)\Big].$$

Subtracting the same expression with $\hat{\mu}_c$ replaced by $\hat{\mu}'_c$, the synthetic means cancel, and $\|\nabla_{\boldsymbol{x}}\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|_{\mathrm{op}} \le L$ together with part (a) gives

$$\big\|\nabla_{\boldsymbol{s}_j}\hat{\mathcal{L}}_{\mathrm{DeDD}} - \nabla_{\boldsymbol{s}_j}\mathcal{L}_{\mathrm{DeDD}}\big\|_2 \le \frac{2L}{|\mathcal{S}^c|}\big\|\hat{\mu}'_c - \hat{\mu}_c\big\|_2 \le \frac{6BL\Delta_c}{\tau^2|\mathcal{S}^c|}. \qquad \square$$

Corollary 2 (Consistency of the plug-in objective)**.** *Assume the conditions of Theorem 2 and Eq. (27). Then Eq. (10) remains valid with* $\bar{\epsilon}$ *replaced by*

$$\hat{\epsilon} := \frac{L_\ell B\,\pi_{\max}}{\lambda}\left(\sqrt{\mathbb{E}_{\tilde{\mathcal{R}}}\big[\hat{\mathcal{L}}_{\mathrm{DeDD}}\big]} + \frac{3\sqrt{2}\,B}{\tau^{3/2}}\sqrt{\mathbb{E}_{\tilde{\mathcal{R}}}[\Delta]}\right). \tag{28}$$

If, in addition, the MPE estimator is consistent, i.e., $\Delta_c \to 0$ *in probability as* $|\tilde{\mathcal{R}}| \to \infty$ *for every* c*, then* $\hat{\epsilon} \to \bar{\epsilon}$*, and both the loss and the gradient of the plug-in objective converge to those of Theorem 1.*

Proof. By Proposition 2(b), $\mathcal{L}_{\mathrm{DeDD}} \le \hat{\mathcal{L}}_{\mathrm{DeDD}} + 18B^2\Delta/\tau^3$ for every realization of $\tilde{\mathcal{R}}$. Taking $\mathbb{E}_{\tilde{\mathcal{R}}}$ and using the subadditivity of $\sqrt{\cdot}$,

$$\sqrt{\mathbb{E}_{\tilde{\mathcal{R}}}\big[\mathcal{L}_{\mathrm{DeDD}}\big]} \le \sqrt{\mathbb{E}_{\tilde{\mathcal{R}}}\big[\hat{\mathcal{L}}_{\mathrm{DeDD}}\big]} + \frac{\sqrt{18}\,B}{\tau^{3/2}}\sqrt{\mathbb{E}_{\tilde{\mathcal{R}}}[\Delta]},$$

and substituting this into Eq. (10) yields Eq. (28) with $\sqrt{18} = 3\sqrt{2}$. For the second claim, the estimates of Eq. (13) satisfy $\hat{\alpha}_c, \hat{\beta}_c \in [0,1]$ whenever $\hat{\kappa}_c^{+}, \hat{\kappa}_c^{-} \in [0,1)$, so that $\Delta_c \le 2$ and $\Delta \le 2K$ is bounded. Convergence in probability of a bounded sequence implies convergence in expectation, so $\mathbb{E}_{\tilde{\mathcal{R}}}[\Delta] \to 0$ and the second term of Eq. (28) vanishes. The corresponding statements for the loss and the gradient follow from parts (b) and (c) of Proposition 2. $\square$

Remark 5. The perturbation caused by plug-in estimation is Lipschitz in the parameter error, with constants governed by the margin τ of Eq. (27). The guarantees therefore degrade gracefully rather than breaking down, and they are recovered exactly in the limit $\Delta \to 0$. Here τ acts as the factor by which the inversion amplifies the parameter error, which enters the three bounds multiplied by $1/\tau^2$, $1/\tau^3$, and $1/\tau^2$. A vanishing $\hat{\Lambda}_c$ can arise only from the estimates, since the true parameters satisfy $\Lambda_c > 0$ by the standing assumption of Theorem 1, and the clipping of Eq. (27) keeps this amplification bounded. Empirically, Figure 1 measures Δ_c directly and shows that it converges to a small value on all three datasets, including the real-world CIFAR-10N, so the residual term in Eq. (28) is small in practice. The target computed by Algorithm 1 is therefore close to the unbiased target of Theorem 1, and the two coincide as the estimates approach the true parameters. We state the guarantees for known (α_c, β_c) in the main text because they isolate what the mixture inversion contributes, and we quantify here the price of estimating these parameters, which is the only additional source of error introduced by the practical algorithm.

B Related Work

Dataset Distillation. Dataset distillation (DD) compresses a large training set into a small synthetic set that preserves its training utility (Wang et al., 2018). Existing methods differ in the learning signal they match: training gradients (Zhao et al., 2021; Zhao & Bilen, 2021), training trajectories (Cazenavette et al., 2022; Guo et al., 2024), or feature-space distributions (Zhao & Bilen, 2023; Zhao et al., 2023; Wang et al., 2025), with recent work adding decoupled objectives (Yin et al., 2023; Shao et al., 2024; Wang et al., 2026), data selection (Sun et al., 2024; Hu et al., 2026), generative priors (Su et al., 2024; 2026), and plug-and-play refinements (Shang et al., 2025; Cui et al., 2025). Almost all of them assume reliable labels, so their matching targets become biased once the labels are corrupted. Cheng et al. (2024) report that distillation suppresses random corruption to some extent but absorbs structured noise into the distilled samples, and Robust Dataset Condensation (RDC) (Kim & Song, 2025) combines supervised contrastive learning with a Golden MixUp (Zhang et al., 2018) strategy built on high-confidence clean or relabeled samples. DeDD neither identifies clean instances nor corrects labels: it treats each observed class-wise distribution as a contaminated mixture and matches the synthetic set against the recovered clean class statistics, tying distillation from label-corrupted data directly to its clean-data counterpart.

Learning with Noisy Labels. Noisy-label learning offers several complementary tools. One line models corruption as a class-dependent transition process and corrects the training objective with the estimated transition matrix (Natarajan et al., 2013; Patrini et al., 2017; Yao et al., 2020); another selects likely clean samples or trains cooperatively, as in Co-teaching (Han et al., 2018) and DivideMix (Li et al., 2020). Instance-dependent noise is harder and may not be identifiable without further assumptions (Liu et al., 2023). Closest to our formulation, mixture proportion estimation (MPE) recovers the proportion of a latent distribution inside an observed mixture without instance-level labels (Scott, 2015; Ramaswamy et al., 2016; Garg et al., 2021), consistently so under irreducibility conditions, and Per-Class Statistic Estimation (Luo et al., 2025) relates noisy and clean class statistics to train a classifier without sample selection. DeDD uses such estimation differently: it recovers only the class-wise feature statistics that distillation needs, so noisy-label learning acts inside the distillation objective rather than as a separate preprocessing or classifier-training stage.

C Algorithm

Algorithm 1 shows the algorithmic details of the proposed approach.

D Experimental Details

D.1 Datasets and Baselines

Datasets. Fashion-MNIST (Xiao et al., 2017) contains 60,000 training and 10,000 test grayscale images of resolution 28×28 over 10 clothing classes. CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) contain 50,000 training and 10,000 test images of resolution 32×32, spread over 10 and

Algorithm 1 DeDD

Input: Noisy-label dataset $\tilde{\mathcal{R}}$, Gaussian-initialized synthetic set $\mathcal{S}$, iterations T and coefficients λ_{m} and δ.
Output: Distilled synthetic dataset $\mathcal{S}$.

 # Phase 1: Parameter Estimation
for class $c = 1$ **to** K **do**
  Partition $\tilde{\mathcal{R}}$ into the positive set $\tilde{\mathcal{R}}^c$ and its complement $\tilde{\mathcal{R}}^{\neg c}$.
  Estimate mixture proportions $\hat{\kappa}_c^+, \hat{\kappa}_c^-$ via MPE.
  Recover $\hat{\alpha}_c, \hat{\beta}_c$ via Eq. (13).
end for
 # Phase 2: Decontaminating Dataset Distillation
for iteration $t = 1$ **to** T **do**
  Sample an encoder $\boldsymbol{\vartheta} \sim P_{\boldsymbol{\vartheta}}$.
  for class $c = 1$ **to** K **do**
    Sample batches from $\tilde{\mathcal{R}}^c$ and $\tilde{\mathcal{R}}^{\neg c}$.
    Compute the decontaminated prototypes $\hat{\mu}_c$ and $\hat{\mu}_{\neg c}$ via Eqs. (8) and (14) with $(\hat{\alpha}_c, \hat{\beta}_c)$.
    Compute the synthetic class mean $\mu_c^{\mathcal{S}}$ from $\mathcal{S}^c$.
  end for
  Update $\mathcal{S}$ by a gradient step on Eq. (16).
end for

100 classes, respectively. Tiny-ImageNet (Le et al., 2015) contains 100,000 training and 10,000 test images of resolution 64×64 over 200 classes. The six ImageNet subsets of Cazenavette et al. (2022), namely ImageNette, ImageWoof, ImageFruit, ImageMeow, ImageSquawk, and ImageYellow, each group 10 ImageNet (Deng et al., 2009) classes at resolution 128×128, with ImageNette collecting easily distinguishable classes and ImageWoof collecting dog breeds that are hard to separate. On all of these benchmarks the training labels are corrupted as described in Appendix D.2, while the test set is left clean. CIFAR-10N and CIFAR-100N (Wei et al., 2022) instead reuse the CIFAR images with labels collected from human annotators, so their noise is real rather than simulated. We use the *aggregate* label set for CIFAR-10N and the *noisy fine* label set for CIFAR-100N.

Baselines. All baselines are run with their official implementations and the hyperparameters reported for the clean-label setting. We exclude distillation methods that rely on external priors, such as pretrained teachers, generative priors, or teacher-generated soft labels (Yin et al., 2023; Gu et al., 2024; Dey et al., 2026), because their priors are learned from clean data and would confound the effect of label noise.

D.2 Construction of the Noisy-Label Benchmarks

For each standard benchmark we corrupt only the training labels, leaving the test set untouched. We group the clean training set by true label into K buckets, one per class, and then fill the set that will carry label c by drawing without replacement a fraction α_c of its samples from bucket c and the remaining $1 - \alpha_c$ from the other buckets, chosen uniformly at random. Every label set therefore keeps its original size. We refer to this as the *bucket procedure*. Since each label set draws its mislabeled samples uniformly from the other buckets, every other class is mislabeled as c at the same rate, so the assumption of Lemma 1 holds exactly for these benchmarks. The construction leaves a single free parameter per class, because each of the other $K - 1$ label sets draws a fraction $\frac{1-\alpha_c}{K-1}$ of its samples from bucket c, so that

$$1 - \beta_c = \Pr(y = c \mid \tilde{y} \neq c) = \frac{1 - \alpha_c}{K - 1}, \qquad \text{i.e.} \qquad \beta_c = 1 - \frac{1 - \alpha_c}{K - 1}. \tag{29}$$

Choosing α_c thus fixes β_c as well. Note that β_c approaches 1 as K grows, because the same amount of noise is spread over more competing classes. The large β_c values in Table 7 reflect this dilution rather than a low noise level.

D.3 Noise Structures for Robustness Evaluation

Table 6 evaluates four noise structures on the CIFAR-10 and CIFAR-100 datasets. As in the main experiments, only the training labels are corrupted and the test set is left clean.

Table 7: Resolution and noise parameters of the benchmarks with synthetic noise, where β_c follows from α_c and the class count K via Eq. (29).

Dataset	Resolution	$(\boldsymbol{\alpha_c}, \boldsymbol{\beta_c})$
Fashion-MNIST	28×28	(0.6, 0.956)
CIFAR-10	32×32	(0.6, 0.956)
CIFAR-100	32×32	(0.6, 0.996)
Tiny-ImageNet	64×64	(0.7, 0.998)
ImageNet Subsets	128×128	(0.6, 0.956)

Shared Clean Fraction. This is the noise structure of the main experiments. We apply the bucket procedure with the same clean fraction $\alpha_c = 0.6$ for every class, which gives 40% corruption.

Per-Class Clean Fractions. We apply the same bucket procedure but give each class its own clean fraction α_c, spread evenly around the value used above so that the corruption rate varies across classes while averaging 40%. Since the label set of each class c' draws its mislabeled fraction $1 - \alpha_{c'}$ uniformly from the other $K - 1$ buckets, Eq. (29) generalizes to

$$1 - \beta_c = \frac{1}{(K-1)^2} \sum_{c' \neq c} \big(1 - \alpha_{c'}\big),$$

which varies across classes and reduces to Eq. (29) when all α_c are equal.

Symmetric Flips. Each training label is flipped independently with probability 0.4 to a class drawn uniformly from the other $K - 1$ classes. This corresponds to $p(\tilde{y} = c \,|\, y = c) = 0.6$ and $p(\tilde{y} = c' \,|\, y = c) = 0.4/(K-1)$ for $c' \neq c$, and gives 40% overall corruption.

Asymmetric Flips. Each eligible training label is flipped independently with probability 0.4 to a fixed class-specific target. For CIFAR-10, we follow Patrini et al. (2017) and use the mappings truck→automobile, bird→airplane, deer→horse, and cat↔dog. Only these five source classes are eligible, so the overall corruption is about $0.4 \times 5/10 = 20\%$. For CIFAR-100, every class c is mapped to $(c+1) \bmod 100$, so all classes are eligible and the overall corruption is again 40%. Unlike the other three structures, the mislabeled samples in each observed partition then come from a single source class instead of evenly from all other classes, so the equality $p(\tilde{y} = c \,|\, \boldsymbol{x}, y \neq c) = p(\tilde{y} = c \,|\, y \neq c)$ assumed in Lemma 1 fails, although the noise is still instance-independent. This setting therefore evaluates our method under a misspecified model.

D.4 Implementation Details

Distillation. We use a 3-layer ConvNet (ConvNet-3) with instance normalization as the random encoder $\psi_{\boldsymbol{\vartheta}}$. Synthetic images are initialized from $\mathcal{N}(0, 1)$ noise and optimized with SGD at momentum 0.5 for $20{,}000$ iterations, with learning rate 10.0 at IPC 50 and on the ImageNet subsets and 0.2 otherwise. At every iteration, we sample a fresh encoder $\psi_{\boldsymbol{\vartheta}}$ and, for each class c, draw 256 real images from each of the two observed partitions $\tilde{\mathcal{R}}^c$ and $\tilde{\mathcal{R}}^{\neg c}$. Real and synthetic images pass through the same differentiable augmentation (Zhao & Bilen, 2021) (color, crop, cutout, flip, scale, rotate) before encoding, and each synthetic image is split into $\rho^2 = 4$ sub-images by multi-formation (Zhao et al., 2023). The prototype margin loss uses $\lambda_{\mathrm{m}} = 0.1$ and $\delta = 1.0$.

Evaluation. Following Zhao & Bilen (2023), we train a ConvNet-3 with instance normalization from scratch on the distilled set for $1{,}000$ epochs with SGD at learning rate 0.01, momentum 0.9, and weight decay 5×10^{-4}, using batch size 256 and dividing the learning rate by 10 after 500 epochs. We apply the same differentiable augmentation as in distillation and use this recipe in all settings.

Mixture Proportion Estimation with TEDN. We solve the two MPE instances in Eqs. (11) and (12) with TEDN (Garg et al., 2021). Given samples from a component $\mathcal{P}$ and a mixture $\mathcal{U}$, TEDN trains a classifier to separate $\mathcal{P}$ from $\mathcal{U}$ and alternates two steps. The *estimate* step applies

Table 8: Clean-test accuracy (%) as mean ± std over 3 seeds. One-stage methods distill directly from the label-corrupted training set, whereas two-stage methods (†) first relabel it with DivideMix and then distill. In each column, the best result is in **bold** and the second best is underlined.

	Fashion-MNIST		CIFAR-10		CIFAR-100		Tiny-ImageNet		CIFAR-10N		CIFAR-100N	
IPC	**10**	**50**	**10**	**50**	**10**	**50**	**1**	**10**	**10**	**50**	**10**	**50**
Ratio (%)	0.17	0.83	0.2	1	2	10	0.2	2	0.2	1	2	10
Random	54.75±0.42	62.66±0.36	23.26±0.23	32.05±0.27	9.52±0.23	18.51±0.32	1.14±0.01	4.20±0.11	24.04±0.24	37.62±0.42	13.99±0.27	24.95±0.19
Herding	63.28±0.54	69.47±0.90	25.70±0.87	34.65±0.10	12.43±0.16	19.43±0.13	2.21±0.06	4.54±0.11	27.35±0.28	39.67±0.27	16.53±0.17	26.55±0.26
K-Center	29.75±1.39	24.56±0.41	12.90±0.42	13.64±0.35	2.31±0.07	2.86±0.09	1.08±0.09	1.17±0.05	13.68±0.38	16.77±0.27	5.02±0.04	9.94±0.33
DC	70.91±0.87	54.90±1.34	47.07±0.32	45.34±0.26	23.58±0.10	20.58±0.05	4.04±0.10	10.65±0.33	48.02±0.69	47.18±0.61	23.97±0.24	23.25±0.25
DSA	79.92±0.19	77.87±0.56	46.14±0.43	48.35±0.77	27.43±0.38	25.83±0.47	4.13±0.04	11.66±0.23	46.29±0.33	49.89±0.16	27.64±0.08	29.83±0.30
DM	81.96±0.02	82.29±0.24	43.99±0.36	43.96±0.38	23.93±0.25	22.44±0.12	3.63±0.09	9.49±0.12	44.88±0.80	43.59±0.67	24.57±0.30	22.72±0.15
IDM	81.37±0.14	85.44±0.26	44.44±0.35	56.63±0.56	23.61±0.32	34.80±0.25	4.16±0.07	11.35±0.06	45.03±0.63	58.03±0.44	25.09±0.28	34.61±0.42
MTT	79.39±0.19	78.51±0.38	54.66±1.21	58.06±0.23	29.76±0.08	28.28±0.15	1.56±0.03	5.15±0.02	53.51±1.37	60.17±0.93	29.30±0.25	29.60±0.32
DATM	83.24±0.08	**87.05±0.14**	46.95±0.88	59.20±0.37	29.41±0.18	33.91±0.23	4.17±0.08	8.84±0.13	49.40±0.62	60.66±0.20	27.59±0.11	36.06±0.14
TGDD	83.04±0.25	84.83±0.31	42.43±0.35	58.10±0.78	20.79±0.45	35.50±0.37	2.73±0.05	9.34±0.03	42.63±0.50	58.25±0.34	23.22±0.30	34.62±0.13
AT-BPTT	81.24±1.14	82.45±0.14	43.35±0.63	55.18±0.13	21.54±0.18	32.15±0.12	1.67±0.05	8.42±0.24	44.15±0.72	46.52±0.72	24.54±0.53	31.35±0.91
NCFM	82.24±2.04	79.96±2.22	50.29±0.62	57.04±0.57	27.66±0.04	36.12±0.11	2.84±0.11	10.14±0.07	42.12±0.33	48.83±0.31	23.43±0.16	27.37±0.24
DC†	78.65±0.17	79.30±0.21	50.18±0.32	56.27±0.16	27.76±0.02	29.78±0.17	4.50±0.18	13.29±0.29	50.16±0.32	56.18±0.26	26.72±0.40	28.61±0.36
DSA†	79.78±0.06	81.29±0.03	50.87±0.72	58.68±0.17	31.59±0.53	37.64±0.07	4.51±0.07	13.82±0.38	51.40±0.14	59.18±0.40	30.33±0.22	36.02±0.31
DM†	78.02±0.06	78.64±0.11	46.72±0.22	46.83±0.07	28.21±0.18	25.17±0.38	3.94±0.11	11.13±0.12	48.18±0.69	45.80±0.12	26.84±0.32	24.44±0.05
IDM†	79.20±0.10	81.73±0.22	47.13±0.48	60.71±0.68	28.67±0.14	39.73±0.07	4.64±0.24	13.92±0.30	46.41±0.60	61.23±0.39	26.67±0.03	36.23±0.06
MTT†	81.54±0.31	81.15±0.18	57.30±0.61	65.47±0.37	33.18±0.21	37.53±0.09	1.82±0.07	11.89±0.46	54.93±0.86	66.16±0.44	30.76±0.15	36.52±0.17
DATM†	80.65±0.26	80.67±0.16	56.75±0.50	65.01±0.15	28.55±0.18	40.37±0.27	4.68±0.14	9.25±0.15	52.15±0.39	**67.65±0.36**	27.42±0.18	37.89±0.20
TGDD†	78.50±0.10	80.34±0.08	46.75±0.27	61.06±0.32	29.00±0.11	40.94±0.49	3.41±0.07	12.83±0.12	46.86±0.19	60.52±0.27	27.25±0.06	36.13±0.06
AT-BPTT†	76.54±0.45	78.64±0.26	44.06±0.81	58.32±0.51	22.21±0.17	29.45±0.12	2.27±0.08	9.02±0.05	46.24±0.82	49.62±0.62	27.52±0.81	32.54±0.19
NCFM†	77.52±1.09	84.10±1.93	54.48±0.17	60.20±0.37	28.79±0.24	38.60±0.22	4.24±0.24	12.24±0.30	49.36±0.48	60.19±0.27	27.38±0.17	34.37±0.17
RDC	77.67±3.35	80.72±1.93	48.99±0.66	59.86±0.72	31.54±0.06	37.01±0.12	4.03±0.11	13.53±0.13	48.96±0.12	60.38±0.32	27.10±0.27	35.54±0.22
Ours	**84.01±0.17**	85.78±0.13	**57.86±0.42**	**65.70±0.22**	**34.64±0.33**	**41.98±0.43**	**5.63±0.21**	**14.84±0.34**	**55.81±0.12**	66.83±0.25	**31.89±0.28**	**38.44±0.17**

Best Bin Estimation (BBE) on a held-out split: for every score threshold, the fraction of $\mathcal{U}$ samples above it divided by the fraction of $\mathcal{P}$ samples above it upper-bounds κ, and the smallest such ratio, with a confidence correction, gives the estimate of κ. The *discard* step retrains the classifier with the Conditional Value Ignoring Risk (CVIR) objective, which ignores the estimated fraction of $\mathcal{U}$ samples most likely to come from $\mathcal{P}$ and treats the rest as drawn from $\mathcal{N}$. Alternating the two steps sharpens the classifier and hence the estimate. For each class c, we obtain $\hat{\kappa}_c^+$ with $\tilde{\mathcal{R}}^c$ as the mixture and $\tilde{\mathcal{R}}^{\neg c}$ as the component, and $\hat{\kappa}_c^-$ with the roles swapped.

In our implementation, all $2K$ instances of a dataset share a single ResNet-50 backbone with a K-way one-vs-rest head, where the c-th output separates $\tilde{\mathcal{R}}^{\neg c}$ from $\tilde{\mathcal{R}}^c$ and thus serves both instances of class c. The network is trained on the noisy-label training set with SGD at learning rate 10^{-2}, momentum 0.9, weight decay 5×10^{-4}, and batch size 256, for 2 warm-start epochs followed by 10 refinement epochs of alternating estimate and discard steps. We hold out 20% of the data as a validation split on which BBE is computed, and convert the resulting $\hat{\kappa}_c^+$ and $\hat{\kappa}_c^-$ into $\hat{\alpha}_c$ and $\hat{\beta}_c$ via Eq. (13). This estimation runs once before distillation and typically finishes within a few minutes, which is negligible compared with the cost of the distillation process.

D.5 Additional Results

This subsection collects the results that the main text abbreviates for space.

Table 8 extends Table 1 with two Fashion-MNIST columns. Coreset methods stay far behind, relabeling with DivideMix lifts most distillation baselines at the cost of a second stage, and our method is the best one-stage method in every column except Fashion-MNIST at IPC 50, where DATM leads by about one point. Counting all methods, it ranks first or second in every column, trailing only DATM there and the two-stage DATM† on CIFAR-10N at IPC 50.

Table 9 extends Table 2 with two ImageWoof columns and two more coreset baselines. Our method ranks first in all twelve settings of the six subsets. ImageWoof, whose ten classes are similar dog breeds, is among the hardest subsets, and our margins there are among the smallest.

Table 10 complements Table 4 with cross-architecture results on ImageNette at IPC 10. Data distilled with ConvNet-3 by our method again transfer best to all three unseen architectures, with the largest margin on AlexNet (54.27% vs. 31.40% for the strongest baseline).

Table 9: Clean-test accuracy (%) on six ImageNet subsets at 128×128. All methods are one-stage without relabeling. In each column, the best result is in **bold** and the second best is underlined.

	ImageNette		ImageWoof		ImageFruit		ImageMeow		ImageSquawk		ImageYellow	
IPC	**1**	**10**	**1**	**10**	**1**	**10**	**1**	**10**	**1**	**10**	**1**	**10**
Ratio (%)	0.105	1.05	0.11	1.1	0.077	0.77	0.077	0.77	0.077	0.77	0.077	0.77
Random	7.87±0.50	21.27±2.02	10.33±1.05	13.40±0.82	8.80±0.71	11.13±1.20	9.53±0.19	12.53±0.57	10.07±0.25	14.33±1.09	10.67±0.25	21.47±1.36
Herding	13.93±0.09	23.27±1.09	12.27±0.74	15.07±0.34	10.87±1.52	12.73±0.52	11.40±0.65	13.47±0.90	13.60±0.91	20.07±1.59	15.93±1.81	22.67±1.11
K-Center	11.07±0.66	10.27±1.09	10.33±0.41	9.13±0.52	12.13±0.77	10.60±0.49	10.20±1.47	7.60±0.16	10.00±0.71	13.93±0.94	11.47±0.74	10.20±0.59
DC	20.00±1.18	24.00±1.93	15.93±0.47	15.20±1.28	12.60±0.71	13.93±1.37	14.73±0.38	12.07±0.66	23.07±0.19	21.07±3.49	24.47±1.52	24.87±1.41
DSA	21.73±1.91	18.33±0.96	16.27±1.61	14.53±1.04	13.93±1.55	12.80±1.02	12.80±0.57	12.80±1.23	21.20±1.28	18.60±3.10	23.40±0.98	18.60±2.08
DM	20.27±1.32	26.13±1.86	16.20±0.71	19.07±1.91	14.00±0.43	13.73±0.98	13.93±1.37	12.93±2.46	19.27±1.23	21.73±2.25	22.27±1.31	31.33±2.22
IDM	16.47±0.09	30.07±3.09	13.07±0.66	14.27±1.41	11.00±1.73	17.27±0.90	12.67±0.50	16.60±2.29	14.13±0.77	20.33±1.91	14.87±1.11	31.40±1.42
MTT	14.07±0.90	15.67±0.84	14.93±0.74	13.07±1.23	9.87±0.96	11.00±0.28	10.80±0.59	11.47±0.41	14.07±1.46	14.13±0.68	12.40±1.23	16.33±0.82
DATM	14.13±0.66	22.53±1.79	11.33±1.36	14.73±0.25	9.73±0.66	16.00±1.73	9.73±1.00	15.73±0.41	10.20±1.07	20.53±1.18	14.93±1.73	26.27±0.47
TGDD	22.67±1.91	29.60±1.85	15.60±0.16	18.00±1.82	13.67±0.84	16.87±2.74	11.33±1.23	13.73±0.47	20.53±2.03	30.00±1.30	21.00±0.98	28.80±2.21
AT-BPTT	23.40±1.41	19.07±1.43	18.27±1.20	14.27±2.51	16.87±1.06	17.20±0.33	14.93±1.16	14.47±1.33	22.87±1.27	18.13±1.48	25.40±0.57	32.53±2.53
NCFM	20.00±0.75	22.07±0.98	19.73±0.66	20.53±0.98	18.27±1.00	19.27±0.41	9.87±0.96	12.33±0.81	20.80±0.65	29.13±0.47	28.67±0.19	37.87±0.41
RDC	20.25±0.15	33.15±1.19	15.13±1.18	17.24±0.73	15.52±0.63	16.46±0.93	9.67±0.57	14.13±0.82	24.49±0.34	32.56±0.59	28.73±0.28	40.73±0.49
Ours	**32.73**±1.79	**54.00**±2.99	**21.33**±1.11	**27.20**±0.33	**20.13**±2.23	**28.40**±1.48	**15.80**±1.03	**27.73**±2.45	**31.00**±1.23	**50.67**±2.04	**35.87**±0.25	**50.00**±2.16

Table 10: Cross-architecture accuracy on ImageNette at IPC 10.

	ConvNet-3	**AlexNet**	**VGG-11**	**ResNet-18**
DC	24.00±1.93	17.33±3.19	34.13±0.90	22.07±0.77
DSA	18.33±0.96	24.53±1.67	31.87±0.90	22.33±2.19
DM	26.13±1.86	31.40±1.57	40.80±0.75	26.20±0.86
IDM	30.07±3.09	26.80±1.73	43.53±1.18	26.60±2.47
MTT	15.67±0.84	17.20±1.30	18.73±0.93	21.40±1.30
DATM	22.53±1.79	26.60±0.71	35.40±1.56	36.13±1.27
TGDD	29.60±1.85	23.27±3.51	44.93±1.05	36.00±0.59
Ours	**54.00**±2.99	**54.27**±1.09	**60.67**±0.74	**50.40**±0.91