DELTA: Robustly Training Label-Conditional Diffusion ModELs with Weak AnnoTAtions

Dong-Dong Wu[1,2], **Jiacheng Cui**[3], **Wei Wang**[2],
Zhiqiang Shen[3], **Masashi Sugiyama**[2,1]
[1]The University of Tokyo
[2]RIKEN Center for Advanced Intelligence Project
[3]Mohamed bin Zayed University of Artificial Intelligence

Abstract

While label-conditional diffusion models have recently exhibited remarkable generative capabilities, their success heavily relies on massive, cleanly labeled datasets. In practice, categorical supervision is rarely perfect: it is often corrupted by noise, clouded by ambiguity, or partially missing. Training directly on such weak annotations severely degrades generation quality, and existing robust methods offer only fragmented solutions that depend on scarce auxiliary priors. To overcome these limitations, we propose *DELTA*, a unified framework that robustly trains Diffusion modELs across all three weak annoTAtion types. By treating the unknown true label as a latent variable, *DELTA* optimizes a principled variational objective that jointly recovers the clean data distribution and infers the true label posteriors. To make this joint training computationally tractable, we further introduce a median-centered timestep sampling strategy that efficiently concentrates estimation where the diffusion process is most informative. Extensive experiments on synthetic and real-world benchmarks demonstrate that *DELTA* produces high-fidelity, class-consistent samples across diverse weak supervision scenarios, outperforming specialized baselines while demanding strictly less prior knowledge.

1 Introduction

Diffusion models (DMs) [1–4] have become the de facto framework for high-fidelity image generation across diverse visual domains [5–7]. In practice, such generation is usually paired with a target concept, and the choice of conditioning signal shapes what the model can produce. Two of the most prominent paradigms are *text-conditional* diffusion [8, 9], which targets open-ended generation from natural-language prompts, and *label-conditional* diffusion models (LCDMs) [10–12], which target class-conditional distributions over discrete labels. Which paradigm applies depends on the form of supervision available in the target domain.

It is often assumed that natural language can express richer and more fine-grained descriptions than discrete class labels, leading the community's attention toward text-conditional generation [13, 14, 9, 15]. Yet *outside natural-language interaction settings*, the supervision routinely comes in categorical form: web-scale media is overwhelmingly labeled by tags rather than captions [16]; medical imaging is annotated with diagnostic codes such as lesion types, rather than free-text descriptions [17, 18]; and remote-sensing imagery is supervised by land-cover or defect-category metadata [19, 20]. *Label-conditional diffusion remains uniquely suited to these settings*: it conditions on categorical supervision and can synthesize samples for any target class, helping to mitigate practical data challenges such as costly real-world acquisition, limited distribution coverage, and scarce per-class samples.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

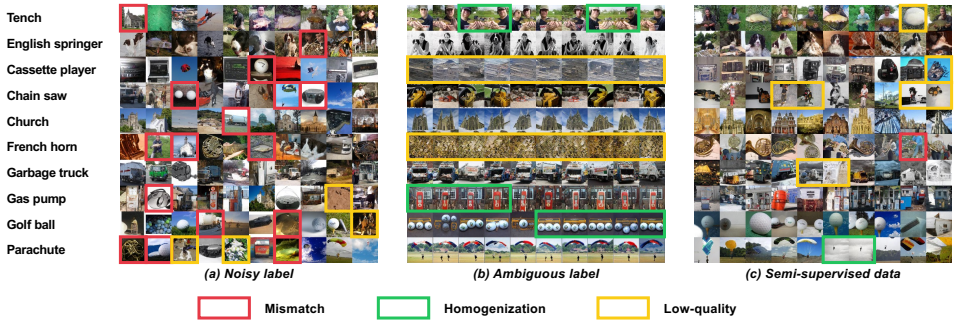


Figure 1: Randomly generated ImageNette samples from naive LCDMs trained under three forms of weak annotations: (a) noisy labels, (b) ambiguous labels, and (c) semi-supervised data. Colored boxes highlight three distinct failure modes that arise across these settings: class–image *mismatch*; per-class *homogenization*; and *low-quality* generations.

Yet in such LCDM scenarios, the real-world data used for training rarely comes with clean labels. In reality, such label supervision commonly manifests in three forms: labels are *noisy* when accumulated from heterogeneous sources such as user tags or automated taggers [21]; *ambiguous* when the true class is hard to single out, such as among similar lesion types in medical imaging [22, 23]; or simply *missing* when much of the collected data carries no associated label at all [24]. We refer to these three forms of imperfect supervision collectively as *weak annotations* [25]. Training an LCDM directly on such imperfect supervision is fundamentally hazardous: as Figure 1 illustrates, the generated samples consistently suffer from class–image *mismatch*, per-class *homogenization*, and degraded overall *quality*. To address these challenges, recent studies have proposed adaptations of diffusion models, such as noise-robust [12, 26–28] and semi-supervised [29] variants. However, these approaches each target a single weak form and further rely on task-specific external supervision, such as pre-estimated noise transition matrices [12] or risk confidence scores [26], which are seldom available in practice and cannot be straightforwardly combined to cover all three weak forms jointly. What is missing is a single LCDM training framework that handles all forms of weak annotations without invoking such auxiliary priors or committing to any single weak-label assumption.

In this paper, we propose *DELTA*, a unified framework for robustly training label-conditional Diffusion modELs under all three forms of weak annoTAtions. Our key idea is that the joint likelihood of data and weak annotations admits a variational lower bound (Section 4.1) whose maximization naturally yields an Expectation-Maximization (EM)-style training procedure. The **E-step** estimates per-sample true-label posteriors with the model fixed, and the **M-step** updates the model using two synergistic loss terms: a *Generative Term* for posterior-weighted score matching (Section 4.2) and a *Classification Term* for refining the model's class-posterior estimates (Section 4.3). Because both the E-step and the M-step involve expensive per-sample posterior computations, we further propose an efficient timestep sampling strategy (Section 4.4) that substantially reduces this overhead without sacrificing accuracy. Extensive experiments on image generation and classification validate that *DELTA* is robust across all three weak forms, outperforms specialized baselines that rely on auxiliary priors, and yields posteriors that rival dedicated classifiers.

Our contributions can be summarized as follows:

- We propose *DELTA*, a unified LCDM training framework that handles noisy, ambiguous, and missing labels within a single latent-variable objective, requiring less prior knowledge.
- We derive a median-centered timestep sampling strategy, supported by both theoretical and empirical analyses, that reduces posterior estimation cost by $27.5\times$ with no accuracy loss.
- Extensive experiments validate that *DELTA* outperforms specialized baselines that rely on auxiliary supervision, with analyses confirming posterior quality and sampling efficiency.

2 Related Work

Robust Training of LCDMs Standard LCDMs [1, 10] are trained with clean labels, yet real-world supervision is often noisy, ambiguous, or missing, prompting a recent line of work on robust LCDM

training under such weak annotations [12, 26–28, 30, 29, 31]. For noisy labels, prior work either explicitly models the data distribution via noise transition matrices [12] or risk-parameterized Stochastic Differential Equations (SDEs) [26], or repairs the conditioning signal through trainable pseudo-conditions [28], confidence-based reweighting [27, 32], and post-hoc discriminator correction [30]. A separate line addresses missing labels via consistency regularization [29] or positive-unlabeled formulations [31]. Despite these efforts, each handles only a *single* weak form and relies on task-specific external supervision (e.g., transition matrices and confidence scores) rarely available in practice. *DELTA* closes this gap by treating clean labels as latent variables, yielding a single EM-style LCDM training procedure that handles all three weak forms without such priors or auxiliary networks.

Diffusion Classifiers A pretrained diffusion model can itself act as a zero-shot classifier by inferring class likelihoods via Bayes' rule on its denoising-based evidence lower bound (ELBO) [33, 34]. This paradigm has since been refined for adversarial robustness via likelihood maximization [35], certified robustness via randomized smoothing [36], matched-noise selection for sharper logits [37], and computational efficiency [38]; recent work has also probed its compositional understanding [39]. However, all of these methods deploy the diffusion classifier exclusively at *inference time* on a frozen, pretrained backbone, where the expensive per-class likelihood evaluation across many timesteps is amortized over a single classification call. *DELTA* is, to our knowledge, the first to embed the diffusion classifier *into training itself*, using its posterior to guide updates of the same model. This is made practical by our proposed median-centered timestep sampling strategy.

3 Background

Diffusion Models. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the d-dimensional input space. Given a clean sample $\mathbf{x}$ from the real data distribution with density $q(\mathbf{x})$, the forward diffusion process injects Gaussian noise to generate a sequence of latent variables $\{\mathbf{x}_t\}_{t=1}^T$. This transition is defined by the conditional Gaussian density $\mathcal{N}$:

$$q(\mathbf{x}_t \,|\, \mathbf{x}) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \tag{1}$$

where $\alpha_t > 0$ and $\sigma_t > 0$ are the signal scaling and noise standard deviation schedules, respectively, and $\mathbf{I}$ represents the identity matrix. The reverse generative process aims to invert this corruption by learning a denoising distribution $p_\theta(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t)$ to recover $\mathbf{x}$ from a pure noise prior $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{0}$ denotes the zero vector. This is typically achieved by maximizing the ELBO on the log-likelihood $\log p_\theta(\mathbf{x})$. By parameterizing the reverse step with a score network $\mathbf{s}_\theta(\mathbf{x}_t, t)$, the ELBO simplifies to a weighted denoising score matching (DSM) objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\mathbf{x},\mathbf{x}_t}\left[w_t \left\| \mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \,|\, \mathbf{x}) \right\|_2^2 \right], \tag{2}$$

where the expectation is taken over timestep t, clean samples $\mathbf{x} \sim q(\mathbf{x})$, and noisy samples $\mathbf{x}_t \sim q(\mathbf{x}_t \,|\, \mathbf{x})$; w_t is a weighting function, and the analytic conditional score $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \,|\, \mathbf{x}) = (\alpha_t \mathbf{x} - \mathbf{x}_t)/\sigma_t^2$ serves as the tractable training target [40]. By varying schedules α_t, σ_t, w_t, this framework unifies Denoising Diffusion Probabilistic Models (DDPM) [1], Score-based SDEs [2], and Elucidated Diffusion Models (EDM) [41] (see Appendix C.1). In this work, we adopt the EDM formulation as our default backbone for its well-optimized parameterization.

Label-Conditional Diffusion Models (LCDMs). For label-conditional generation, the score network additionally takes a class label y as input and is trained to approximate the clean class-conditional score $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \,|\, y)$ via the conditional analog of Eq. (2):

$$\mathcal{L}_{\text{LCDM}}(\theta) = \mathbb{E}_{t,\mathbf{x},y,\mathbf{x}_t}\left[w_t \left\| \mathbf{s}_\theta(\mathbf{x}_t, y, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \,|\, \mathbf{x}, y) \right\|_2^2 \right]. \tag{3}$$

Eq. (3) is the standard LCDM objective and presupposes access to clean labels y.

Weak Annotations. In real-world settings, we observe pairs (X, Z) where the input $X \in \mathcal{X}$ is paired with a weak annotation Z that conveys imperfect information about the unobserved true label $Y \in \mathcal{Y} = \{1, \ldots, c\}$. We denote the dataset of N such observations as $\mathcal{D} = \{(\mathbf{x}^{(i)}, z^{(i)})\}_{i=1}^N$, and focus on three representative weak annotation scenarios:

- *Noisy Labels [42]:* The annotation $Z = \hat{Y}$ is a corrupted version of Y drawn from a noise transition $p(\hat{Y} \mid Y)$, common when labels come from user tags or automated taggers.
- *Ambiguous Labels [43]:* The annotation $Z \subseteq \mathcal{Y}$ is a candidate set containing the true label ($Y \in Z$), arising when the true class is hard to single out among look-alike categories.
- *Semi-supervised Data [44]:* The annotation $Z \in \mathcal{Y} \cup \{\emptyset\}$ is either the true label Y (labeled instances) or $\emptyset$ (unlabeled ones), since much of the collected data carries no label at all.

4 Methodology

In this section, we develop a unified probabilistic framework for training LCDMs under weak annotations, in which the learning objective is derived from a maximum-likelihood perspective and decomposed into generative and classification components.

4.1 Unified Probabilistic Framework

Given a dataset with inputs X and weak annotations Z, we propose to treat the unobserved true label Y as a latent variable, and to learn the optimal model parameters θ^* by maximizing the log-likelihood of the observed joint observation (X, Z):

$$\theta^* = \arg\max_\theta \log p_\theta(X, Z) = \arg\max_\theta \log \sum\nolimits_{Y \in \mathcal{Y}} p_\theta(X, Y, Z). \tag{4}$$

Since Eq. (4) involves the log of the marginalization over latent variables and cannot generally be solved in closed form, we instead maximize a variational lower bound on the log-likelihood:

$$\begin{aligned} \theta^* &= \arg\max_\theta \mathbb{E}_{p_\phi(Y \mid X, Z)} \big[\log p_\theta(X, Y, Z) \big] \\ &= \arg\max_\theta \Big\{ \underbrace{\mathbb{E}_{p_\phi(Y \mid X, Z)} \big[\log p_\theta(X \mid Z) \big]}_{\text{Generative Term}} + \underbrace{\mathbb{E}_{p_\phi(Y \mid X, Z)} \big[\log p_\theta(Y \mid X, Z) \big]}_{\text{Classification Term}} \Big\}, \end{aligned} \tag{5}$$

where $p_\phi(Y \mid X, Z)$ is a parameterized proxy for the true class-posterior, with a detailed derivation in Appendix C.2. This decomposition naturally gives rise to an EM-like training procedure: in the **E-step**, p_ϕ is evaluated with ϕ held fixed to supply per-sample soft posteriors; in the **M-step**, θ is updated via the two loss terms using these posteriors. In practice, we set ϕ to an exponential moving average (EMA) of θ, so the posteriors are stably refined alongside the model, and a better θ yields a more accurate p_ϕ that improves posterior estimates in the next E-step. The two terms reflect the dual role of the diffusion model: the *Generative Term* recovers clean class-conditional distributions, while the *Classification Term* aligns the model's own posterior $p_\theta(Y \mid X, Z)$ with the E-step soft targets, whose updates propagate back to p_ϕ via EMA. In this paper, we adopt the standard class-conditional assumption [45], where the generation of the weak annotation Z only depends on the true label Y, i.e., $p(Z \mid Y, X) = p(Z \mid Y)$. Each term in Eq. (5) is maximized by minimizing a corresponding loss, derived in Sections 4.2 and 4.3 respectively.

4.2 Generative Learning via Posterior-Weighted Score Matching

In this subsection, we focus on optimizing the *Generative Term* in Eq. (5) using only weak pairs $(\mathbf{x}, z)$ to learn a class-conditional score network that approximates the clean-conditional score.

Mathematically, since $\log p_\theta(X \mid Z)$ does not depend on Y, the expectation $\mathbb{E}_{p_\phi(Y \mid X, Z)}$ in Eq. (5) drops out, and the *Generative Term* reduces to maximizing $\log p_\theta(X \mid Z)$ directly. To learn the clean-conditional scores $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y)$ from data paired only with z, we need to bridge the weak and clean conditional distributions. Inspired by the relationships between noisy and clean distributions explored in noise-robust diffusion training [12], we formally establish the link between the observable *weak-conditional score* and the latent *clean-conditional scores* (Appendix C.3):

Theorem 4.1 (Score Decomposition under Weak Annotations)**.** *Under the class-conditional independence assumption ($X \perp Z \mid Y$), the score of the perturbed data density given a weak annotation z can be decomposed as a posterior-weighted combination of clean-conditional scores:*

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z) = \sum\nolimits_{y=1}^{c} p(y \mid \mathbf{x}_t, z)\, \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y). \tag{6}$$

Posterior-Weighted Score Matching. Theorem 4.1 reveals an asymmetry that motivates our loss design. The clean-conditional scores on the right-hand side are precisely what a class-conditional generative model needs to learn, but they cannot be targeted directly from weakly annotated data. The weak-conditional score on the left-hand side has the opposite property: it is not a desirable learning target since fitting it would produce mixed-class generations, yet it admits a closed-form training target through standard DSM against $(\mathbf{x}, z)$ pairs. Eq. (6) thus suggests a strategy: substituting the labels y in the standard LCDM objective Eq. (3) with weak annotations z and replacing the score network output $\mathbf{s}_\theta(\mathbf{x}_t, y, t)$ with its posterior-weighted aggregation, so that the clean-conditional network can be trained indirectly against the observable weak-conditional target:

$$\mathcal{L}_{\text{Gen}}(\theta) = \mathbb{E}_{t,\mathbf{x},z,\mathbf{x}_t}\left[w_t \left\| \sum\nolimits_{y=1}^{c} p_\phi(y \,|\, \mathbf{x}_t, z)\, \mathbf{s}_\theta(\mathbf{x}_t, y, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \,|\, \mathbf{x}, z) \right\|_2^2\right], \tag{7}$$

where the target $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}, z) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x})$ is the closed-form forward perturbation score. Since the true posterior $p(y \,|\, \mathbf{x}_t, z)$ in Eq. (6) is unobservable, we replace it with the EMA-parameterized estimate $p_\phi(y \mid \mathbf{x}_t, z)$ derived from the diffusion model itself via Bayes' rule, as described next. Intuitively, when $p_\phi(y \mid \mathbf{x}_t, z)$ is confident, the gradient primarily updates $\mathbf{s}_\theta$ for the most likely class, while an uncertain posterior spreads the gradient across multiple classes and prevents the model from anchoring to a potentially wrong label.

Proposition 4.2 (Identifiability of Clean-Conditional Scores)**.** *When the posterior estimate converges to the true posterior, i.e., $p_\phi(y \mid \mathbf{x}_t, z) = p(y \mid \mathbf{x}_t, z)$, the score network $\mathbf{s}_{\theta^*}$ that minimizes the posterior-weighted score matching objective recovers the clean-conditional scores:*

$$\mathbf{s}_{\theta^*}(\mathbf{x}_t, y, t) = \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \,|\, y), \quad \forall y \in \mathcal{Y}. \tag{8}$$

The proof is in Appendix C.4. In practice, ϕ is maintained as an EMA copy of θ, so p_ϕ is progressively refined alongside the model in an EM-style cycle: a better θ yields a more accurate p_ϕ, which in turn improves θ via Eq. (7). This iteration progressively drives p_ϕ toward the true posterior, eventually satisfying the condition of Proposition 4.2.

4.3 Classification via Diffusion-Based Posterior Inference

Both Eq. (7) and the *Classification Term* of Eq. (5) require computing class posteriors: $p_\phi(y \,|\, \mathbf{x}_t, z)$ for score weighting and $p_\theta(y \,|\, \mathbf{x}, z)$ for prediction. Training an external classifier on the same weak annotations would face an identical supervision problem and would introduce objective mismatch. We therefore compute these posteriors directly from the diffusion model itself: since $\log p_\theta(\mathbf{x}_t \,|\, y)$ is already modeled by the generative network, Bayes' rule yields the posterior with no extra parameters, allowing generation and classification to mutually reinforce each other [34].

Diffusion Classifier. Following Chen et al. [36], we obtain the exact posterior by applying Bayes' rule to the class-conditional log-likelihood and normalizing over all classes via a softmax, $p_\theta(y \mid \mathbf{x}_t) = \exp\{\log p_\theta(\mathbf{x}_t \,|\, y) + \log p(y)\} / \sum_{y' \in \mathcal{Y}} \exp\{\log p_\theta(\mathbf{x}_t \,|\, y') + \log p(y')\}$, where the conditional ELBO approximates the log-likelihood as

$$\log p_\theta(\mathbf{x}_t \,|\, y) \approx -\sum_{\tau=t+1}^{T-1} w_\tau\, \hbar(\tau, y), \quad \hbar(\tau, y) = \mathbb{E}_{\mathbf{x}_\tau \sim q(\mathbf{x}_\tau \,|\, \hat{\mathbf{x}})}\left[\|\mathbf{h}_\theta(\mathbf{x}_\tau, y, \tau) - \hat{\mathbf{x}}\|_2^2\right], \tag{9}$$

where $\mathbf{h}_\theta(\mathbf{x}_\tau, y, \tau) = (\mathbf{x}_\tau + \sigma_\tau^2\, \mathbf{s}_\theta(\mathbf{x}_\tau, y, \tau))/\alpha_\tau$ is the class-conditional denoised estimator, $\hat{\mathbf{x}}$ is the unconditional estimate of the clean sample from $\mathbf{x}_t$ (so $\hbar(\tau, y)$ implicitly depends on $\mathbf{x}_t$ through $\hat{\mathbf{x}}$), and w_τ is the EDM log-normal weight. We assume a uniform class prior $p(y) = 1/c$ throughout, with the extension to non-uniform priors deferred to Appendix D.2. For brevity, we write $f_\theta(\mathbf{x}_t) \in \Delta^{c-1}$ for the resulting normalized posterior, with $[f_\theta(\mathbf{x}_t)]_y = p_\theta(y \mid \mathbf{x}_t)$, and let f_ϕ denote its EMA counterpart, so that $p_\phi(y \,|\, \mathbf{x}_t) = [f_\phi(\mathbf{x}_t)]_y$.

Classification Term under Weak Annotations. Under the class-conditional assumption of Section 4.1, the model posterior factorizes as $p_\theta(y \,|\, \mathbf{x}, z) = p(z \,|\, y)\, p_\theta(y \,|\, \mathbf{x}) / p_\theta(z \,|\, \mathbf{x})$. Since $p(z \,|\, y)$ is θ-independent and $\log p_\theta(z \,|\, \mathbf{x}) \le 0$, the *Classification Term* in Eq. (5) admits the tractable lower bound $\mathbb{E}_{p_\phi(y|\mathbf{x},z)}[\log p_\theta(y \,|\, \mathbf{x})]$ (derivation and boundedness of the dropped term in Appendix C.5), whose maximization is equivalent to minimizing the posterior-weighted cross-entropy below. To

Table 1: Generative results under various weak annotations. The gray-shaded 'Clean' column represents the best performance trained on the fully clean dataset. 'uncond'/'cond' denote unconditional/conditional metrics. **Bold** numbers indicate better performance.

		Metric		Noisy label				Ambiguous label				Semi-supervised data				Clean
				Sym-40%		Asym-40%		Random		Class-dept.		1% Labels		10% Labels		
				Vanilla	*DELTA*	*Vanilla*	*DELTA*	*Vanilla*	*DELTA*	*Vanilla*	*DELTA*	*Vanilla*	*DELTA*	*Vanilla*	*DELTA*	
CIFAR-10	uncond	FID	(↓)	**3.33**	3.47	3.23	**3.10**	7.76	**2.26**	11.75	**2.77**	3.16	**3.12**	2.93	**2.89**	2.05
		IS	(↑)	9.56	**9.68**	9.02	**9.73**	9.09	**9.80**	9.62	**9.68**	10.03	**10.57**	9.80	**9.83**	10.61
		Density	(↑)	101.39	**109.75**	100.06	**109.69**	103.21	**106.49**	108.76	**109.06**	97.19	**108.18**	99.96	**108.87**	112.59
		Coverage	(↑)	81.12	**81.21**	80.71	**81.30**	68.45	**82.69**	64.90	**81.52**	78.44	**81.00**	81.85	**82.00**	83.27
	cond	CW-FID	(↓)	29.84	**13.85**	14.70	**13.24**	27.18	**10.65**	32.44	**11.56**	16.25	**16.12**	11.84	**11.77**	9.83
		CW-Density	(↑)	72.98	**107.23**	90.85	**107.07**	102.04	**105.75**	102.43	**108.66**	89.99	**100.73**	96.29	**107.94**	111.70
		CW-Coverage	(↑)	73.39	**80.11**	79.63	**79.65**	65.45	**82.09**	61.45	**81.24**	75.03	**76.84**	80.80	**81.12**	83.91
ImageNette	uncond	FID	(↓)	14.11	**13.44**	13.93	**13.91**	79.13	**72.62**	91.28	**79.12**	23.88	**19.26**	14.32	**12.84**	11.52
		IS	(↑)	12.69	**13.21**	12.51	**13.73**	9.19	**9.40**	**9.27**	9.11	12.23	**13.72**	12.80	**13.16**	13.81
		Density	(↑)	109.31	**112.52**	**111.66**	106.78	95.33	**99.83**	94.29	**102.58**	115.94	**125.68**	105.27	**109.23**	117.23
		Coverage	(↑)	76.62	**76.81**	78.32	**79.81**	21.44	**32.48**	16.69	**22.30**	53.53	**55.39**	73.79	**75.55**	80.12
	cond	CW-FID	(↓)	80.31	**60.12**	62.26	**58.20**	157.76	**63.58**	163.45	**67.92**	71.66	**70.27**	49.22	**44.31**	40.20
		CW-Density	(↑)	73.99	**81.12**	93.53	**94.58**	93.38	**95.83**	91.50	**95.21**	115.90	**118.69**	103.41	**115.67**	120.35
		CW-Coverage	(↑)	67.89	**71.94**	74.18	**75.82**	19.76	**24.35**	15.88	**18.93**	51.73	**52.15**	72.61	**74.85**	78.48

improve optimization robustness under weak annotations, we replace the raw posterior estimate $p_\phi(y \mid \mathbf{x}, z)$ with a rectified target $\tilde{p}_\phi(y \mid \mathbf{x}, z)$. For each weak-annotation regime, the rectification exploits its specific structure, yielding a unified family of objectives that generalizes three regularizers:

$$\mathcal{L}_{\text{Cls}}(\theta) = -\mathbb{E}_{\mathbf{x},z}\Big[\sum\nolimits_{y\in\mathcal{Y}} \tilde{p}_\phi(y \mid \mathbf{x}, z)\, \log[f_\theta(\mathbf{x})]_y\Big], \quad \text{with}$$

$$\tilde{p}_\phi(y \mid \mathbf{x}, z) = \begin{cases} \text{sg}\big(\mathbb{I}(y{=}z) - [f_\theta(\mathbf{x})]_y\,(\delta - [f_\phi(\mathbf{x})]_y)/(1-\delta)\big) & \text{(Noisy)} \\ \mathbb{I}(y \in z)\,[f_\phi(\mathbf{x})]_y \,/\, \sum_{y'\in z}[f_\phi(\mathbf{x})]_{y'} & \text{(Ambiguous)} \\ \mathbb{I}(z \in \mathcal{Y})\,\mathbb{I}(y = z) + \mathbb{I}(z = \emptyset)\,[f_\phi(\mathbf{x})]_y & \text{(Semi-sup.)} \end{cases} \tag{10}$$

For *noisy labels*, $\text{sg}(\cdot)$ denotes the stop-gradient operator and $\delta = \langle f_\theta(\mathbf{x}), f_\phi(\mathbf{x})\rangle$ is the inner product between the current and EMA posteriors; the closed-form rectification is a correction term we derive to suppress noisy memorization (Appendix D.1), generalizing early-learning regularization [46]. For *ambiguous labels*, our masked renormalization of $f_\phi(\mathbf{x})$ over the candidate set z extends EMA-stabilized progressive identification [47]. For *semi-supervised data*, our indicator switch reduces to mean-teacher self-training [48] on the unlabeled portion.

4.4 Efficient Posterior Estimation via Optimized Timestep Sampling

Computing $p_\theta(y \mid \mathbf{x}_t)$ via Eq. (9) requires evaluating the conditional ELBO across all T timesteps and classes, which is the dominant per-iteration cost. Chen et al. [36] alleviate this by sharing $\mathbf{x}_\tau$ across classes and sampling τ uniformly from $\{t+1, \ldots, T-1\}$, yet Figure 2(b) shows uniform sampling is far from efficient. The single-noise-level accuracy curve in Figure 2(a) explains why: low noise levels σ_τ induce trivial reconstructions with low label sensitivity, while high noise levels are noise-dominated and unreliable, so only an intermediate band carries meaningful class signal.

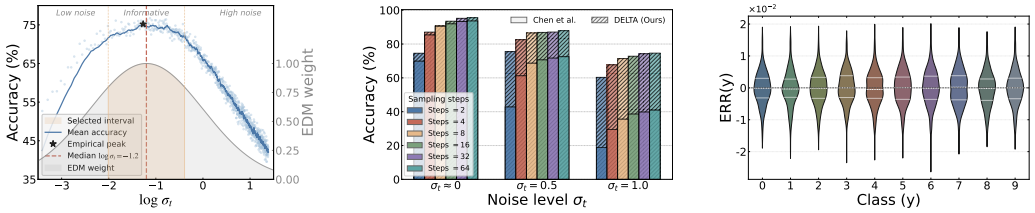
Optimal Timestep Sampling. We therefore seek a contiguous sub-range $\{l, \ldots, r\}$ of fixed width $\Delta = r - l$ such that its weighted average approximates the full sum:

$$\Big\| \frac{1}{\Delta} \sum_{\tau=l}^{r} w_\tau \hbar(\tau, y) - \frac{1}{T-1} \sum_{\tau=1}^{T-1} w_\tau \hbar(\tau, y) \Big\|_2^2. \tag{11}$$

Since $\hbar(\tau, y)$ is data-dependent and evolves during training, directly optimizing Eq. (11) is impractical. We instead adopt a $\hbar$-agnostic surrogate: choose the sub-range $\{l, \ldots, r\}$ that maximizes the total weight $\sum_{\tau=l}^{r} w_\tau$, so that the partial sum closely approximates the full sum, as illustrated in Figure 2(a) (orange areas: the partial sum; gray areas: the full sum). For the EDM log-normal weight, this maximization is modeled through the interval-mass objective and solved numerically, yielding a window centered around the median region of the weight density, as formalized below.

Theorem 4.3 (Median-Centered Interval via Weight Mass)**.** *For the EDM log-normal weight $w(\tau)$ with cumulative weight $\mathcal{F}_w(b) = \int_0^b w(\tau)\,\mathrm{d}\tau$ and a fixed length Δ, the interval $[l, r]$ that maximizes the covered weight mass $\mathcal{F}_w(r) - \mathcal{F}_w(l)$ is uniquely determined by*

$$\mathcal{F}_w(l) + \mathcal{F}_w(l + \Delta) = 1, \qquad r = l + \Delta. \tag{12}$$



(a) EDM weight & Acc. distribution. (b) Accuracy across sampling budgets. (c) Class-wise ERR distribution.

Figure 2: Noise-level sampling analysis on CIFAR-10. (a): Single-timestep accuracy (blue) and EDM weight (gray). (b): Our median-centered scheme outperforms the uniform sampling of Chen et al. [36] at all budgets. (c): Violin distribution of the optimality residual ERR; closer to zero is better.

The optimal l is solved by numerical root-finding [49]. As $\Delta \to 0$ the interval collapses to the log-normal weight-density median $e^{-1.2}$ ($\log \sigma_t = -1.2$) and matches Figure 2(a). The proof and an analogous result for DDPM [1] are given in Appendix C.6.

5 Experiments

We evaluate *DELTA* on two categories of datasets. **Synthetic benchmarks** apply controlled corruptions to Fashion-MNIST [50], CIFAR-10 [51], and ImageNette [52]. **Real-world benchmarks** use naturally occurring weak annotations: the noisy-label sets CIFAR10-N and CIFAR100-N [53], and the ambiguous-label set PLCIFAR10 [22]. All models are trained from scratch under the EDM framework; full implementation details are in Appendix B. As an internal ablation, *Vanilla* uses the same joint objective structure as *DELTA* but replaces the posterior-weighted generative term (Eq. 7) with a naive formulation that trains directly on weak pairs $(\mathbf{x}, z)$ without posterior reweighting.

Weak Annotation Protocols for Synthetic Benchmarks. We corrupt synthetic benchmarks as follows: (1) **Noisy Label**: labels are flipped with 40% probability via symmetric noise ('Sym-40%', uniform flipping to any incorrect class) or asymmetric noise ('Asym-40%', flipping to semantically similar classes); (2) **Ambiguous Label**: candidate sets are generated via a random policy ('Random', including non-target labels with 50% probability) or a class-dependent policy ('Class-Dept.', selecting semantically similar labels with specified probabilities); (3) **Semi-Supervised Data**: labeled samples are retained at a fixed fraction per class, denoted as '1% Labels' and '10% Labels'.

5.1 Image Generation

Setup. We assess generation quality with four unconditional metrics: FID, IS, Density (abbreviated as 'Dens.'), and Coverage (abbreviated as 'Cov.') [54], and three class-conditional metrics: CW-FID, CW-Density, and CW-Coverage [55]. Detailed metric definitions are in Appendix B.3.

Comparison with Vanilla. Table 1 reports generation quality across all six weak annotation settings on CIFAR-10 and ImageNette. *DELTA* consistently outperforms *Vanilla* on both unconditional and conditional metrics. The improvement is most pronounced under ambiguous-label settings, where the naive posterior collapses to a mixture, causing class-image mismatch and per-class homogenization. For example, *Vanilla*'s CW-FID on CIFAR-10 (random ambiguous) is 27.18 versus *DELTA*'s 10.65. Across all settings, *DELTA*'s results approach the clean-label upper bound (gray column), confirming that posterior-weighted score matching effectively resolves the mixture bias.

Comparison with Noise-Robust Diffusion Baselines. Table 2 benchmarks *DELTA* against four dedicated noise-robust diffusion methods (CAD [27], TDSM [12], SBDC [30], and RTDC [28]) on CIFAR-10 with 40% noisy labels. These baselines each assume access to task-specific auxiliary supervision such as transition matrices, oracle confidence scores, or auxiliary discriminators, whereas *DELTA* operates without such priors. Despite this strictly weaker assumption, *DELTA* achieves the best results on nearly all metrics, and in particular dominates the class-conditional metrics that directly measure class–image alignment, demonstrating that our posterior-weighted formulation is a practically stronger alternative. Notably, *Vanilla* attains a competitive unconditional FID under Sym-40% yet the worst CW-FID, confirming that unconditional metrics alone are insensitive to the label-condition mismatch that robust training aims to resolve.



Figure 3: The three left panels show random samples from *DELTA* trained under symmetric 40% label noise (Sym-40%), on Fashion-MNIST, CIFAR-10, and ImageNette (left to right). The rightmost panel reports classification accuracy of *DELTA* with and without pseudo-label correction.

Table 2: Comparison against noise-robust diffusion baselines on CIFAR-10.

	Sym-40%							Asym-40%						
Method	FID↓	IS↑	Dens.↑	Cov.↑	CW-FID↓	CW-Dens.↑	CW-Cov.↑	FID↓	IS↑	Dens.↑	Cov.↑	CW-FID↓	CW-Dens.↑	CW-Cov.↑
Vanilla	**3.33**	9.56	101.39	81.12	29.84	72.98	73.39	3.23	9.02	100.06	80.71	14.70	90.85	79.63
CAD [27]	4.10	9.08	101.48	77.04	19.66	93.51	76.43	3.87	9.16	98.27	75.81	14.33	99.25	78.12
TDSM [12]	3.85	9.40	106.32	79.66	16.42	101.27	78.63	3.96	**10.12**	107.85	80.47	14.96	102.67	78.99
SBDC [30]	4.25	8.95	97.83	74.92	21.30	89.83	75.37	4.02	9.05	95.14	73.29	15.70	96.14	77.25
RTDC [28]	4.00	9.20	104.51	78.35	17.92	98.13	77.70	3.85	9.30	103.63	78.06	13.92	101.03	78.68
DELTA	3.47	**9.68**	**109.75**	**81.21**	**13.85**	**107.23**	**80.11**	**3.10**	9.73	**109.69**	**81.30**	**13.24**	**107.07**	**79.65**

Performance on Real-World Weak Annotations. Table 3 shows that under real-world weak annotations (CIFAR10-N, PLCIFAR10, CIFAR100-N), *DELTA* achieves low FID, competitive IS scores, and strong generation quality, validating its effectiveness on real-world datasets.

Table 3: Performance on real-world weakly annotated datasets.

Dataset	FID↓	IS↑	Dens.↑	Cov.↑
PLCIFAR10	2.95	9.82	109.1	81.7
CIFAR10-N	3.22	9.66	108.3	80.9
CIFAR100-N	4.85	10.42	96.4	71.2

Table 4: Inference latency (CIFAR-10, single A100).

Metric	Chen et al.	*DELTA*
Time (min)	18.43	**0.67**
Acc (%)	87.13	**88.63**
Speedup	1×	**27.5**×

Qualitative Results. The three left panels of Figure 3 show random class-conditional samples from *DELTA* trained under symmetric 40% label noise. Although the training labels are heavily corrupted, the generated images remain visually consistent with the class condition, preserve distinct modes rather than homogenizing, and retain clear structure instead of degrading into low-quality outputs.

5.2 Probing the Learned Posterior

In this subsection, we examine the quality of the learned class posteriors $p_\phi(y \mid \mathbf{x}, z)$ through two lenses: its evolution during training and its accuracy as a standalone classifier.

Evolution of Learned Class Posteriors. Figure 4 traces the classification accuracy of p_ϕ throughout training on CIFAR-10 under all six weak annotation settings. *Vanilla*'s accuracy rises initially before degrading, reflecting that a naive posterior collapses under sustained label noise. In contrast, *DELTA*'s posterior improves monotonically and stabilizes, validating the EM-style refinement described in Section 4: a better generative model yields a more accurate p_ϕ, which in turn guides a better update.

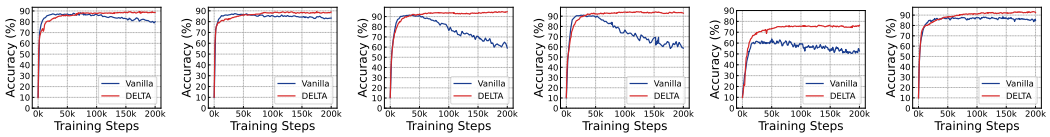
Posterior as a Classifier. A converged p_ϕ can be directly evaluated as a discriminative classifier. Table 5 compares *DELTA* against advanced specialized discriminative baselines across all three weak annotation types on Fashion-MNIST and CIFAR-10. For fairness, all discriminative classifiers use Wide-ResNet-40-10 (55.84M parameters), matching the parameter count of our diffusion model (55.73M), and all models are trained from scratch. Despite being a generative framework, *DELTA* consistently achieves the best accuracy across all settings, demonstrating that diffusion-based posterior inference is a strong discriminative signal in its own right.

5.3 Inference Efficiency of Timestep Sampling

Section 4.4 proposes a median-centered interval timestep sampling strategy that concentrates posterior estimation on the most informative noise levels. We validate its efficiency and effectiveness empirically. Table 4 confirms that it reduces per-step posterior estimation cost by $27.5\times$ over full-timestep

Table 5: Classification results (test accuracy, %) under various weak annotations (♠: noisy label, ♡: ambiguous label, ♣: semi-supervised data). **Bold** denotes best performance.

Dataset	Type	*CE*	*Mixup* [56]	*Coteaching* [57]	*ELR* [46]	*PENCIL* [58]	*Vanilla*	*DELTA*
Fashion-MNIST♠	Sym-40%	76.18±0.26	92.21±0.03	92.17±0.34	93.13±0.13	90.85±0.58	90.11±1.24	**93.40±0.40**
	Asym-40%	82.01±0.06	92.01±1.02	92.78±0.25	92.82±0.09	91.77±0.69	85.41±0.96	**93.20±0.30**
CIFAR-10♠	Sym-40%	67.22±0.26	84.26±0.64	86.54±0.57	85.68±0.13	85.91±0.26	80.22±0.10	**88.63±0.12**
	Asym-40%	76.98±0.42	83.21±0.85	79.38±0.39	81.32±0.31	84.89±0.49	86.31±0.10	**88.83±0.33**
	Type	*PRODEN* [47]	*IDGP* [59]	*PiCO* [60]	*CRDPLL* [61]	*DIRK* [62]	*Vanilla*	*DELTA*
Fashion-MNIST♡	Random	93.31±0.07	92.26±0.25	93.32±0.12	94.03±0.14	94.11±0.22	80.20±1.29	**94.27±0.55**
	Class-Dept.	93.44±0.21	93.07±0.16	93.32±0.33	93.80±0.23	93.99±0.24	66.03±1.43	**94.20±0.15**
CIFAR-10♡	Random	90.02±0.22	89.65±0.53	86.40±0.89	92.74±0.26	93.48±0.14	60.25±0.17	**94.70±0.49**
	Class-Dept.	90.44±0.44	90.83±0.34	87.51±0.66	92.89±0.27	93.22±0.37	56.34±0.50	**93.53±0.12**
	Type	*Dash* [63]	*CoMatch* [64]	*FlexMatch* [65]	*SimMatch* [66]	*SoftMatch* [67]	*Vanilla*	*DELTA*
Fashion-MNIST♣	1% Labels	84.73±0.09	85.31±0.29	84.43±0.30	84.69±0.17	84.72±0.23	78.37±0.72	**85.92±0.13**
	10% Labels	91.16±0.20	90.52±0.12	90.69±0.03	91.18±0.13	91.22±0.11	90.50±1.00	**92.97±0.21**
CIFAR-10♣	1% Labels	70.14±0.69	61.45±1.46	70.72±0.93	73.33±1.02	73.74±0.82	53.49±0.15	**76.40±0.54**
	10% Labels	81.50±0.68	77.79±0.53	81.35±0.48	82.90±0.43	88.66±0.60	85.13±0.12	**92.47±0.39**



(a) NL Sym-40% (b) NL Asym-40% (c) AL Random (d) AL Class-Dept. (e) SS 1% Labels (f) SS 10% Labels

Figure 4: Test accuracy curves of *Vanilla* versus *DELTA* on CIFAR-10 under various weak annotations: Noisy Labels (NL), Ambiguous Labels (AL), and Semi-Supervised data (SS).

computation [36] with no accuracy loss. Figure 2(b) further shows that the median-centered scheme consistently outperforms uniform sampling at every step budget, with the gap widening as the budget shrinks. Figure 2(c) reports the class-wise residual $\textsc{Err}(y)$ as a violin plot, where $\textsc{Err}(y)$ measures the deviation from the necessary condition of Eq. (11) in Appendix C.7; smaller absolute values indicate less deviation. The distribution is tightly concentrated near zero across all classes, confirming that the median-centered timestep sampling strategy achieves effective coverage in practice.

5.4 Synergy with Pseudo-label Correction

Our generative framework is orthogonal to discriminative label correction strategies. To validate this complementarity, we use DivideMix [68] to assign pseudo-labels to the noisy-label training set and then train *DELTA* on those pseudo-labeled examples. Pseudo-labels are never fully reliable, so residual label noise remains and the supervision stays noisy. As shown in the rightmost panel of Figure 3 on CIFAR-10 under symmetric 40% label noise, *DELTA* consistently improves upon the corrector's standalone performance, validating that our method effectively complements existing approaches by acting as a robust denoiser for residual noise.

6 Discussion and Limitations

Beyond the experiments above, a potential extension of *DELTA* is *weakly supervised* dataset condensation [69], which aims to distill large weakly annotated datasets into clean and compact synthetic counterparts for efficient downstream training, such as neural architecture search [70, 71]. Most prior methods [72–76] assume clean labels. By contrast, *DELTA* acts as a *generative compressor*, compressing the original, weakly annotated training data into a model that can synthesize clean, class-representative samples on demand. Appendix F provides further discussion and experiments.

Our limitations are mainly twofold. On one hand, we assume a fixed and finite label space; we therefore do not treat unseen categories at test time or settings with a gradually expanding label space, and extending *DELTA* to such cases is left for future work. On the other hand, our theory relies on class-conditional weak annotations, where the annotation mechanism depends on the latent class but not directly on instance-specific visual factors. In practice, data-collection pipelines may also depend on image difficulty or style. Relaxing this assumption would be a useful direction for future work.

7 Conclusion

We presented *DELTA*, a unified framework for training LCDMs under weak annotations. By treating the unobserved clean label as a latent variable, *DELTA* optimizes a single variational objective that couples posterior-weighted score matching with diffusion-based posterior refinement, allowing generation and label inference to reinforce each other within one model. A median-centered timestep sampling strategy further makes the per-sample posterior estimation tractable during joint training. Unlike prior approaches that target a single form of weak annotation and rely on auxiliary priors, *DELTA* handles all weak annotations within one objective while requiring less prior knowledge, offering a step toward making diffusion models broadly robust to imperfect supervision.

Acknowledgments

DDW was supported by the RIKEN Junior Research Associate Program. MS was supported by JST ASPIRE Grant Number JPMJAP2405.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[2] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[3] Michael Fuest, Pingchuan Ma, Ming Gui, Johannes Schusterbauer, Vincent Tao Hu, and Björn Ommer. Diffusion models and representation learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):7209–7228, 2026.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10674–10685, 2022.

[5] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646, 2022.

[6] Pingyu Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Improved video VAE for latent video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18124–18133, 2025.

[7] Ben Fei, Yixuan Li, Weidong Yang, Hengjun Gao, Jingyi Xu, Lipeng Ma, Yatian Yang, and Pinghong Zhou. A diffusion model for universal medical image enhancement. *Communications Medicine*, 5(1): 294, 2025.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 12606–12633, 2024.

[9] Pu Cao, Feng Zhou, Qing Song, and Lu Yang. Controllable generation with text-to-image diffusion models: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4):4771–4791, 2026.

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[11] Rahul Shenoy, Zhihong Pan, Kaushik Balakrishnan, Qisen Cheng, Yongmoon Jeon, Heejune Yang, and Jaewon Kim. Gradient-free classifier guidance for diffusion model sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3162–3171, 2026.

[12] Byeonghu Na, Yeongmin Kim, HeeSun Bae, Jung Hyun Lee, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Label-noise robust diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[13] Yuanzhi Zhu, Zhaohai Li, Tianwei Wang, Mengchao He, and Cong Yao. Conditional text image generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14235–14244, 2023.

[14] Zheyuan Zhan, Defang Chen, Jian-Ping Mei, Zhenghe Zhao, Jiawei Chen, Chun Chen, Siwei Lyu, and Can Wang. Conditional image synthesis with diffusion models: A survey. *Transactions on Machine Learning Research*, 2025.

[15] Chihiro Noguchi, Shun Fukuda, and Masao Yamanaka. Scene text image super-resolution based on text-conditional diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1485–1495, 2024.

[16] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision*, pages 181–196, 2018.

[17] Amirhossein Kazerouni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz, Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive survey. *Medical Image Analysis*, 88:102846, 2023.

[18] Srikar Yellapragada, Alexandros Graikos, Zilinghan Li, Kostas Triaridis, Varun Belagali, Saarthak Kapse, Tarak Nath Nandi, Ravi K. Madduri, Prateek Prasanna, Tahsin Kurc, Rajarsi R. Gupta, Joel Saltz, and Dimitris Samaras. PixCell: A generative foundation model for digital histopathology images. *arXiv preprint arXiv:2506.05127*, 2025.

[19] Samar Khanna, Patrick Liu, Linqi Zhou, Chenlin Meng, Robin Rombach, Marshall Burke, David B. Lobell, and Stefano Ermon. DiffusionSat: A generative foundation model for satellite imagery. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Datao Tang, Xiangyong Cao, Xingsong Hou, Zhongyuan Jiang, Junmin Liu, and Deyu Meng. CRS-Diff: Controllable remote sensing image generation with diffusion model. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024.

[21] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. WebVision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[22] Wei Wang, Dong-Dong Wu, Jindong Wang, Gang Niu, Min-Ling Zhang, and Masashi Sugiyama. Realistic evaluation of deep partial-label learning algorithms. In *The Thirteenth International Conference on Learning Representations*, 2025.

[23] Hsiu-Hsuan Wang, Tan-Ha Mai, Nai-Xuan Ye, Wei-I Lin, and Hsuan-Tien Lin. CLImage: Human-annotated datasets for complementary-label learning. *Transactions on Machine Learning Research*, 2025.

[24] Wei He, Kai Han, Ying Nie, Chengcheng Wang, and Yunhe Wang. Species196: A one-million semi-supervised dataset for fine-grained species recognition. In *Advances in Neural Information Processing Systems*, pages 44957–44975, 2023.

[25] Masashi Sugiyama, Han Bao, Takashi Ishida, Nan Lu, Tomoya Sakai, and Gang Niu. *Machine Learning from Weak Supervision: An Empirical Risk Minimization Approach*. Adaptive Computation and Machine Learning. MIT Press, 2022.

[26] Yangming Li, Max Ruiz Luyten, and Mihaela van der Schaar. Risk-sensitive diffusion: Robustly optimizing diffusion models with noisy samples. In *International Conference on Learning Representations*, 2025.

[27] Nicolas Dufour, Victor Besnier, Vicky Kalogeiton, and David Picard. Don't drop your samples! Coherence-aware training benefits conditional diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6264–6273, 2024.

[28] Xin Chen, Gillian Dobbie, Xinyu Wang, Feng Liu, Di Wang, and Jingfeng Zhang. Robust learning of diffusion models with extremely noisy conditions. *arXiv preprint arXiv:2510.10149*, 2025.

[29] Zebin You, Yong Zhong, Fan Bao, Jiacheng Sun, Chongxuan Li, and Jun Zhu. Diffusion models and semi-supervised learners benefit mutually with few labels. In *Advances in Neural Information Processing Systems*, pages 43479–43495, 2023.

[30] Dat Nguyen Cong, Hieu Tran Bao, and Tung Hoang-Thanh. Guiding noisy label conditional diffusion models with score-based discriminator correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18531–18541, 2025.

[31] Hiroshi Takahashi, Tomoharu Iwata, Atsutoshi Kumagai, Yuuki Yamanaka, and Tomoya Yamashita. Positive-unlabeled diffusion models for preventing sensitive data generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[32] Weichen Yu, Ziyan Yang, Shanchuan Lin, Qi Zhao, Jianyi Wang, Liangke Gui, Matt Fredrikson, and Lu Jiang. Is your text-to-image model robust to caption noise? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pages 3789–3798, 2026.

[33] Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero-shot classifiers. In *Advances in Neural Information Processing Systems*, volume 36, pages 58921–58937, 2023.

[34] Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2206–2217, 2023.

[35] Huanran Chen, Yinpeng Dong, Zhengyi Wang, Xiao Yang, Chengqi Duan, Hang Su, and Jun Zhu. Robust classification via a single diffusion model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 6643–6665, 2024.

[36] Huanran Chen, Yinpeng Dong, Shitong Shao, Zhongkai Hao, Xiao Yang, Hang Su, and Jun Zhu. Diffusion models are certifiably robust classifiers. In *Advances in Neural Information Processing Systems*, pages 50062–50097, 2024.

[37] Yanghao Wang and Long Chen. Noise matters: Optimizing matching noise for diffusion classifiers. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[38] Zipeng Qi, Buhua Liu, Shiyan Zhang, Bao Li, Zhiqiang Xu, Haoyi Xiong, and Zeke Xie. A simple and efficient baseline for zero-shot generative classification. *arXiv preprint arXiv:2412.12594*, 2024.

[39] Yujin Jeong, Arnas Uselis, Seong Joon Oh, and Anna Rohrbach. Diffusion classifiers understand compositionality, but conditions apply. In *Advances in Neural Information Processing Systems*, 2025.

[40] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[41] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, pages 26565–26577, 2022.

[42] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W. Tsang, James T. Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020.

[43] Yingjie Tian, Xiaotong Yu, and Saiji Fu. Partial label learning: Taxonomy, analysis and outlook. *Neural Networks*, 161:708–734, 2023.

[44] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023.

[45] 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, pages 21475–21496, 2023.

[46] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems*, volume 33, pages 20331–20342, 2020.

[47] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 6500–6510, 2020.

[48] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017.

[49] Richard P Brent. *Algorithms for Minimization without Derivatives*. Courier Corporation, 2013.

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

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

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

[53] 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.

[54] Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, Yi-Chen Lo, Chia-Che Chang, Yu-Lun Liu, Yu-Lin Chang, Chia-Ping Chen, and Chun-Yi Lee. Denoising likelihood score matching for conditional score-based data generation. In *International Conference on Learning Representations*, 2022.

[55] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Label-noise robust generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2467–2476, 2019.

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

[57] 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*, pages 8527–8537, 2018.

[58] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019.

[59] Congyu Qiao, Ning Xu, and Xin Geng. Decompositional generation process for instance-dependent partial label learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[60] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. PiCO+: Contrastive label disambiguation for robust partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3183–3198, 2024.

[61] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 24212–24225, 2022.

[62] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Distilling reliable knowledge for instance-dependent partial label learning. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, volume 38, pages 15888–15896, 2024.

[63] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 11525–11536, 2021.

[64] Junnan Li, Caiming Xiong, and Steven C. H. Hoi. CoMatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021.

[65] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, pages 18408–18419, 2021.

[66] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. SimMatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14471–14481, 2022.

[67] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. SoftMatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *International Conference on Learning Representations*, 2023.

[68] 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.

[69] 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.

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

[71] Jiawei Du, Yidi Jiang, Vincent Y. F. Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3758, 2023.

[72] 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*, pages 7856–7865, 2023.

[73] Yang He, Lingao Xiao, Joey Tianyi Zhou, and Ivor W. Tsang. Multisize dataset condensation. In *The Twelfth International Conference on Learning Representations*, 2024.

[74] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 12352–12364, 2022.

[75] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 11102–11118, 2022.

[76] 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*, pages 10718–10727, 2022.

[77] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035, 2019.

[78] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, volume 33, pages 12438–12448, 2020.

[79] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[80] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[81] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7176–7185, 2020.

[82] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018.

[83] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

[84] Junxiang Wu, Zhiqiang Kou, Hongwei Zeng, Wenke Huang, Biao Liu, Hanlin Gu, Yuheng Jia, Di Jiang, Yang Liu, and Xin Geng. Trustworthy federated label distribution learning under annotation quality disparity. In *Forty-third International Conference on Machine Learning*, 2026.

[85] Zhiqiang Kou, Junyang Chen, Xin-Qiang Cai, Xiaobo Xia, Ming-Kun Xie, Dong-Dong Wu, Biao Liu, Yuheng Jia, Xin Geng, Masashi Sugiyama, and Tat-Seng Chua. Positive–unlabeled reinforcement learning distillation for on-premise small models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 60447–60479, 2026.

[86] Jinfu Fan, Jiangnan Li, Xiaohui Zhong, Kangrui Ren, Zhencun Jiang, Min Gan, Tianhao Gu, and Linqing Huang. Evidential deep partial label learning to quantify disambiguation uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24770–24779, 2026.

[87] Jinfu Fan, Jiangnan Li, Linqing Huang, Min Gan, and CL Philip Chen. Partial label zero-shot learning with semantic mining and instance–label alignment. *Engineering Applications of Artificial Intelligence*, 177:114854, 2026.

[88] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

[89] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

[90] 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.

[91] Haobo Wang, Mingxuan Xia, Yixuan Li, Yuren Mao, Lei Feng, Gang Chen, and Junbo Zhao. SoLar: Sinkhorn label refinery for imbalanced partial-label learning. In *Advances in Neural Information Processing Systems*, pages 8104–8117, 2022.

[92] Zhiqiang Kou, Yucheng Xie, Hailin Wang, Junyang Chen, Jingq Wang, Ming-Kun Xie, Shuo Chen, Yuheng Jia, Tongliang Liu, and Xin Geng. Rankmatch: A novel approach to semi-supervised label distribution learning leveraging rank correlation between labels. In *Proceedings of the 39th Conference on Neural Information Processing Systems*, 2025.

[93] Zhiqiang Kou, Si Qin, Hailin Wang, Jing Wang, Mingkun Xie, Shuo Chen, Yuheng Jia, Tongliang Liu, Masashi Sugiyama, and Xin Geng. Label distribution learning with biased annotations assisted by multi-label learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

[94] Zhiqiang Kou, Jing Wang, Jiawei Tang, Yuheng Jia, Boyu Shi, and Xin Geng. Exploiting multi-label correlation in label distribution learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 4326–4334, 2024.

[95] Dong-Dong Wu, Chilin Fu, Weichang Wu, Wenwen Xia, Xiaolu Zhang, Jun Zhou, and Min-Ling Zhang. Efficient model stealing defense with noise transition matrix. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24305–24315. IEEE, 2024.

[96] Dong-Dong Wu, Zhaoyi Li, Xiang Li, and Zhiqiang Shen. Making partial-label datasets easier: A simple yet highly effective data augmentation for deep partial-label learning. In *European Conference on Computer Vision*, pages 506–524. Springer, 2026.

[97] Yu Shi, Dong-Dong Wu, Xin Geng, and Min-Ling Zhang. Robust representation learning for unreliable partial label learning. *arXiv preprint arXiv:2308.16718*, 2023.

[98] 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, pages 7260–7271, 2020.

[99] Zhiqiang Kou, Junxiang Wu, Wenke Huang, Wenwen He, Ming-Kun Xie, Changwei Wang, Yuheng Jia, Di Jiang, Yang Liu, Xin Geng, and Qiang Yang. Fedharmony: Harmonizing heterogeneous label correlations in federated multi-label learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[100] Zhiqiang Kou, Jing Wang, Yuheng Jia, Biao Liu, and Xin Geng. Instance-dependent inaccurate label distribution learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):1425–1437, 2025.

[101] JinFu Fan, Xiaohui Zhong, Kangrui Ren, JiangNan Li, Linqing Huang, Min Gan, and CL Philip Chen. Diffusion disambiguation models for partial label learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(10):14621–14636, 2026.

[102] Jinfu Fan, Linqing Huang, Chaoyu Gong, Yang You, Min Gan, and Zhongjie Wang. Kmt-pll: K-means cross-attention transformer for partial label learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2789–2800, 2025.

[103] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 6403–6413, 2021.

[104] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7164–7173, 2019.

[105] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[106] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, pages 8792–8802, 2018.

[107] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *Advances in Neural Information Processing Systems*, pages 960–970, 2017.

[108] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020.

[109] Fei Zhang, Lei Feng, Bo Han, Tongliang Liu, Gang Niu, Tao Qin, and Masashi Sugiyama. Exploiting class activation value for partial-label learning. In *International Conference on Learning Representations*, 2022.

[110] Ning Xu, Biao Liu, Jiaqi Lv, Congyu Qiao, and Xin Geng. Progressive purification for instance-dependent partial label learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 38551–38565, 2023.

[111] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 10948–10960, 2020.

[112] Shi-Yu Xia, Jiaqi Lv, Ning Xu, and Xin Geng. Ambiguity-induced contrastive learning for instance-dependent partial label learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 3615–3621, 2022.

[113] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. USB: A unified semi-supervised learning benchmark for classification. In *Advances in Neural Information Processing Systems*, pages 3938–3961, 2022.

[114] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, International Conference on Machine Learning*, pages 1–6, 2013.

[115] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019.

[116] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268, 2020.

[117] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A. Raffel, Ekin D. Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608, 2020.

[118] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. FreeMatch: Self-adaptive thresholding for semi-supervised learning. In *Eleventh International Conference on Learning Representations*, 2023.

[119] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1):180161, 2018.

[120] 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*, pages 2857–2866, 2025.

[121] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.

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

[123] 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*, pages 9390–9399, 2024.

[124] 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, pages 73582–73603, 2023.

Appendix

A Notation and Definitions **19**

B Implementation and Evaluation Protocols **20**

B.1 Weak-Annotation Construction 20

B.2 Model and Training Setup 20

B.3 Evaluation Metrics 20

C Theoretical Derivations and Proofs **21**

C.1 Connections among Different Diffusion Models. 21

C.2 Derivation of Eq. (5) 21

C.3 Proof of Theorem 4.1 22

C.4 Proof of Proposition 4.2 23

C.5 Lower-Bound Surrogate of the Classification Term 24

C.6 Proof of Theorem 4.3 25

C.7 Necessary Optimality Condition for the Selected Timestep Interval 26

C.8 Derivation of Eq. (2) 27

D Additional Method Analysis **29**

D.1 Gradient Analysis of the Noisy-Label Rectification in Eq. (10) 29

D.2 Class-Prior Estimation in Datasets with Weak Annotations 30

E Additional Experimental Results **32**

E.1 Full Results in Weakly Supervised Classification 32

E.2 Results under Coexisting Weak-Annotation Types 33

E.3 Results on a Domain-Realistic Medical Dataset 34

E.4 Ablation Results of Median-Centered Timestep Sampling 34

E.5 Sensitivity Analysis of the Sampling Interval Width 34

F Weakly Supervised Dataset Condensation **35**

F.1 Setup 35

F.2 Representative Results 35

F.3 Comprehensive IPC Analyses 35

F.4 Visualization of Weakly Supervised Condensed Datasets 35

A Notation and Definitions

Table 6 summarizes the main notation used throughout the paper.

Table 6: List of common mathematical symbols used in this paper.

Symbol	Definition
$\mathbf{x}$	Data sample
X	Random variable of data samples
y	Latent clean class label
Y	Random variable of clean class labels
z	Observed weak annotation
Z	Random variable of weak annotations
$\hat{y}$	Observed noisy label
$\hat{Y}$	Random variable of noisy labels
$\emptyset$	Missing-label symbol for unlabeled samples
$\mathcal{X}$	Input space
$\mathcal{Y}$	Class-label space
c	Number of classes
$\mathcal{D}$	Weakly annotated training dataset
θ	Current trainable LCDM parameters
ϕ	EMA copy of the LCDM parameters
$q(\mathbf{x})$	Real data density
$q_t(\mathbf{x}_t \mid y)$	Perturbed clean class-conditional density
$q(\mathbf{x}_t \mid \mathbf{x})$	Forward Gaussian perturbation distribution
$p_\theta(\cdot)$	Current class-posterior estimate
$p_\phi(\cdot)$	EMA class-posterior estimate
$\mathbf{x}_t$	Noisy latent state at timestep t
t	Diffusion timestep
T	Total number of diffusion timesteps
α_t	Signal scale in the forward process
σ_t	Noise scale in the forward process
w_t	Timestep-dependent weight in score matching
$\mathbf{s}_\theta(\mathbf{x}_t, y, t)$	Class-conditional score estimator
$\mathbf{h}_\theta(\mathbf{x}_\tau, y, \tau)$	Class-conditional denoised estimator
f_θ	Current diffusion-classifier posterior estimator
f_ϕ	EMA diffusion-classifier posterior estimator
$\mathcal{L}_{\text{Gen}}(\theta)$	Generative loss for posterior-weighted score matching
$\mathcal{L}_{\text{Cls}}(\theta)$	Classification loss for posterior refinement

B Implementation and Evaluation Protocols

This section details the experimental protocols used in the main paper, including weak-annotation construction, model training, and evaluation metrics. Our implementation is based on PyTorch 1.12 [77], and all experiments were conducted on NVIDIA Tesla A100 GPUs with CUDA 12.4.

B.1 Weak-Annotation Construction

For all class-dependent ambiguous-label datasets, we construct a 10×10 circulant transition matrix T where each row maps a true label to a candidate set of labels with varying probabilities. We set $q = 0.5$ and define the matrix as:

$$T = \begin{bmatrix} 1 & q{+}0.2 & q & \cdots & q{+}0.2 & q & q{-}0.2 \\ q{-}0.2 & 1 & q{+}0.2 & \cdots & q & q{-}0.2 & q{+}0.2 \\ q & q{-}0.2 & 1 & \cdots & q{-}0.2 & q{+}0.2 & q \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ q{+}0.2 & q & q{-}0.2 & \cdots & q & q{-}0.2 & 1 \end{bmatrix}.$$

For noisy-label datasets with asymmetric noise (40% flip probability), we adopt the following mappings: *Fashion-MNIST:* 'Pullover'→'Sneaker', 'Dress'→'Bag', 'Sandal'→'Shirt', 'Shirt'→'Sandal'. *CIFAR-10:* 'Truck'→'Automobile', 'Bird'→'Airplane', 'Deer'→'Horse', 'Cat'→'Dog', 'Dog'→'Cat'. *ImageNette:* 'Tench'→'English springer', 'Cassette player'→'Garbage truck', 'Chain saw'→'Church', 'Golf ball'→'Parachute', 'Parachute'→'Golf ball'.

B.2 Model and Training Setup

We build our framework upon the EDM formulation [41], utilizing the DDPM++ network architecture with a U-Net backbone [78] for all experiments. Optimization is performed using the Adam optimizer with a learning rate of 1×10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, and $\epsilon = 1 \times 10^{-8}$. We apply an Exponential Moving Average (EMA) decay of 0.5. The batch sizes are set to 128 for Fashion-MNIST, 64 for CIFAR-10, and 16 for ImageNette. Regarding the proposed diffusion classifier, the timestep interval length Δ is set to 6.4. All models are trained from scratch for a total of 200k iterations.

B.3 Evaluation Metrics

We evaluate the trained CDMs using four unconditional metrics, Fréchet Inception Distance (FID) [79], Inception Score (IS) [80], Density, and Coverage [81], together with three class-wise conditional metrics, CW-FID, CW-Density, and CW-Coverage [54]. All metrics are computed using the official implementation of DLSM [54].

Unconditional metrics. Unconditional metrics ignore class labels when comparing real and generated distributions. In our experiments, samples are generated conditionally per class and then pooled before evaluation, following prior conditional-generation protocols [55, 54]. FID measures feature-space distribution distance, IS reflects realism and class separability, and Density and Coverage are robust analogs of Precision and Recall.

Conditional metrics. To measure class-conditional consistency, we use class-wise variants of the above metrics. Each metric is computed within each class and then averaged across classes. CW-FID, also known as intra-FID, is a standard measure of conditional distribution quality [82, 55].

Remark. Fashion-MNIST is not suitable for these image-generation metrics, so we do not report FID, IS, Density, Coverage, or their class-wise variants on Fashion-MNIST.

C Theoretical Derivations and Proofs

This section collects the derivations and proofs supporting the main method. We first relate our diffusion notation to common formulations, then prove the score decomposition, the posterior-weighted score-matching property, the timestep-sampling rule, and the variational objectives used in training.

C.1 Connections among Different Diffusion Models.

The diffusion model used in this paper admits a flexible parameterization that aligns with common diffusion frameworks, including DDPM [1], SMLD [83], VE-SDE [2], and VP-SDE [2]. This compatibility allows us to analyze these frameworks through specific choices of the schedules α_t and σ_t.

DDPM. DDPM defines a sequence $\{\beta_t\}_{t=0}^T$ and $\mathbf{x}_t = \sqrt{\prod_{i=0}^t (1-\beta_i)}\mathbf{x}_0 + \sqrt{1-\prod_{i=0}^t (1-\beta_i)}\boldsymbol{\epsilon}$, which can be seen as a special case of Eq. (1) where we can set $\alpha_t = \sqrt{\prod_{i=0}^t (1-\beta_i)}$ and $\sigma_t = \sqrt{1-\prod_{i=0}^t (1-\beta_i)}$.

SMLD. SMLD defines a noise schedule $\sigma(t)_{t=0}^T$ and $\mathbf{x}_t = \mathbf{x}_0 + \sigma(t)\boldsymbol{\epsilon}$, with $\sigma(1) < \sigma(2) < \cdots < \sigma(T)$. In this scenario, Eq. (1) reduces to $\alpha_t = 1$, $\sigma_t = \sigma(t)$.

VP-SDE. VP-SDE is the continuous-time counterpart of DDPM and defines the stochastic differential equation

$$dX_t = -\frac{1}{2}\beta(t)X_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}W_t, \quad t \in [0,1],$$

where $\beta(t) = \beta_{t\cdot T}\cdot T$. In this setup, $\alpha_t = \sqrt{\exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right)}$, and $\sigma_t = \sqrt{1-\exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right)}$.

VE-SDE. VE-SDE is the continuous-time counterpart of SMLD, with forward process

$$dX_t = \sqrt{\frac{\mathrm{d}\sigma(t)^2}{\mathrm{d}t}}\,\mathrm{d}W_t.$$

In this setup, $\alpha_t = 1$ and $\sigma_t = \sqrt{\sigma^2(t) - \sigma^2(0)}$.

Weighting function w_t. Different frameworks typically adopt specific prediction targets (e.g., ϵ-prediction in DDPM or score-prediction in SMLD), which implicitly corresponds to the usage of specific weighting functions w_t in the score matching objective. For instance, when the model is trained to predict $\boldsymbol{\epsilon}$, the resulting objective is equivalent to a score-matching loss with weight $w_t \propto \sigma_t^2$; training to predict $\mathbf{x}_0$ instead induces a different rescaling, and hence a different effective weighting. While the models above each define a specific framework, EDM [41] provides a unified structure and optimizes these choices, including network input and output preconditioning, for stable training. We therefore adopt EDM as our default implementation framework. We provide the conversion logic in Algorithm 1 to demonstrate how our score network $\mathbf{s}_\theta$ adapts to different prediction targets ($\mathbf{x}_0$, $\boldsymbol{\epsilon}$, or $\boldsymbol{v}$), thereby effectively recovering the distinct loss weightings used in prior works.

C.2 Derivation of Eq. (5)

To maximize the log-likelihood $\log p_\theta(X, Z)$, we introduce an approximate posterior distribution $p_\phi(Y \mid X, Z)$ over the latent variable Y. Applying Jensen's inequality, we derive the ELBO as

follows:

$$\begin{aligned}\log p_\theta(X,Z) &= \log \sum_{Y\in\mathcal{Y}} p_\theta(X,Y,Z) \\ &= \log \sum_{Y\in\mathcal{Y}} p_\phi(Y\,|\,X,Z)\frac{p_\theta(X,Y,Z)}{p_\phi(Y\,|\,X,Z)} \\ &\geq \sum_{Y\in\mathcal{Y}} p_\phi(Y\,|\,X,Z)\log\frac{p_\theta(X,Y,Z)}{p_\phi(Y\,|\,X,Z)} \\ &= \mathbb{E}_{p_\phi(Y|X,Z)}\Big[\log p_\theta(X,Y,Z) - \log p_\phi(Y\,|\,X,Z)\Big].\end{aligned}$$

Since the entropy term $-\mathbb{E}_{p_\phi}[\log p_\phi]$ does not depend on θ, maximizing the ELBO w.r.t. θ simplifies to

$$\arg\max_\theta \ \mathbb{E}_{p_\phi(Y|X,Z)}\big[\log p_\theta(X,Y,Z)\big].$$

Using the chain rule of probability, we can decompose the log-likelihood as

$$\log p_\theta(X,Y,Z) = \log p_\theta(X\,|\,Z) + \log p_\theta(Y\,|\,X,Z) + \log p_\theta(Z).$$

Substituting this back into the expectation:

$$\begin{aligned}\theta^* &= \arg\max_\theta \ \mathbb{E}_{p_\phi(Y|X,Z)}\Big[\log p_\theta(X\,|\,Z) + \log p_\theta(Y\,|\,X,Z) + \log p_\theta(Z)\Big] \\ &= \arg\max_\theta \ \Big\{\mathbb{E}_{p_\phi(Y|X,Z)}\big[\log p_\theta(X\,|\,Z)\big] + \mathbb{E}_{p_\phi(Y|X,Z)}\big[\log p_\theta(Y\,|\,X,Z)\big]\Big\}.\end{aligned}$$

The term $\log p_\theta(Z)$ is constant w.r.t. the optimization and thus omitted, yielding the objective in Eq. (5).

C.3 Proof of Theorem 4.1

The derivation is analogous to Theorem 1 in Na et al. [12], but we rewrite it here using the notation and assumptions of the current paper for completeness. Under the class-conditional independence assumption $X \perp Z \mid Y$, the perturbed variable $\mathbf{x}_t$ also satisfies $\mathbf{x}_t \perp Z \mid Y$, because $\mathbf{x}_t$ is generated from X through the forward diffusion process and therefore does not depend on Z once Y is fixed. Hence

$$q_t(\mathbf{x}_t \mid y, z) = q_t(\mathbf{x}_t \mid y).$$

Marginalizing over the latent clean label Y gives

$$q_t(\mathbf{x}_t \mid z) = \sum_{y=1}^{c} q_t(\mathbf{x}_t, y \mid z) = \sum_{y=1}^{c} q_t(\mathbf{x}_t \mid y, z)\, p(y \mid z) = \sum_{y=1}^{c} q_t(\mathbf{x}_t \mid y)\, p(y \mid z).$$

Algorithm 1 Formulation Conversion (Our Model to EDM)

Require: A score network $\mathbf{s}_\theta$, a noisy input $\mathbf{x}_t$, timestep t, and forward-process schedules $\{\alpha_i\}_{i=1}^T$ and $\{\sigma_i\}_{i=1}^T$.

Estimate the clean sample using $\mathbf{s}_\theta$: $\hat{\mathbf{x}}_0 = (\mathbf{x}_t + \sigma_t^2 \mathbf{s}_\theta(\mathbf{x}_t, t))/\alpha_t$
if performing $\mathbf{x}_0$-prediction **then**
    Return $\hat{\mathbf{x}}_0$.
end if
Estimate the noise component: $\hat{\boldsymbol{\epsilon}} = \frac{\mathbf{x}_t - \alpha_t \hat{\mathbf{x}}_0}{\sigma_t}$
if performing $\boldsymbol{\epsilon}$-prediction **then**
    Return $\hat{\boldsymbol{\epsilon}}$.
end if
Estimate the velocity component: $\hat{\boldsymbol{v}} = \alpha_t \hat{\boldsymbol{\epsilon}} - \sigma_t \hat{\mathbf{x}}_0$
if performing $\boldsymbol{v}$-prediction **then**
    Return $\hat{\boldsymbol{v}}$.
end if

Taking the gradient of the log-density, we obtain

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z) = \frac{\sum_{y=1}^{c} p(y \mid z)\, \nabla_{\mathbf{x}_t} q_t(\mathbf{x}_t \mid y)}{q_t(\mathbf{x}_t \mid z)}.$$

Multiplying and dividing each term by $q_t(\mathbf{x}_t \mid y)$ yields

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z) = \sum_{y=1}^{c} \frac{p(y \mid z)\, q_t(\mathbf{x}_t \mid y)}{q_t(\mathbf{x}_t \mid z)} \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y).$$

Now apply Bayes' rule:

$$p(y \mid \mathbf{x}_t, z) = \frac{q_t(\mathbf{x}_t \mid y, z)\, p(y \mid z)}{q_t(\mathbf{x}_t \mid z)} = \frac{q_t(\mathbf{x}_t \mid y)\, p(y \mid z)}{q_t(\mathbf{x}_t \mid z)},$$

where the second equality again uses $\mathbf{x}_t \perp Z \mid Y$. Therefore,

$$\frac{p(y \mid z)\, q_t(\mathbf{x}_t \mid y)}{q_t(\mathbf{x}_t \mid z)} = p(y \mid \mathbf{x}_t, z).$$

Substituting this identity into the previous display gives Eq. (6):

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z) = \sum_{y=1}^{c} p(y \mid \mathbf{x}_t, z)\, \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y).$$

This completes the proof.

C.4 Proof of Proposition 4.2

Assume that the posterior estimate is exact, i.e., $p_\phi(y \mid \mathbf{x}_t, z) = p(y \mid \mathbf{x}_t, z)$. Define the posterior-weighted aggregation of the class-wise scores as

$$g_\theta(\mathbf{x}_t, z, t) := \sum_{y=1}^{c} p(y \mid \mathbf{x}_t, z)\, \mathbf{s}_\theta(\mathbf{x}_t, y, t).$$

With this notation, Eq. (7) is exactly a denoising score matching objective for the weak-conditional distribution:

$$\mathcal{L}_{\mathrm{DSM}}(\theta) := \mathbb{E}_{t,z}\Big[w_t\, \mathbb{E}_{\mathbf{x}_t, \mathbf{x} \sim q(\mathbf{x}_t, \mathbf{x} \mid z)} \big\| g_\theta(\mathbf{x}_t, z, t) - a(\mathbf{x}_t, \mathbf{x}, z) \big\|_2^2 \Big],$$

where

$$a(\mathbf{x}_t, \mathbf{x}, z) := \nabla_{\mathbf{x}_t} \log q_{t|0}(\mathbf{x}_t \mid \mathbf{x}, z).$$

Let

$$b(\mathbf{x}_t, z) := \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z)$$

be the explicit weak-conditional score. By the standard DSM identity,

$$\mathbb{E}_{\mathbf{x} \sim q(\mathbf{x} \mid \mathbf{x}_t, z)} \big[a(\mathbf{x}_t, \mathbf{x}, z) \big] = b(\mathbf{x}_t, z).$$

Therefore, conditioning on $(\mathbf{x}_t, z)$ gives the bias–variance decomposition

$$\mathbb{E}_{\mathbf{x} \sim q(\mathbf{x} \mid \mathbf{x}_t, z)} \big\| g_\theta(\mathbf{x}_t, z, t) - a(\mathbf{x}_t, \mathbf{x}, z) \big\|_2^2 = \big\| g_\theta(\mathbf{x}_t, z, t) - b(\mathbf{x}_t, z) \big\|_2^2 + C(\mathbf{x}_t, z),$$

where $C(\mathbf{x}_t, z)$ is independent of θ. Hence $\mathcal{L}_{\mathrm{DSM}}(\theta)$ differs from the explicit score matching objective

$$\mathcal{L}_{\mathrm{ESM}}(\theta) := \mathbb{E}_{t,z}\Big[w_t\, \mathbb{E}_{\mathbf{x}_t \sim q(\mathbf{x}_t \mid z)} \big\| g_\theta(\mathbf{x}_t, z, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z) \big\|_2^2 \Big]$$

only by a constant independent of θ. Thus the two objectives share the same minimizers, and any optimal aggregated predictor satisfies

$$g_{\theta^*}(\mathbf{x}_t, z, t) = \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z).$$

By Theorem 4.1, this weak-conditional score decomposes as

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid z) = \sum_{y=1}^{c} p(y \mid \mathbf{x}_t, z)\, \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y).$$

Combining the last two displays with the definition of g_{θ^*} gives

$$\sum_{y=1}^{c} p(y \mid \mathbf{x}_t, z)\, \mathbf{s}_{\theta^*}(\mathbf{x}_t, y, t) = \sum_{y=1}^{c} p(y \mid \mathbf{x}_t, z)\, \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y).$$

Equivalently,

$$\sum_{y=1}^{c} p(y \mid \mathbf{x}_t, z) \Big(\mathbf{s}_{\theta^*}(\mathbf{x}_t, y, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y) \Big) = 0.$$

In the identifiable class-wise solution of this system, the residual for each class vanishes, yielding

$$\mathbf{s}_{\theta^*}(\mathbf{x}_t, y, t) = \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid y), \qquad \forall y \in \mathcal{Y}.$$

This is precisely Eq. (8) and proves the proposition.

C.5 Lower-Bound Surrogate of the Classification Term

Section 4.3 replaces the exact *Classification Term* of Eq. (5) with a tractable lower-bound surrogate. We justify this substitution below and discuss the boundedness of the dropped term.

Under the class-conditional assumption ($X \perp Z \mid Y$), Bayes' rule gives

$$p_\theta(y \,|\, \mathbf{x}, z) = \frac{p(z \,|\, y)\, p_\theta(y \,|\, \mathbf{x})}{p_\theta(z \,|\, \mathbf{x})}, \quad p_\theta(z \,|\, \mathbf{x}) = \sum_{y' \in \mathcal{Y}} p(z \,|\, y')\, p_\theta(y' \,|\, \mathbf{x}). \tag{13}$$

Substituting this factorization into the exact *Classification Term* yields

$$\begin{aligned} &\sum_{y \in \mathcal{Y}} p_\phi(y \,|\, \mathbf{x}, z)\, \log p_\theta(y \,|\, \mathbf{x}, z) \\ &= \sum_{y \in \mathcal{Y}} p_\phi(y \,|\, \mathbf{x}, z)\, \log p_\theta(y \,|\, \mathbf{x}) + \underbrace{\sum_{y \in \mathcal{Y}} p_\phi(y \,|\, \mathbf{x}, z)\, \log p(z \,|\, y)}_{\theta\text{-independent}} - \log p_\theta(z \,|\, \mathbf{x}). \end{aligned}$$

The middle term is independent of θ. Moreover, since $p_\theta(z \,|\, \mathbf{x}) \in (0, 1]$, we have $\log p_\theta(z \,|\, \mathbf{x}) \leq 0$, and hence

$$\sum_{y \in \mathcal{Y}} p_\phi(y \,|\, \mathbf{x}, z)\, \log p_\theta(y \,|\, \mathbf{x}) + \text{const} \leq \sum_{y \in \mathcal{Y}} p_\phi(y \,|\, \mathbf{x}, z)\, \log p_\theta(y \,|\, \mathbf{x}) - \log p_\theta(z \,|\, \mathbf{x}) + \text{const}. \tag{14}$$

Equivalently, after removing the θ-independent constant, maximizing

$$\sum_{y \in \mathcal{Y}} p_\phi(y \,|\, \mathbf{x}, z)\, \log p_\theta(y \,|\, \mathbf{x}) \tag{15}$$

maximizes a tractable lower-bound surrogate of the exact *Classification Term*.

Boundedness of the dropped term. The dropped term $\log p_\theta(z \,|\, \mathbf{x})$ is bounded both above and below in all three weak-annotation scenarios, ensuring that the surrogate gap is finite and the gradient $\nabla_\theta \log p_\theta(z \,|\, \mathbf{x})$ remains bounded. Since $p_\theta(y \,|\, \mathbf{x})$ is a probability distribution over y, the marginal admits the lower bound

$$p_\theta(z \,|\, \mathbf{x}) = \sum_{y' \in \mathcal{Y}} p(z \,|\, y')\, p_\theta(y' \,|\, \mathbf{x}) \;\geq\; \min_{y' \in \mathcal{Y}} p(z \,|\, y') \cdot \sum_{y \in \mathcal{Y}} p_\theta(y \,|\, \mathbf{x}) = \min_{y' \in \mathcal{Y}} p(z \,|\, y'), \tag{16}$$

which we instantiate per scenario:

- *Noisy labels* ($z = \hat{y}$). With transition matrix $T_{y,\hat{y}} = p(\hat{y} \,|\, y)$, $\log p_\theta(z \,|\, \mathbf{x}) \geq \log \min_{y'} T_{y',\hat{y}}$. In practice this minimum is moderate: e.g., it equals 0.01 in the realistic CIFAR-10 benchmark of [1] ($\log p_\theta(z \,|\, \mathbf{x}) \geq -4.61$) and 0.4 in our experimental setting ($\log p_\theta(z \,|\, \mathbf{x}) \geq -0.92$).

- *Ambiguous labels* ($z \subseteq \mathcal{Y}$). $\log p_\theta(z \mid \mathbf{x}) \geq \log \min_{y'} p(z \mid y')$, which is also moderate in practice: e.g., it equals 0.121 in the realistic benchmark of [2] ($\log p_\theta(z \mid \mathbf{x}) \geq -2.11$) and 0.5 / 0.3 in our random ($\log p_\theta(z \mid \mathbf{x}) \geq -0.69$) / class-dependent ($\log p_\theta(z \mid \mathbf{x}) \geq -1.20$) settings.
- *Semi-supervised data* ($z \in \mathcal{Y} \cup \{\emptyset\}$). With labeling rate $\eta = 0.1$, unlabeled samples have $p(\emptyset \mid y) = 1 - \eta$, giving $\log p_\theta(z \mid \mathbf{x}) \geq -0.11$.

In all cases, $0 \geq \log p_\theta(z \mid \mathbf{x}) \geq \log \min_{y'} p(z \mid y') > -\infty$, so its gradient $\nabla_\theta \log p_\theta(z \mid \mathbf{x})$ remains bounded throughout training. The surrogate gap further shrinks as $p_\theta(y \mid \mathbf{x})$ concentrates on the correct class.

C.6 Proof of Theorem 4.3

We define the median-centered surrogate by choosing a subinterval $[l, r]$ whose excluded weight mass is balanced around the median τ_{mid} of the weight distribution $w(\tau)$. This construction targets the informative intermediate-noise region while remaining independent of the data-dependent reconstruction error $\hbar(\tau, y)$.

Case 1: Log-Normal Distribution (EDM Framework). In the EDM formulation [41], the weighting function $w(\tau)$ is designed to peak at intermediate noise levels, effectively inducing a distribution over timesteps. Specifically, the weight density follows a log-normal shape:

$$w(\tau) \propto \frac{1}{\tau} \exp\left(-\frac{(\ln \tau - P_{\text{mean}})^2}{2P_{\text{std}}^2}\right),$$

with $P_{\text{mean}} = -1.2$ and $P_{\text{std}} = 1.2$. Normalizing this weight function yields a probability density function $p_w(\tau) = w(\tau)/Z$, which corresponds to the PDF of a log-normal distribution for τ. The normalized cumulative weight mass is given by

$$\mathcal{F}_w(\tau) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\ln \tau - P_{\text{mean}}}{P_{\text{std}}\sqrt{2}}\right)\right],$$

where $\operatorname{erf}(x)$ denotes the error function.

Let τ_{mid} be the median of this weight distribution, satisfying $\mathcal{F}_w(\tau_{\text{mid}}) = 0.5$. To construct a median-centered interval of length Δ, we impose that the cumulative weight mass excluded on the left equals that excluded on the right:

$$\mathcal{F}_w(l) = 1 - \mathcal{F}_w(r) = 1 - \mathcal{F}_w(l + \Delta).$$

Rearranging terms yields the implicit equation for the median-centered start time l:

$$\mathcal{F}_w(l) + \mathcal{F}_w(l + \Delta) = 1. \tag{17}$$

Given the monotonicity of the CDF, this equation has a unique solution for l, which we compute numerically (e.g., using Brent's method [49]). The selected interval is then $[l, l + \Delta]$.

Case 2: Uniform Weighting (DDPM Framework). In standard DDPM [1], discrete timesteps are often weighted equally in simplified objectives, implying a constant weighting function $w(\tau) = \text{const}$. After normalizing the timestep range to $[0, 1]$, the corresponding cumulative weight function is linear: $\mathcal{F}_w(\tau) = \tau$. Applying the same symmetry condition from Eq. (17):

$$\begin{aligned} l + (l + \Delta) &= 1 \\ 2l &= 1 - \Delta \\ l &= \frac{1 - \Delta}{2}. \end{aligned}$$

Consequently, the canonical median-centered interval becomes $[\frac{1-\Delta}{2}, \frac{1+\Delta}{2}]$, which is symmetric around the midpoint 0.5. Since the weight is constant in this case, all intervals of the same length have the same weight mass; the centered interval is chosen as the symmetry-preserving representative.

Conclusion. In both cases, the construction gives an interval with equal omitted weight mass on the two sides of the weight median. As $\Delta \to 0$, the interval converges to the median of the weight distribution, identifying the single timestep selected by the median-centered surrogate. This is consistent with empirical findings that useful class information in diffusion classifiers concentrates around intermediate noise levels [34, 37, 84–87].

C.7 Necessary Optimality Condition for the Selected Timestep Interval

In this subsection, we define the class-wise residual $\mathrm{ERR}(y)$ used in Figure 2(c).

Theorem C.1 (Necessary optimality condition)**.** *A necessary condition for the optimal subinterval is that the expected error within the interval equals the average error at the boundaries:*

$$\int_{l^*}^{r^*} \frac{w_\tau \hbar(\tau, y)}{W_{[l^*,r^*]}} \,\mathrm{d}\tau = \frac{\hbar(l^*, y) + \hbar(r^*, y)}{2}. \tag{18}$$

Definition of the $\textsc{Err}(y)$. Theorem C.1 motivates the following class-wise optimality residual:

$$\textsc{Err}(y) := \frac{1}{W_{[l,r]}} \int_l^r w(\tau)\hbar(\tau, y)\, d\tau - \frac{\hbar(l, y) + \hbar(r, y)}{2}. \tag{19}$$

For Figure 2(c), we compute Eq. (19) over evaluation samples and report the resulting distribution for each class y.

Proof of Theorem C.1. For notational conciseness, we omit the class dependency y in the reconstruction error, denoting it simply as $\hbar(\tau)$, and define $w(\tau)$ as the weighting function (referring to w_τ in the main text).

1. Objective Formulation. Define the weight mass $W_{[l,r]}$ and weighted error $H_{[l,r]}$ on $[l, r]$ as:

$$W_{[l,r]} := \int_l^r w(\tau)\,\mathrm{d}\tau, \quad H_{[l,r]} := \int_l^r w(\tau)\hbar(\tau)\,\mathrm{d}\tau.$$

Let $\mu_{[l,r]} = H_{[l,r]}/W_{[l,r]}$ and $\mu_{\mathcal{T}}$ denote the expected errors on $[l, r]$ and the effective range $\mathcal{T} = [u(t + 1), u(T - 1)]$, respectively. We minimize their squared difference subject to $W_{[l,r]} = C$:

$$\min_{l,r} \quad \mathcal{J}(l, r) := (\mu_{[l,r]} - \mu_{\mathcal{T}})^2, \quad \text{s.t.} \quad W_{[l,r]} - C = 0. \tag{20}$$

2. Optimization via Lagrange Multipliers. We construct the Lagrangian $\mathcal{L}(l, r, \lambda)$:

$$\mathcal{L}(l, r, \lambda) = (\mu_{[l,r]} - \mu_{\mathcal{T}})^2 + \lambda(W_{[l,r]} - C).$$

Applying the Leibniz integral rule, the partial derivatives w.r.t. the interval boundaries are

$$\frac{\partial H}{\partial l} = -w(l)\hbar(l), \quad \frac{\partial H}{\partial r} = w(r)\hbar(r), \quad \frac{\partial W}{\partial l} = -w(l), \quad \frac{\partial W}{\partial r} = w(r).$$

Differentiating $\mu_{[l,r]} = H_{[l,r]}/W_{[l,r]}$ yields

$$\frac{\partial \mu_{[l,r]}}{\partial l} = \frac{w(l)}{W_{[l,r]}}(\mu_{[l,r]} - \hbar(l)), \quad \frac{\partial \mu_{[l,r]}}{\partial r} = \frac{w(r)}{W_{[l,r]}}(\hbar(r) - \mu_{[l,r]}).$$

Setting the gradients of the Lagrangian $\nabla\mathcal{L}$ to zero yields the first-order necessary conditions for the optimal interval $[l^*, r^*]$:

$$\left.\frac{\partial \mathcal{L}}{\partial l}\right|_{l^*} = 2(\mu_{[l^*,r^*]} - \mu_{\mathcal{T}}) \cdot \frac{w(l^*)}{W}(\mu_{[l^*,r^*]} - \hbar(l^*)) - \lambda w(l^*) = 0,$$
$$\left.\frac{\partial \mathcal{L}}{\partial r}\right|_{r^*} = 2(\mu_{[l^*,r^*]} - \mu_{\mathcal{T}}) \cdot \frac{w(r^*)}{W}(\hbar(r^*) - \mu_{[l^*,r^*]}) + \lambda w(r^*) = 0$$

Because $w(l^*), w(r^*) > 0$, we can divide by $w(\cdot)$ and equate the terms involving λ:

$$\frac{2}{W}(\mu_{[l^*,r^*]} - \mu_{\mathcal{T}})(\mu_{[l^*,r^*]} - \hbar(l^*)) = \lambda = \frac{2}{W}(\mu_{[l^*,r^*]} - \mu_{\mathcal{T}})(\hbar(r^*) - \mu_{[l^*,r^*]}).$$

We then divide by the common factor $\frac{2}{W}(\mu_{[l^*,r^*]} - \mu_{\mathcal{T}})$ to obtain:

$$\mu_{[l^*,r^*]} - \hbar(l^*) = \hbar(r^*) - \mu_{[l^*,r^*]} \quad \Longrightarrow \quad \mu_{[l^*,r^*]} = \frac{\hbar(l^*) + \hbar(r^*)}{2}.$$

C.8 Derivation of Eq. (2)

Maximizing the evidence lower bound (ELBO) is equivalent to minimizing the weighted score matching objective under a specific weighting function w_t. Adopting the derivation from Eq. (58) in [88], we have

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_q[-D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T)) + \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) - \sum_{t>1} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]$$

Leveraging the Gaussian transition assumption, we can evaluate the KL divergence term by Bayes rule:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

Using Eq. (1), we derive the transition distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1})$:

$$\begin{aligned}
\mathbf{x}_t &= \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}_0 \\
&= \alpha_t \left(\frac{\mathbf{x}_{t-1} - \sigma_{t-1}\boldsymbol{\epsilon}_0^*}{\alpha_{t-1}}\right) + \sigma_t \boldsymbol{\epsilon}_0 \\
&= \frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_{t-1} + \sigma_t \boldsymbol{\epsilon}_0 - \frac{\alpha_t}{\alpha_{t-1}}\sigma_{t-1}\boldsymbol{\epsilon}_0^* \\
&= \frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_{t-1} + \sqrt{\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2}\boldsymbol{\epsilon}_{t-1} \\
&= \mathcal{N}(\mathbf{x}_t; \frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_{t-1}, \sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2\mathbf{I}).
\end{aligned}$$

Now, we can proceed to calculate the form of $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ by substituting into the Bayes rule expansion:

$$\begin{aligned}
&q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&= \frac{\mathcal{N}(\mathbf{x}_t; \frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_{t-1}, \sqrt{\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2}\mathbf{I})\mathcal{N}(\mathbf{x}_{t-1}; \alpha_{t-1}\mathbf{x}_0, \sigma_{t-1}\mathbf{I})}{\mathcal{N}(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t\mathbf{I})} \\
&\propto \exp\left\{-\left[\frac{(\mathbf{x}_t - \frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_{t-1})^2}{2(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)} + \frac{(\mathbf{x}_{t-1} - \alpha_{t-1}\mathbf{x}_0)^2}{2\sigma_{t-1}^2} - \frac{(\mathbf{x}_t - \alpha_t\mathbf{x}_0)^2}{2\sigma_t^2}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left[\frac{-2\frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_t\mathbf{x}_{t-1} + (\frac{\alpha_t}{\alpha_{t-1}})^2\mathbf{x}_{t-1}^2}{\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2} + \frac{\mathbf{x}_{t-1}^2 - 2\alpha_{t-1}\mathbf{x}_{t-1}\mathbf{x}_0}{\sigma_{t-1}^2} + C(\mathbf{x}_t, \mathbf{x}_0)\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\left(\frac{(\frac{\alpha_t}{\alpha_{t-1}})^2}{\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2} + \frac{1}{\sigma_{t-1}^2}\right)\mathbf{x}_{t-1}^2 - 2\left(\frac{\frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_t}{\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2} + \frac{\alpha_{t-1}\mathbf{x}_0}{\sigma_{t-1}^2}\right)\mathbf{x}_{t-1}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left[\frac{\sigma_{t-1}^2(\frac{\alpha_t}{\alpha_{t-1}})^2 + (\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)}{(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)\sigma_{t-1}^2}\mathbf{x}_{t-1}^2 - 2\left(\frac{\frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_t}{\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2} + \frac{\alpha_{t-1}\mathbf{x}_0}{\sigma_{t-1}^2}\right)\mathbf{x}_{t-1}\right]\right\} \\
&\propto \mathcal{N}(\mathbf{x}_{t-1}; \frac{\frac{\alpha_t}{\alpha_{t-1}}\mathbf{x}_t\sigma_{t-1}^2 + (\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)\alpha_{t-1}\mathbf{x}_0}{\sigma_t^2}, \frac{(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)\sigma_{t-1}^2}{\sigma_t^2}\mathbf{I})
\end{aligned}$$

where in the fourth equation, $C(\mathbf{x}_t, \mathbf{x}_0)$ is constant w.r.t. $\mathbf{x}_{t-1}$ and depends only on $\mathbf{x}_t$, $\mathbf{x}_0$, and the α values. We have therefore shown that at each step, $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is normally distributed, with mean $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ that is a function of $\mathbf{x}_t$ and $\mathbf{x}_0$, and variance $\boldsymbol{\Sigma}_q(t)$ as a function of α and σ coefficients.

If we set the variances of the two Gaussians to match, optimizing the KL divergence term reduces to minimizing the difference between their means:

$$\begin{aligned} &\arg\min_\theta D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \\ =&\arg\min_\theta \frac{1}{2\sigma_q^2(t)} \left[\left\| \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) \right\|_2^2 \right], \end{aligned} \tag{21}$$

where $\sigma_q^2(t) = \frac{(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)\sigma_{t-1}^2}{\sigma_t^2}$. The derivation follows Eq. (92) in [88]. To obtain the score-matching form, we use Tweedie's formula [89], which states $\mathbb{E}[\boldsymbol{\mu}_z|z] = z + \boldsymbol{\Sigma}_z \nabla_z \log q(z)$ for a Gaussian variable $z \sim \mathcal{N}(z; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$. In this case, we apply it to predict the true posterior mean of $\mathbf{x}_t$ given its samples. We can obtain

$$\begin{aligned} \mathbb{E}[\boldsymbol{\mu}_{\mathbf{x}_t}|\mathbf{x}_t] = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) &= \alpha_t \mathbf{x}_0 \\ \therefore \mathbf{x}_0 &= \frac{\mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)}{\alpha_t} \end{aligned} \tag{22}$$

Then, we can plug Eq. (22) into our ground-truth denoising transition mean $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ once again and derive a new form:

$$\begin{aligned} \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) &= \frac{\frac{\alpha_t}{\alpha_{t-1}}\sigma_{t-1}^2 \mathbf{x}_t + (\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)\alpha_{t-1} \cdot \frac{\mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)}{\alpha_t}}{\sigma_t^2} \\ &= \frac{\frac{\alpha_t}{\alpha_{t-1}}\sigma_{t-1}^2}{\sigma_t^2}\mathbf{x}_t + \frac{\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2}{\sigma_t^2 \frac{\alpha_t}{\alpha_{t-1}}}\mathbf{x}_t + \frac{(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)\sigma_t^2 \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)}{\sigma_t^2 \frac{\alpha_t}{\alpha_{t-1}}} \\ &= \frac{\alpha_{t-1}}{\alpha_t}\mathbf{x}_t + \left(\frac{\alpha_{t-1}}{\alpha_t}\sigma_t^2 - \frac{\alpha_t}{\alpha_{t-1}}\sigma_{t-1}^2 \right) \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) \\ &= \frac{\alpha_{t-1}}{\alpha_t} \left[\mathbf{x}_t + \left(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2 \right) \mathbf{s}_\theta(\mathbf{x}_t, t) \right] \end{aligned} \tag{23}$$

Finally, substituting Eq. (23) into the optimization objective in Eq. (21) gives

$$\begin{aligned} &\arg\min_\theta D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \\ =&\arg\min_\theta \frac{1}{2\sigma_q^2(t)} \left[\left\| \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) \right\|_2^2 \right] \\ =&\arg\min_\theta \frac{1}{2\frac{(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)\sigma_{t-1}^2}{\sigma_t^2}} \cdot (\frac{\alpha_{t-1}}{\alpha_t})^2 \cdot (\sigma_t^2 - \frac{\alpha_t^2}{\alpha_{t-1}^2}\sigma_{t-1}^2)^2 \|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)\|_2^2 \\ =&\frac{\sigma_t^2}{2}(\frac{\sigma_t^2 \alpha_{t-1}^2}{\sigma_{t-1}^2 \alpha_t^2} - 1)\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)\|_2^2 \end{aligned}$$

This result recovers the exact form of Eq. (2).

D Additional Method Analysis

This section provides additional analysis supporting the design choices in *DELTA*. We first analyze the noisy-label regularizer in Eq. (10) and discuss how class priors can be estimated when they are not directly observed.

D.1 Gradient Analysis of the Noisy-Label Rectification in Eq. (10)

We analyze the noisy-label branch of the classification objective in Eq. (10). For noisy labels, the weak annotation is $z = \hat{y}$. Let $\hat{\mathbf{y}} \in \{0,1\}^c$ denote the one-hot vector associated with the observed noisy label $\hat{y}$, and let $f_\theta(x), f_\phi(x) \in \Delta^{c-1}$ denote the current and EMA diffusion-classifier posteriors, respectively. For compactness, define

$$\delta(x) := \langle f_\theta(x), f_\phi(x) \rangle . \tag{24}$$

For finite logits and $c > 1$, we have $\delta(x) < 1$.

The noisy-label rectification used in Eq. (10) can be understood as a gradient-level implementation of early-learning regularization. To make the connection explicit, define the early-learning regularization [46] style correction vector

$$g_{\theta,\phi}(x) := \frac{f_\theta(x) \odot (\delta(x)\mathbf{1} - f_\phi(x))}{1 - \delta(x)}, \tag{25}$$

where $\mathbf{1} \in \mathbb{R}^c$ is the all-one vector and $\odot$ denotes the Hadamard product. The noisy-label branch of Eq. (10) corresponds to the fixed training target

$$\tilde{q}^{\mathrm{NL}}_{\phi,\lambda}(x, \hat{y}) := \mathrm{sg}(\hat{\mathbf{y}} - \lambda\, g_{\theta,\phi}(x)) , \tag{26}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. The main-text form is recovered by setting $\lambda = 1$; keeping $\lambda \geq 0$ in this appendix makes the connection to the original ELR strength explicit.

For a noisy-label dataset $D_{\mathrm{NL}} = \{(x^{(i)}, \hat{y}^{(i)})\}_{i=1}^n$, the corresponding classification loss is

$$L^{\mathrm{NL}}_{\mathrm{Cls}}(\theta) := -\frac{1}{n} \sum_{i=1}^{n} \left\langle \tilde{q}^{\mathrm{NL}}_{\phi,\lambda}(x^{(i)}, \hat{y}^{(i)}), \log f_\theta(x^{(i)}) \right\rangle . \tag{27}$$

The purpose of this target is to realize the ELR correction in the logit gradient.

Lemma D.1 (Logit-gradient equivalence to ELR)**.** *Let $\psi_\theta(x)$ denote the pre-softmax logits such that $f_\theta(x) = \mathrm{softmax}(\psi_\theta(x))$. For each sample $(x^{(i)}, \hat{y}^{(i)})$, define*

$$\delta^{(i)} := \left\langle f_\theta(x^{(i)}), f_\phi(x^{(i)}) \right\rangle , \qquad g^{(i)} := g_{\theta,\phi}(x^{(i)}).$$

Then the per-sample loss

$$\ell^{(i)}_{\mathrm{NL}} := -\left\langle \mathrm{sg}\left(\hat{\mathbf{y}}^{(i)} - \lambda g^{(i)}\right), \log f_\theta(x^{(i)}) \right\rangle$$

satisfies

$$\frac{\partial \ell^{(i)}_{\mathrm{NL}}}{\partial \psi_\theta(x^{(i)})} = f_\theta(x^{(i)}) - \hat{\mathbf{y}}^{(i)} + \lambda g^{(i)}, \tag{28}$$

where $f_\phi(x^{(i)})$ is treated as fixed during the current update.

Proof. We first show that $g_{\theta,\phi}(x)$ has zero total mass. Let $\delta = \langle f_\theta(x), f_\phi(x) \rangle$. Then

$$\mathbf{1}^\top g_{\theta,\phi}(x) = \frac{1}{1-\delta} \sum_{y \in Y} [f_\theta(x)]_y \big(\delta - [f_\phi(x)]_y\big) = \frac{\delta - \langle f_\theta(x), f_\phi(x) \rangle}{1 - \delta} = 0. \tag{29}$$

Therefore,

$$\mathbf{1}^\top \tilde{q}^{\mathrm{NL}}_{\phi,\lambda}(x, \hat{y}) = \mathbf{1}^\top \hat{\mathbf{y}} - \lambda\, \mathbf{1}^\top g_{\theta,\phi}(x) = 1.$$

For any fixed vector $a \in \mathbb{R}^c$ with $\mathbf{1}^\top a = 1$, the softmax identity gives

$$\frac{\partial}{\partial \psi} \left[- \langle a, \log \mathrm{softmax}(\psi) \rangle \right] = \mathrm{softmax}(\psi) - a. \tag{30}$$

Applying this identity with $a = \mathrm{sg}(\hat{\mathbf{y}}^{(i)} - \lambda g^{(i)})$ yields Eq. (28).

□

Interpretation. For each class $y \in Y$, the correction term can be written as

$$g_y^{(i)} = \frac{[f_\theta(x^{(i)})]_y \left(\delta^{(i)} - [f_\phi(x^{(i)})]_y\right)}{1 - \delta^{(i)}}. \tag{31}$$

Thus the sign of $g_y^{(i)}$ is determined by whether the EMA posterior $[f_\phi(x^{(i)})]_y$ is above or below the f_θ-weighted average $\delta^{(i)}$. During early learning, the EMA posterior tends to put larger mass on the true class $y^\star$. When $[f_\phi(x^{(i)})]_{y^\star} > \delta^{(i)}$, we have $g_{y^\star}^{(i)} < 0$.

This negative correction has the desired ELR effect. For a cleanly labeled sample $\hat{y}^{(i)} = y^\star$, the ordinary cross-entropy coefficient $[f_\theta(x^{(i)})]_{y^\star} - 1$ becomes close to zero once the sample is learned, so the sample would otherwise contribute little gradient. The negative correction $\lambda g_{y^\star}^{(i)}$ keeps increasing the true-class logit and preserves the clean-label signal. For a mislabeled sample $\hat{y}^{(i)} \neq y^\star$, the ordinary cross-entropy coefficient on the true class is $[f_\theta(x^{(i)})]_{y^\star} > 0$, which would decrease the true-class logit under gradient descent. Adding the negative term $\lambda g_{y^\star}^{(i)}$ offsets this harmful direction.

Similarly, if the observed noisy class $\hat{y}^{(i)}$ is not supported by the EMA posterior, i.e., $[f_\phi(x^{(i)})]_{\hat{y}^{(i)}} < \delta^{(i)}$, then $g_{\hat{y}^{(i)}}^{(i)} > 0$. This positive correction offsets the negative cross-entropy coefficient $[f_\theta(x^{(i)})]_{\hat{y}^{(i)}} - 1$, thereby suppressing the update that would otherwise increase the logit of the noisy label. In this sense, the rectification preserves early clean-label information while reducing memorization of false labels.

D.2 Class-Prior Estimation in Datasets with Weak Annotations

The main method assumes a class prior when forming diffusion-classifier posteriors. When this prior is not directly available, it can be estimated using standard class-prior estimation techniques [90–97]. We outline practical estimators for the three weak-annotation regimes considered in the paper.

D.2.1 Ambiguous-label datasets

In ambiguous-label learning, each instance is associated with a candidate label set rather than a single ground-truth label. This label ambiguity makes it difficult to estimate the class prior distribution, since simply counting training samples per class is no longer feasible. To address this issue, we adopt an iterative estimation strategy that updates the class prior in a moving-average manner.

We use the model's predicted labels as a proxy for class prior estimation. Since predictions in the early stage of training are often unreliable, we design a moving-average update rule that gradually stabilizes the estimated distribution. The update starts from a uniform prior $\mathbf{r} = [1/c, \ldots, 1/c]$, and is refined at each training epoch as

$$\mathbf{r} \leftarrow \mu\mathbf{r} + (1-\mu)\mathbf{z}, \qquad \mathbf{z}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left(j = \arg\max_{y \in S_i} f_y(x_i)\right), \tag{32}$$

where $\mu \in [0, 1]$ is a momentum parameter, S_i is the candidate label set for sample x_i, and $f_j(x_i)$ denotes the model prediction for class j. This rule progressively refines $\mathbf{r}$ as the model improves, leading to more accurate and stable estimation.

D.2.2 Semi-supervised datasets

In the case of semi-supervised datasets, the estimation of class prior is relatively straightforward. We assume that the distribution of the labeled dataset is consistent with that of the unlabeled dataset. Therefore, the class prior can be obtained by counting the class distribution over the labeled dataset.

D.2.3 Noisy-label datasets

We consider the widely adopted class-dependent label noise setting [98–102], where the observed noisy label of each $\mathbf{x} \in \mathcal{X}$ depends only on its underlying clean label. Formally, the transition probability from class i to class j is defined as

$$P(\widetilde{Y} = e_j \,|\, Y = e_i, X = \mathbf{x}) = P(\widetilde{Y} = e_j \,|\, Y = e_i) = T_{ij}, \quad \forall i, j \in [[c]],$$

where $\mathbf{T} = [T_{ij}] \in [0,1]^{c\times c}$ is the noise transition matrix. To make the estimation of $\mathbf{T}$ feasible, we follow prior work and impose the following assumptions.

Assumption D.2 (Sufficiently Scattered Assumption [103])**.** The clean class posterior $P(Y \mid X) = [P(Y = e_1 \mid X), \ldots, P(Y = e_c \mid X)]^\top \in [0,1]^c$ is said to be sufficiently scattered if there exists a set $\mathcal{H} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ such that the matrix $\mathbf{H} = [P(Y \mid X = \mathbf{x}_1), \ldots, P(Y \mid X = \mathbf{x}_m)]$ satisfies: (i) $\mathcal{Q} \subseteq \mathrm{cone}\{\mathbf{H}\}$, where $\mathcal{Q} = \{\mathbf{v} \in \mathbb{R}^c \mid \mathbf{v}^\top \mathbf{1} \geq \sqrt{c-1}\|\mathbf{v}\|_2\}$, and $\mathrm{cone}\{\mathbf{H}\}$ denotes the convex cone generated by the columns of $\mathbf{H}$; (ii) $\mathrm{cone}\{\mathbf{H}\} \nsubseteq \mathrm{cone}\{\mathbf{U}\}$ for any unitary matrix $\mathbf{U} \in \mathbb{R}^{c\times c}$ that is not a permutation matrix.

Assumption D.3 (Nonsingular $\mathbf{T}$)**.** The transition matrix $\mathbf{T}$ is nonsingular, i.e., $\mathrm{Rank}(\mathbf{T}) = c$.

Assumption D.2 ensures that the clean posteriors are sufficiently scattered so that the ground-truth $\mathbf{T}$ can be identified, while Assumption D.3 guarantees the invertibility of $\mathbf{T}$.

Let ϵ denote the noise rate. For symmetric label noise, we have $T_{ii} = 1 - \epsilon$ and $T_{ij} = \frac{\epsilon}{c-1}$ with $j \neq i$. In practice, the transition matrix can be estimated by solving the following optimization problem [103]:

$$\min_{\theta, \widehat{\mathbf{T}}} L(\theta, \widehat{\mathbf{T}}) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(\widehat{\mathbf{T}}^\top h_\theta(\mathbf{x}^{(i)}), \widetilde{y}^{(i)}\big) + \lambda \cdot \log\det(\widehat{\mathbf{T}}), \tag{33}$$

where ℓ is a loss function, $h_\theta(\cdot)$ is the output of a neural network parameterized by θ, and the regularization term $\log\det(\widehat{\mathbf{T}})$ encourages the estimated transition matrix to have minimal simplex volume. Here $\lambda > 0$ is a trade-off hyperparameter. By Assumption D.2, the solution $\widehat{\mathbf{T}}$ converges to the true $\mathbf{T}$ given sufficient noisy data (Theorem 1 in [103]).

Once the transition matrix $\mathbf{T}$ is estimated, the clean class prior $\pi = [\pi_1, \ldots, \pi_c]^\top$ can be obtained by solving the following system of linear equations:

$$\begin{cases} \widetilde{\pi}_1 = T_{11}\pi_1 + T_{21}\pi_2 + \cdots + T_{c1}\pi_c \\ \widetilde{\pi}_2 = T_{12}\pi_1 + T_{22}\pi_2 + \cdots + T_{c2}\pi_c \\ \quad \vdots \\ \widetilde{\pi}_c = T_{1c}\pi_1 + T_{2c}\pi_2 + \cdots + T_{cc}\pi_c \end{cases}, \tag{34}$$

where $\widetilde{\pi}_i = P(\widetilde{Y} = e_i)$ is the noisy class prior of the i-th class. The empirical estimate of $\widetilde{\pi}_i$ can be computed as

$$\widehat{\widetilde{\pi}}_i = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\{\widetilde{y}_j = e_i\}, \quad \forall i \in [[c]]. \tag{35}$$

Solving this system yields the clean class prior π, which is then used in subsequent modeling.

E Additional Experimental Results

E.1 Full Results in Weakly Supervised Classification

This subsection complements the main posterior analysis by evaluating the converged diffusion-classifier posterior as a standalone classifier. We compare against a broader set of specialized discriminative baselines than those shown in the main text.

Comparisons on Noisy Labels. We benchmark our method against seven widely used approaches: *Coteaching* [57], *Coteaching+* [104], *SCE* [105], *GCE* [106], *Decoupling* [107], *ELR* [46], and *JoCoR* [108]. These methods cover a range of strategies, from sample selection and reweighting to robust loss design, thus providing a diverse and rigorous benchmark.

Comparisons on Ambiguous Labels. We compare against nine representative baselines: *PRODEN* [47], *CAVL* [109], *POP* [110], *CC* [111], *IDGP* [59], *PiCO* [60], *ABLE* [112], *CRDPLL* [61], and *DIRK* [62]. For a fair comparison, we follow the hyperparameter settings used in *PLENCH* [22].

Comparisons on Semi-supervised Data. We follow the training and evaluation protocols of *USB* [113], a widely adopted benchmark for fair and unified SSL comparisons. Our evaluation covers a comprehensive spectrum of baselines, ranging from classical methods like *Pseudo-Labeling* [114], *VAT* [115], and *Mean Teacher* [48], to state-of-the-art approaches including *UDA* [116], *FixMatch* [117], *Dash* [63], *FlexMatch* [65], *CoMatch* [64], *SimMatch* [66], *FreeMatch* [118], and *SoftMatch* [67].

Results and Analysis. Quantitative results are reported in Tables 7, 8, and 9, where bold numbers indicate the best performance. *DELTA* remains competitive with or superior to state-of-the-art discriminative baselines across most settings, showing that the learned diffusion posterior is useful beyond generation.

Table 7: Classification test accuracy (%) on noisy-label datasets.

Method	Fashion-MNIST		CIFAR-10		ImageNette	
	Sym-40%	Asym-40%	Sym-40%	Asym-40%	Sym-40%	Asym-40%
CE	76.18±0.26	82.01±0.06	67.22±0.26	76.98±0.42	58.43±0.77	71.81±0.38
Co-learning	90.85±0.63	84.10±2.01	84.97±0.53	80.36±1.09	76.16±0.96	75.37±0.49
Co-teaching	92.17±0.34	92.78±0.25	86.54±0.57	79.38±0.39	66.55±1.00	75.12±0.50
Co-teaching+	91.05±0.06	91.62±0.20	67.28±1.85	79.43±0.47	75.79±0.79	75.17±0.40
SCE	93.62±0.22	88.60±0.20	82.82±0.40	81.54±0.64	77.99±0.39	74.81±1.04
GCE	**93.64±0.03**	87.48±0.09	85.00±0.27	77.97±3.69	81.18±0.35	72.61±1.14
Decoupling	92.24±0.23	92.10±0.44	82.24±0.28	79.89±0.58	75.53±0.69	78.24±0.21
ELR	93.13±0.13	92.82±0.09	85.68±0.13	81.32±0.31	84.03±2.86	73.51±0.31
JoCoR	84.05±1.11	89.45±4.43	77.92±3.92	78.68±0.07	67.82±1.97	74.67±0.43
Mixup	92.21±0.03	92.01±1.02	84.26±0.64	83.21±0.85	76.65±1.62	77.16±0.71
PENCIL	90.85±0.58	91.77±0.69	85.91±0.26	84.89±0.49	81.94±1.26	77.20±1.15
Vanilla	90.11±1.24	85.41±0.96	80.22±0.10	86.31±0.10	55.86±1.95	53.91±1.07
DELTA	93.40±0.40	**93.20±0.30**	**88.63±0.12**	**88.83±0.33**	**84.12±0.18**	**79.30±0.27**

Table 8: Classification test accuracy (%) on ambiguous-label datasets.

Method	Fashion-MNIST		CIFAR-10		ImageNette	
	Random	Class-Dept.	Random	Class-Dept.	Random	Class-Dept.
PRODEN	93.31±0.07	93.44±0.21	90.02±0.22	90.44±0.44	84.75±0.13	83.50±0.60
CAVL	93.09±0.17	92.67±0.25	87.28±0.64	87.16±0.58	41.69±4.12	46.46±7.15
POP	93.59±0.17	93.57±0.19	89.13±0.22	90.19±0.10	84.65±0.55	84.29±0.17
CC	93.17±0.32	92.65±0.29	88.40±0.24	89.12±0.23	81.11±0.50	80.74±0.68
IDGP	92.26±0.25	93.07±0.16	89.65±0.53	90.83±0.34	84.07±0.26	82.18±0.13
PiCO	93.32±0.12	93.32±0.33	86.40±0.89	87.51±0.66	82.15±0.23	84.41±0.93
ABLE	93.02±0.26	93.20±0.16	90.77±0.33	90.74±0.48	71.81±2.46	76.53±1.28
CRDPLL	94.03±0.14	93.80±0.23	92.74±0.26	92.89±0.27	84.31±0.25	88.08±0.34
DIRK	94.11±0.22	93.99±0.24	93.48±0.14	93.22±0.37	87.90±0.11	87.47±0.17
Vanilla	80.20±1.29	66.03±1.43	60.25±0.17	56.34±0.50	56.04±0.61	59.47±0.51
DELTA	**94.27±0.55**	**94.20±0.15**	**94.70±0.49**	**93.53±0.12**	**89.31±0.21**	**88.42±0.43**

Table 9: Classification results on semi-supervised datasets.

Method	Fashion-MNIST		CIFAR-10		ImageNette	
	1% Labels	10% Labels	1% Labels	10% Labels	1% Labels	10% Labels
Pseudo-Label	83.53±0.46	89.59±0.23	50.10±0.95	72.92±0.17	43.00±0.82	68.03±0.32
Mean Teacher	82.34±0.09	89.91±0.15	47.69±0.27	73.01±0.78	40.53±1.56	65.72±0.55
VAT	83.31±0.61	89.35±0.12	49.64±0.90	71.07±1.27	38.63±8.39	63.93±5.18
UDA	84.28±0.41	90.83±0.34	69.20±1.41	80.50±0.55	50.52±3.79	72.53±1.17
FixMatch	84.32±0.33	90.76±0.38	67.48±1.42	80.00±0.63	50.41±4.43	71.32±1.93
Dash	84.73±0.09	91.16±0.20	70.14±0.69	81.50±0.68	57.68±2.19	74.66±0.81
CoMatch	85.31±0.29	90.52±0.12	61.45±1.46	77.79±0.53	63.88±0.78	73.20±0.46
FlexMatch	84.43±0.30	90.69±0.03	70.72±0.93	81.35±0.48	61.39±0.70	73.08±0.13
FreeMatch	84.30±0.37	90.92±0.24	70.15±0.44	80.99±0.56	60.37±1.11	73.14±1.03
SimMatch	84.69±0.17	91.18±0.13	73.33±1.02	82.90±0.43	58.12±2.66	76.12±0.45
SoftMatch	84.72±0.23	91.22±0.11	73.74±0.82	88.66±0.60	58.50±2.31	75.75±0.25
Vanilla	78.37±0.72	90.50±1.00	53.49±0.15	85.13±0.12	49.55±0.99	74.70±0.53
DELTA	**85.92±0.13**	**92.97±0.21**	**76.40±0.54**	**92.47±0.39**	**68.23±0.19**	**77.30±0.15**

E.2 Results under Coexisting Weak-Annotation Types

The main experiments evaluate each weak-annotation regime separately. To further verify that *DELTA* operates as a unified framework, we construct a CIFAR-10 setting in which all three mechanisms coexist: the training set is partitioned into four disjoint subsets with 20% correctly labeled, 20% wrongly labeled (uniformly flipped), 30% ambiguously labeled (random policy), and the remaining 30% unlabeled samples. *DELTA* requires no algorithmic modification in this setting: the rectified target in Eq. (10) is defined per sample according to the form of its annotation—a single label activates the noisy branch, a candidate set the ambiguous branch, and an empty annotation the semi-supervised branch. Table 10 shows that *DELTA* remains stable when the three mechanisms coexist, substantially outperforming *Vanilla* on all generative metrics and in test accuracy.

Table 10: Results on CIFAR-10 where noisy, ambiguous, and missing labels coexist.

Method	FID↓	IS↑	Dens.↑	Cov.↑	CW-FID↓	CW-Dens.↑	CW-Cov.↑	Test Acc. (%)↑
Vanilla	10.92	9.05	97.64	69.83	30.17	80.46	68.52	61.24
DELTA	**3.71**	**9.55**	**105.83**	**79.86**	**16.24**	**103.92**	**77.95**	**85.81**

E.3 Results on a Domain-Realistic Medical Dataset

To assess external validity beyond CIFAR-style benchmarks, we evaluate *DELTA* on HAM10000 [119], a real medical dataset of dermatoscopic images with seven lesion types, corrupted with 20% symmetric label noise. As shown in Table 11, *DELTA* retains its advantage in this real medical domain. Notably, HAM10000 is heavily class-imbalanced (the largest class alone accounts for 67% of the data): *Vanilla*'s accuracy stays at the majority-class level, whereas *DELTA* clearly exceeds it.

Table 11: Results on HAM10000 with 20% symmetric noisy labels.

Method	FID↓	IS↑	Dens.↑	Cov.↑	CW-FID↓	CW-Dens.↑	CW-Cov.↑	Test Acc. (%)↑
Vanilla	48.16	3.25	73.94	44.51	91.38	62.75	40.16	69.87
DELTA	**39.72**	**3.52**	**83.61**	**52.38**	**65.44**	**76.92**	**47.15**	**79.34**

E.4 Ablation Results of Median-Centered Timestep Sampling

A retrained variant without median-centered sampling is intractable, as posterior estimation would revert to full-timestep computation, which is prohibitively expensive in both time and memory (Table 4). We therefore ablate it at the inference stage, comparing it against the uniform timestep sampling of Chen et al. [36] on CIFAR-10 under matched step budgets and input noise levels σ_t. Table 12 reports the numbers underlying Figure 2(b): the median-centered scheme consistently outperforms uniform sampling across all step budgets and noise levels, with the margin growing markedly at higher noise levels (e.g., 60.30% vs. 18.93% at $\sigma_t = 1.0$ with two steps).

Table 12: Classification accuracy (%) of median-centered versus uniform timestep sampling on CIFAR-10.

Noise level	Method	Sampling steps				
		2	4	8	16	32
$\sigma_t \approx 0$	Median-centered (ours)	**74.57**	**87.07**	**90.70**	**93.50**	**95.20**
	Uniform [36]	69.90	85.40	90.57	91.97	93.33
$\sigma_t = 0.5$	Median-centered (ours)	**75.53**	**82.57**	**86.53**	**86.87**	**87.10**
	Uniform [36]	42.83	61.27	68.70	70.67	71.70
$\sigma_t = 1.0$	Median-centered (ours)	**60.30**	**67.70**	**71.43**	**72.80**	**74.27**
	Uniform [36]	18.93	29.57	35.60	38.63	39.83

Table 13: Accuracy (%) under different interval widths Δ ($\sigma_t = 0.5$; CIFAR-10, Sym-40%).

Δ	Sampling steps				
	2	4	8	16	32
4	74.86	81.92	85.71	86.03	86.24
5	75.31	82.14	86.22	86.55	86.79
6.4	**75.53**	**82.57**	**86.53**	**86.87**	**87.10**
7	75.42	82.39	86.44	86.71	87.02
8	75.18	82.08	86.12	86.49	86.71

E.5 Sensitivity Analysis of the Sampling Interval Width

We further examine the sensitivity of the median-centered timestep sampling to the interval width Δ. Fixing $\sigma_t = 0.5$, we sweep $\Delta \in \{4, 5, 6.4, 7, 8\}$ around the default $\Delta = 6.4$ (Appendix B.2) and report the classification accuracy across sampling steps on CIFAR-10 with 40% symmetric label noise. As shown in Table 13, the accuracy varies by less than 0.9 percentage points at every step budget, and even the least favorable width still outperforms uniform sampling by a wide margin (Table 12), confirming that the efficiency gains are robust to the choice of Δ.

F Weakly Supervised Dataset Condensation

This section develops the weakly supervised dataset condensation application discussed in the main paper. The goal is to distill large weakly annotated datasets into compact synthetic counterparts for efficient downstream training. In this setting, a trained *DELTA* model acts as a generative compressor that synthesizes class-representative samples from weakly annotated data. This differs from conventional condensation methods, which usually assume clean labels, and from recent noise-robust extensions such as *RDC* [120], which are tailored to a single weak form.

F.1 Setup

During condensation, we use the trained LCDM to synthesize images according to the specified images per class (IPC). For evaluation, we compare against random selection (*RandSel*), hard-label-based methods (*DC* [70], *DM* [121], *DSA* [122]), soft-label-based methods (*RDED* [123], *SRe2L* [124]), and the noise-robust baseline *RDC* [120]. Following standard protocols [124], we adopt ResNet-18 during condensation and evaluate the condensed datasets on a held-out test set using ResNet-34.

F.2 Representative Results

Tables 14, 15, and 16 report the classification accuracy on condensed datasets across the three weak-annotation scenarios at multiple IPC settings. In Table 15, "-" indicates that the number of available labeled samples per class is already below the target IPC, so no additional condensation experiment is conducted for that setting. Standard condensation methods degrade markedly under heavy label noise. In contrast, *DELTA* attains competitive or superior accuracy compared to the noise-robust baseline *RDC*, and consistently outperforms *RDED* and *SRe2L* in ambiguous-label and semi-supervised scenarios. Qualitative comparisons in Figure 5 indicate that our method generates samples with clearer class semantics. Comprehensive IPC analyses and additional visualizations are provided in the following two subsections.

F.3 Comprehensive IPC Analyses

We evaluate *DELTA* on the weakly supervised dataset condensation task across three weak-annotation scenarios with extended IPC settings: **noisy labels** (Table 14), **semi-supervised data** (Table 15), and **ambiguous labels** (Table 16). First, under label noise, *DELTA* almost achieves the best performance across most datasets and noise types, effectively suppressing noise accumulation in the condensed set. Second, in the semi-supervised setting, *DELTA* exhibits strong data efficiency. Finally, regarding ambiguous labels, our method demonstrates superior disambiguation capabilities. Unlike decoupled methods (e.g., *RDED*, *SRe2L*) that struggle with candidate sets, *DELTA* achieves remarkable accuracy gains. Across scenarios, performance trends are also stable as IPC increases: *DELTA* remains competitive in low-IPC regimes where information is severely limited, while preserving gains at higher IPC where optimization instability and label corruption can otherwise accumulate. The improvements are particularly consistent on more challenging datasets and corruption settings, suggesting that the learned generator captures class-level semantics rather than memorizing noisy supervision. Compared with hard-label condensation baselines, this behavior indicates stronger robustness to supervision mismatch; compared with soft-label decoupling strategies, it reflects better end-to-end alignment between semantic generation and class assignment. These results collectively highlight that our generative strategy can effectively recover clean distributions and generate compact, reliable synthetic datasets under diverse weak-annotation conditions.

F.4 Visualization of Weakly Supervised Condensed Datasets

We visualize synthesized images on Fashion-MNIST and CIFAR-10 in Figure 5. Ideally, a robust condensation method should filter out annotation corruption and generate clean, class-representative samples. Standard optimization-based methods such as *RDED* and *SRe2L*, as well as related hard-label condensation baselines such as *DC*, *DSA*, and *DM*, tend to overfit corrupted annotations, producing semantic mismatches between synthesized images and assigned class labels. In contrast, *DELTA* produces samples with clearer semantics and better class alignment. These qualitative

Table 14: Test accuracy (%) on condensed datasets from the noisy-label datasets.

Dataset	Type	IPC	*RandSel*	*DC*	*DSA*	*DM*	*RDED*	*SRe2L*	*RDC*	*DELTA*
Fashion-MNIST	Sym-40%	10	34.42±0.69	22.85±1.69	42.07±2.49	57.06±1.52	18.57±1.06	15.80±0.38	68.88±1.16	**70.18±0.37**
		50	52.36±0.60	35.64±2.26	55.22±1.51	68.23±0.47	23.19±0.74	19.51±0.96	72.39±1.35	**80.73±0.07**
		100	55.14±0.06	30.46±1.74	41.30±0.85	73.21±0.69	25.43±0.21	19.66±1.91	81.91±0.54	**84.26±0.02**
	Asym-40%	10	48.28±0.34	53.17±1.59	57.15±2.37	63.27±1.60	18.42±1.62	16.45±1.96	**65.17±3.80**	65.02±1.85
		50	69.44±0.17	49.21±0.69	77.20±0.34	76.39±0.57	22.31±0.67	27.07±0.35	73.11±1.48	**79.65±0.63**
		100	70.80±0.91	36.95±0.57	80.24±0.54	78.43±0.63	24.03±0.97	26.52±1.46	81.59±1.29	**83.22±0.33**
CIFAR-10	Sym-40%	10	16.30±0.96	18.11±1.02	18.06±1.72	23.71±0.40	19.85±0.88	13.12±1.04	25.88±1.16	**27.83±0.98**
		50	26.59±0.70	20.63±0.22	28.76±0.57	29.50±0.56	34.64±0.58	14.23±1.67	42.39±1.17	**46.47±0.41**
		100	31.19±0.74	19.91±0.54	29.45±0.34	32.26±0.75	44.03±0.21	14.21±0.93	51.81±0.93	**56.53±0.03**
	Asym-40%	10	24.89±1.65	18.51±1.35	22.23±1.80	26.53±0.07	23.48±0.65	14.64±1.03	**27.39±2.92**	24.94±0.49
		50	40.95±0.59	25.97±0.97	40.81±0.29	43.09±0.76	39.12±0.13	16.03±0.21	44.41±1.26	**47.77±0.78**
		100	47.49±0.64	27.76±0.72	42.96±0.84	51.61±0.60	44.45±0.19	17.55±0.91	53.81±0.93	**55.89±0.39**
ImageNette	Sym-40%	10	23.09±0.19	15.89±0.73	27.70±1.25	28.83±0.73	21.15±1.05	25.03±1.17	33.23±1.07	**34.36±1.05**
		50	33.83±0.28	24.62±0.73	32.07±1.01	42.66±1.27	35.87±0.39	35.37±0.82	42.88±0.78	**44.93±0.28**
		100	40.04±0.71	22.81±1.22	36.05±1.76	43.25±2.13	35.87±0.39	41.74±1.37	49.76±1.56	**56.23±0.84**
	Asym-40%	10	26.54±0.88	19.26±0.98	30.62±2.09	33.40±0.48	26.23±0.06	25.74±2.21	35.46±1.54	**37.09±0.29**
		50	47.91±0.61	31.68±2.15	43.41±1.24	50.97±1.61	32.75±0.43	35.29±0.14	53.37±1.40	**55.20±0.46**
		100	59.10±1.41	29.19±0.21	53.79±0.84	60.70±1.88	35.48±0.22	42.37±0.34	67.56±0.52	**68.97±0.12**

Table 15: Test accuracy (%) on condensed datasets from the semi-supervised annotations.

Dataset	Type	IPC	*RandSel*	*DC*	*DSA*	*DM*	*RDED*	*SRe2L*	*DELTA*
Fashion-MNIST	1% Labels	10	64.71±1.30	66.32±1.97	67.08±1.90	68.32±1.70	66.63±1.52	63.12±2.47	**69.20±0.35**
		50	83.33±0.37	83.02±2.96	84.46±0.64	83.60±0.22	87.68±3.30	85.82±2.65	**88.99±0.39**
		100	-	-	-	-	-	-	**89.29±0.23**
	10% Labels	10	63.22±0.17	64.42±3.60	64.27±1.86	68.84±3.30	67.92±1.68	69.96±2.49	**70.25±0.53**
		50	80.92±0.06	81.11±0.63	81.31±0.56	83.58±0.31	83.51±1.78	83.37±0.26	**83.85±1.79**
		100	83.85±0.24	83.60±0.76	83.18±0.20	86.29±0.38	**89.74±0.39**	87.03±2.07	84.54±0.22
CIFAR-10	1% Labels	10	24.51±0.34	23.26±0.64	24.90±1.22	28.73±0.37	27.01±9.92	31.87±1.54	**33.43±0.62**
		50	-	-	-	-	-	-	**44.74±0.67**
		100	-	-	-	-	-	-	**52.83±1.09**
	10% Labels	10	23.17±1.63	21.84±1.37	24.90±0.23	29.71±1.69	32.38±0.71	31.61±0.65	**33.53±0.58**
		50	41.51±0.57	41.31±0.93	44.20±0.64	43.71±1.58	43.11±1.18	42.44±0.62	**45.15±0.16**
		100	54.06±0.67	55.58±0.89	57.06±1.04	57.48±1.48	57.11±0.60	56.36±0.88	**58.99±0.95**

Table 16: Test accuracy (%) on condensed datasets from the ambiguous-label annotations.

Dataset	**Random**				**Class-Dept.**			
	IPC	*RDED*	*SRe2L*	*DELTA*	IPC	*RDED*	*SRe2L*	*DELTA*
Fashion-MNIST	10	13.17±4.66	9.93±1.15	**72.02±0.77**	10	13.53±1.12	10.83±0.35	**70.46±0.26**
	50	14.47±1.06	10.30±0.40	**83.98±0.12**	50	15.05±0.26	11.33±0.76	**79.67±0.13**
	100	15.00±0.18	10.48±0.74	**87.30±0.31**	100	16.91±2.75	11.05±0.77	**81.42±0.25**

comparisons support the quantitative findings above and show that the generative framework can yield high-quality condensed datasets from imperfect annotations.

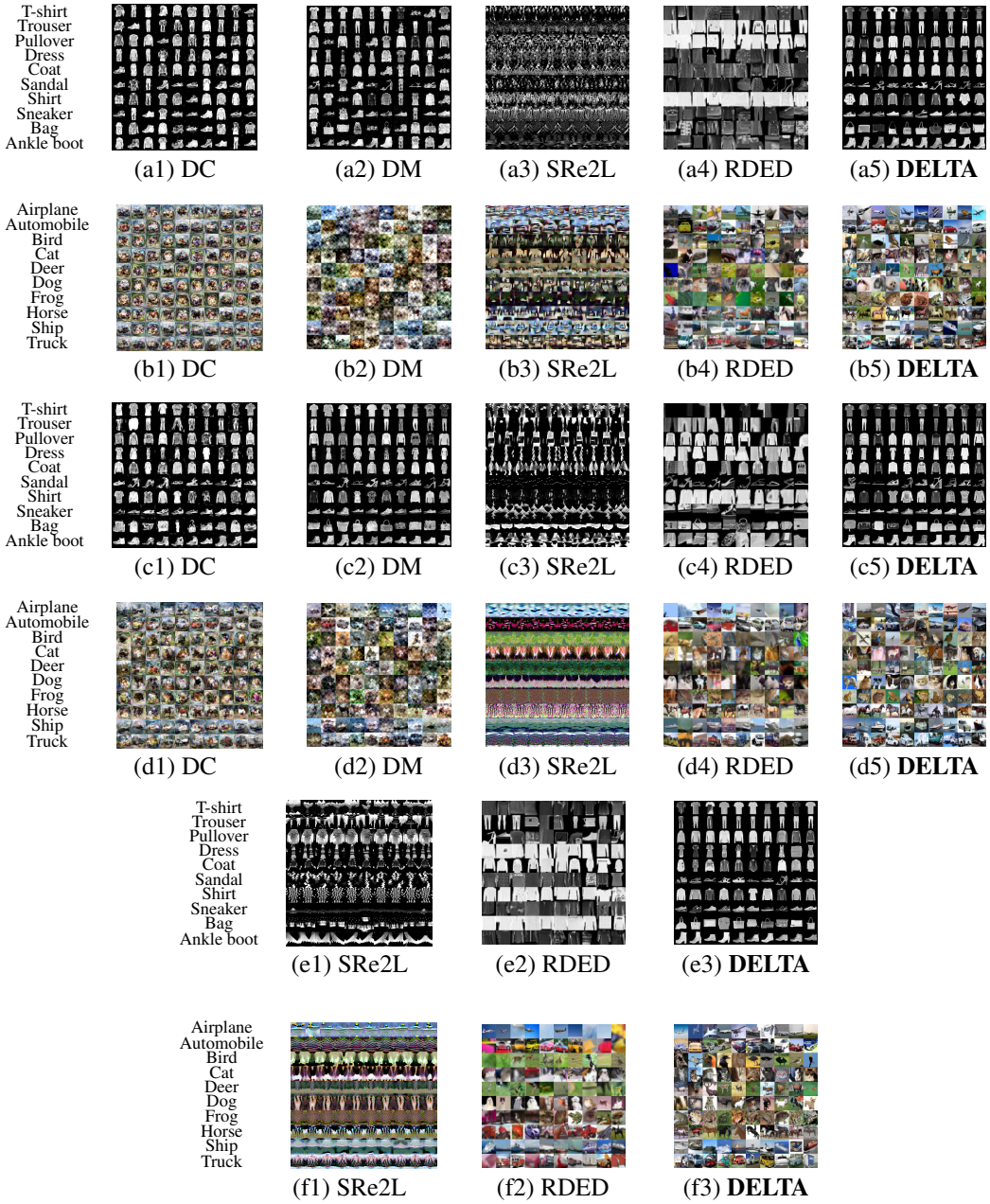


Figure 5: Visualization of condensed images (IPC=10) synthesized by different methods under various weak annotations. Rows (a) and (b) show results under noisy labels (Sym-40%) for Fashion-MNIST and CIFAR-10, respectively. Rows (c) and (d) show results under semi-supervised data (1% Labels) for Fashion-MNIST and CIFAR-10. Rows (e) and (f) display results under ambiguous labels (Random) for Fashion-MNIST and CIFAR-10.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 claim (i) a unified framework handling all three forms of weak annotations with less prior knowledge, (ii) a median-centered timestep sampling strategy, and (iii) consistent improvements over specialized baselines. All three are substantiated in Sections 4 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses two limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results (Theorem 4.1, Proposition 4.2, Theorem 4.3, Theorem C.1) state their assumptions explicitly, and full proofs are provided in Appendices C.3, C.4, C.6, and C.7, respectively.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Appendix B fully specifies the weak-annotation construction (B.1), network architecture, optimizer, learning rate, EMA decay, batch size, and total iterations (B.2), and evaluation metrics (B.3).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used are publicly available; anonymized source code with reproduction scripts is included in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 and Appendix B specify data splits, weak-annotation protocols, optimizer choice and hyperparameters, model architecture, and training schedule.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Classification tables (Tables 5, 7, 8, 9) report mean $\pm$ standard deviation over multiple runs. Generation metrics in Tables 1–3 follow the standard FID/IS evaluation protocol on a fixed number of samples, consistent with prior work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B states that all experiments were run on NVIDIA Tesla A100 GPUs with CUDA 12.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. It uses only publicly available benchmark datasets and involves no human subjects or sensitive data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work improves the data-efficiency of generative models under imperfect supervision, which is broadly beneficial in domains such as medical imaging and remote sensing where clean labels are costly.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our models do not pose elevated risks of misuse beyond those of comparable academic diffusion models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and baseline methods are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.