RETRIEVAL-CALIBRATED ONE-STEP CATEGORICAL DENOISING FOR LEARNING WITH SEVERE LABEL NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning with severe label noise requires a model to recover useful supervision without prematurely committing to potentially incorrect labels. Recent label-diffusion methods address this problem by progressively denoising corrupted label states, but it remains unclear which part of the diffusion formulation is actually responsible for their effectiveness: iterative reverse generation, or learning from label states with systematically varying reliability. We investigate this question through RC-Cat, a retrieval-calibrated categorical denoising framework that separates corruption-aware training from iterative generative inference. RC-Cat first converts each noisy annotation into a sparse candidate distribution using reliability-calibrated neighborhood evidence, thereby preserving uncertainty over the unknown clean class. It then corrupts sampled categorical label states across different noise levels and trains a shared timestep-conditioned denoiser to reconstruct the candidate distribution. Controlled experiments reveal a clear mechanism. On CIFAR-100 with severe PMD-70 noise, introducing categorical corruption improves accuracy from 60.29% to 64.78%, while exposing the corruption timestep further raises it to 71.29%. In contrast, adding the exact categorical posterior regularizer produces no measurable improvement, and iterative reverse inference does not outperform direct prediction from the terminal corruption level. When the representation and candidate supervision are held fixed, RC-Cat exceeds a parameter-matched soft-target classifier by 5.96 percentage points; the advantage remains 2.53 points when externally pretrained CLIP features are replaced by a target-data-only SimCLR representation. Improvements also extend to composite synthetic noise and CIFAR-100N, although they become smaller on Animal-10N and negligible on Clothing1M. These results suggest that the useful inductive bias of label diffusion can arise primarily from corruption-aware learning: exposing a classifier to label states of varying reliability and explicitly informing it of their corruption level. From this perspective, iterative label generation is not necessary to obtain the observed denoising benefit, enabling a one-step categorical formulation that is substantially smaller and faster than a protocol-matched iterative diffusion baseline.

1 INTRODUCTION

Learning from noisy labels entails more than identifying reliable annotations, as a classifier must also learn from the residual uncertainty after correction. This becomes challenging under severe label noise: the observed label may be incorrect, yet replacing it with a single predicted label can reinforce prior mistakes. Under hard correction—where noisy labels are replaced by single predicted labels—the corrected targets may remain erroneous. For this reason, expressive networks can eventually memorize these corrupted supervision signals (Arpit et al., 2017; Zhang et al., 2017; Liu et al., 2020). Fixed soft target method avoids an immediate hard decision, but still leaves the model to learn a useful predictor from an imperfect target (Han et al., 2018; Li et al., 2020; Zhang & Sabuncu, 2018) (illustrated in Figure 1 **Left**). The data-level correction method and the way it enters training therefore need to be considered together.

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

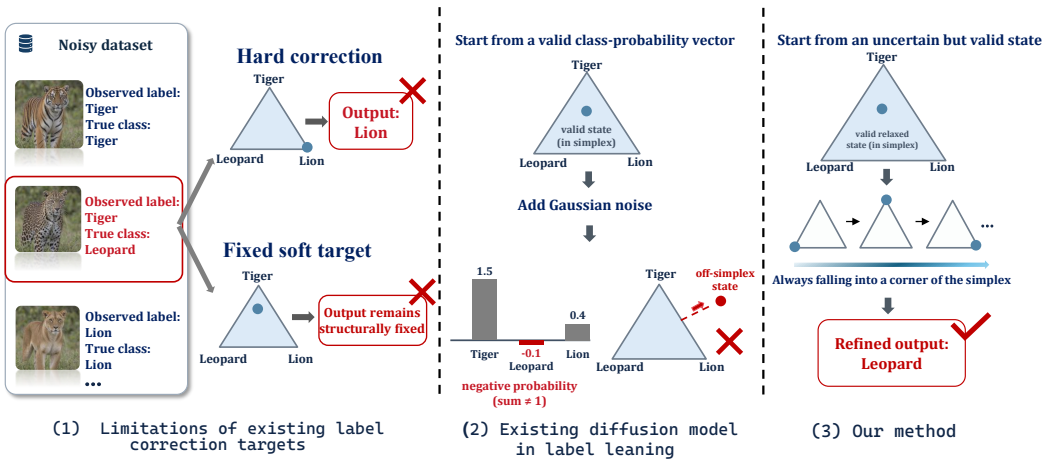

Figure 1: Motivation for categorical label denoising. **Left**: In data-level approaches, hard correction can reinforce erroneous labels, while fixed soft targets cannot adapt to varying label reliability. **Middle**: In model-level approaches, Continuous label diffusion may produce invalid off-simplex states under Gaussian corruption. For example, adding Gaussian noise to a valid three-class probability vector such as $(0.7, 0.2, 0.1)$ may yield a state such as $(1.1, -0.1, 0.3)$, which contains a negative entry and no longer sums to one. **Right**: RC-Cat(our model) preserves uncertainty while operating entirely on valid categorical states.

Label diffusion extends data-level supervision correction by introducing corruption-based denoising into model training. LRA-Diffusion constructs targets from retrieved neighbors (Chen et al., 2023), while DLD models directional and random corruption followed by iterative refinement (Hou et al., 2025). However, continuous Gaussian corruption does not preserve the probability-simplex geometry of class labels. Such off-simplex states are valid intermediate variables for continuous diffusion, but they no longer admit a direct interpretation as categorical label probabilities (illustrated in Figure 1 **Middle**). Moreover, existing methods couple target construction, corruption-aware training, noise-level conditioning, and iterative reverse inference, making it difficult to isolate the source of their denoising benefits. This raises a fundamental question: *can corruption-aware, timestep-conditioned training retain the benefits of label denoising without iterative reverse inference?* To investigate this question, we adopt a categorical corruption to preserve valid label states while separating corruption-aware training from iterative reverse inference (illustrated in Figure 1 **Right**).

We study this mechanistic question with RC-Cat, a retrieval-calibrated categorical denoising framework. RC-Cat first aggregates noisy labels from semantically similar images into a sparse candidate distribution. It then samples a candidate class, applies categorical corruption, and trains a shared denoiser on the image representation, corrupted state, and timestep. The timestep has a direct interpretation in this task: it quantifies the expected reliability of the auxiliary label state. At inference, RC-Cat bypasses the reverse chain and predicts directly from the terminal corruption level.

Controlled experiments support this separation between diffusion-style training and inference. On CIFAR-100 with PMD-70 noise, corruption improves accuracy by 4.49 points, and timestep conditioning adds another 6.51 points, whereas posterior regularization and reverse-chain inference provide no measurable gain. With identical features and candidate targets, RC-Cat outperforms a parameter-matched classifier by 5.96 points, and the improvement persists across multiple synthetic and human-noise settings. Compared with DLD, RC-Cat achieves competitive accuracy with $13.21\times$ fewer optimized parameters, a $26.39\times$ smaller checkpoint, and $1.82\times$ higher inference throughput. These results show that corruption-aware training can retain the benefits of label denoising without iterative inference.

Our contributions are threefold:

- We isolate corruption-aware training from reverse-chain inference in label diffusion and show that the tested iterative chains are unnecessary for the observed denoising benefit.

- We propose RC-Cat, which combines reliability-calibrated candidate targets with timestep-conditioned categorical corruption and one-step terminal prediction.
- We validate the proposed mechanism across representations, noise processes, and datasets, and demonstrate a favorable accuracy–efficiency trade-off against matched classifier and diffusion baselines.

2 Related Work

Learning with noisy labels. Existing methods improve robustness through loss correction, sample selection, semi-supervised learning, and regularization (Song et al., 2023). Co-teaching exchanges small-loss examples between two networks (Han et al., 2018); DivideMix partitions examples by their loss and applies semi-supervised learning (Li et al., 2020); generalized cross entropy and early-learning regularization directly modify the optimization objective (Zhang & Sabuncu, 2018; Liu et al., 2020). These methods primarily decide which labels to trust or how strongly to fit them. Our focus is complementary: given a fixed uncertainty-aware target, we study whether training under controlled categorical corruption provides a useful inductive bias.

Retrieval-based supervision. Semantic neighbors provide evidence beyond a single noisy annotation, but their quality depends on the representation. CLIP transfers strong visual semantics from image-text pretraining (Radford et al., 2021), while LRA-Diffusion uses retrieved examples to construct cleaner supervision (Chen et al., 2023). We aggregate neighbor labels into sparse distributions rather than hard corrections. Crucially, all mechanism comparisons share the same frozen representation and candidate targets, and a target-data-only SimCLR control tests whether the gain depends on external pretraining.

Diffusion for label recovery. LRA-Diffusion and DLD corrupt continuous label representations and recover them through iterative reverse inference (Chen et al., 2023; Hou et al., 2025). Discrete diffusion instead defines transitions with stochastic matrices and admits exact categorical marginals (Austin et al., 2021; Hoogeboom et al., 2021). Subsequent work has extended discrete diffusion through continuous-time Markov formulations, general discrete-state processes, and alternative objectives and parameterizations (Campbell et al., 2022; Santos et al., 2023; Lou et al., 2024). Masked discrete diffusion has further developed this direction for high-dimensional sequence generation (He et al., 2023; Sahoo et al., 2024; Shi et al., 2024). These discrete formulations were primarily developed for generative modeling, where reverse sampling is used to progressively construct a high-dimensional structured sample from noise. Our setting is fundamentally different: the prediction target is a single categorical variable, while the image representation already provides the semantic information needed for classification. We therefore use categorical diffusion as a corruption-aware training mechanism rather than assuming that its generative reverse process must also be retained at inference. This specialization lets us separate two choices that prior label-diffusion systems couple: learning from multiple corruption levels and iteratively generating the final label.

3 Retrieval-Calibrated Categorical Denoising

3.1 Overview

Given noisy annotations, we first construct a sparse candidate distribution s_i by aggregating semantically similar training examples with reliability-calibrated retrieval. Rather than treating s_i as a fixed soft target alone, RC-Cat uses it to generate categorical label states under controlled corruption levels and trains a shared timestep-conditioned denoiser to recover the candidate supervision. Importantly, the corruption trajectory serves as a training mechanism rather than an inference procedure: at inference, the learned denoiser predicts directly from terminal corruption without traversing a reverse chain. Figure 2 summarizes these information paths; Appendix B gives the complete algorithms.

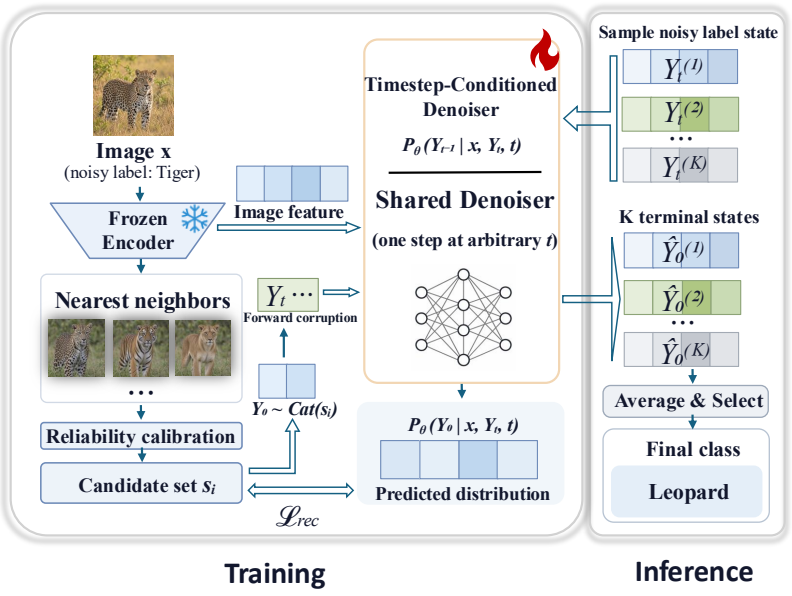

Figure 2: Overview of RC-Cat. Retrieval with reliability calibration constructs an uncertainty-aware candidate distribution from noisy labels. During training, categorical corruption generates label states at varying noise levels, and a shared timestep-conditioned denoiser learns to recover the candidate distribution. At inference, RC-Cat bypasses the reverse chain and directly predicts from multiple terminal states in one step.

3.2 Retrieval-Calibrated Candidate Distribution

Let $\widetilde{\mathcal{D}} = \{(x_i, \widetilde{y}_i)\}_{i=1}^N$ be a training set with observed noisy labels, x_i is input image and $\widetilde{y}_i \in \{1, \dots, C\}$ is noisy label. The clean label is unavailable during target construction and optimization. A frozen encoder maps each image to $z_i = g(x_i)/\|g(x_i)\|_2$. Let $\mathcal{N}_k(i)$ be the k nearest input images under cosine similarity. We retain reciprocal neighbors $\mathcal{M}_i = \{j : j \in \mathcal{N}_k(i),\ i \in \mathcal{N}_k(j)\}$, and fall back to directed neighbors when $|\mathcal{M}_i|$ is too small. We therefore weight each neighbor according to its agreement with the noisy labels in its own neighborhood:

$$r_j = \left[\max\left(\frac{1}{|\mathcal{M}_j|} \sum_{\ell \in \mathcal{M}_j} \mathbb{I}(\widetilde{y}_\ell = \widetilde{y}_j), r_{\min} \right) \right]^{\gamma_r}. \tag{1}$$

The floor $r_{\min}$ prevents a neighbor's contribution from vanishing solely because its local agreement is low; γ_r controls the strength of the weighting. This score measures local agreement with observed labels. Similarity and reliability jointly determine the vote, with an observed-frequency correction applied after aggregation:

$$w_{ij} = \exp\left(\frac{z_i^\top z_j}{\tau_r} \right) r_j, \qquad v_{ic} = \frac{\sum_{j \in \mathcal{M}_i} w_{ij} \mathbb{I}(\widetilde{y}_j = c)}{(n_c/\bar{n})^{\gamma_f}}, \tag{2}$$

where n_c is the observed class frequency and $\bar{n} = N/C$. The temperature τ_r controls similarity weighting, and γ_f controls the correction for differences in observed class frequency. These quantities all use the noisy training data.

After retaining the top-M classes and the observed label, we normalize the votes to ρ_i and form

$$s_i = (1 - \lambda_r) e_{\widetilde{y}_i} + \lambda_r \rho_i, \qquad s_i \in \Delta^{C-1}, \tag{3}$$

where e_c is one-hot. Retrieval introduces alternatives supported by the neighborhood, while the observed annotation retains a share of the target mass even when it is not among the strongest neighbor votes. The mixture therefore preserves several possible classes without replacing the annotation by

a hard pseudo-label. If the retained votes have zero total mass, the construction falls back to the observed-label distribution, as specified in Algorithm 1.

Local agreement can still reflect a shared annotation error, so the candidate distribution is an imperfect source of supervision. RC-Cat keeps this distribution fixed during training rather than progressively replacing it with the head's predictions. This makes its role explicit: candidate construction supplies the target, and the denoising procedure determines how the head is trained with that target.

3.3 Categorical Forward Corruption

The candidate distribution varies across images; the corruption prior is shared across them. For an empirical class prior $\boldsymbol{\pi}$, define $q_\eta(c) = \pi_c^\eta / \sum_j \pi_j^\eta$ (the main configuration uses a uniform q_η). With $\alpha_t = 1 - \beta_t$, the transition and its cumulative form are

$$Q_t = (1-\beta_t)I + \beta_t \mathbf{1} q_\eta^\top, \qquad \bar{Q}_t = Q_1 \cdots Q_t = \bar{\alpha}_t I + (1-\bar{\alpha}_t)\mathbf{1} q_\eta^\top, \tag{4}$$

where $\bar{\alpha}_t = \prod_{u=1}^t \alpha_u$. A transition retains the current state with probability $1-\beta_t$ and otherwise resamples it from q_η. The cumulative form follows because $\mathbf{1} q_\eta^\top$ is idempotent.

Training samples a candidate $Y_0 \sim \mathrm{Cat}(\mathbf{s}_i)$ followed by

$$Y_t \sim \mathrm{Cat}(\bar{\alpha}_t \mathbf{e}_{Y_0} + (1-\bar{\alpha}_t) q_\eta)\,. \tag{5}$$

The timestep is sampled uniformly from $\{1, \ldots, T\}$. Every state is a class index, and repeated visits to an example expose the head to different candidate and corruption states. Because the reconstruction target is the full $\mathbf{s}_i$, Algorithm 2 samples the same input distribution directly from the cumulative marginal

$$\mathbf{u}_{i,t} = \bar{\alpha}_t \mathbf{s}_i + (1-\bar{\alpha}_t) q_\eta, \qquad Y_t \sim \mathrm{Cat}(\mathbf{u}_{i,t}). \tag{6}$$

This implements the categorical sampling process without simulating the preceding transitions. The chosen prior is stationary under the transition: $q_\eta^\top Q_t = q_\eta^\top$.

As corruption increases, the auxiliary state shifts from candidate-dependent evidence toward the shared prior. The timestep identifies where a training input lies in this process. It describes how the state was generated from the candidate distribution; the quality of that candidate distribution still depends on the noisy annotations and retrieval. Keeping these roles separate lets us vary corruption while leaving the supervision unchanged.

3.4 Timestep-Conditioned Denoiser

The network predicts

$$p_\theta(Y_0 \mid z_i, Y_t, t) = \mathrm{softmax}\left(f_\theta(z_i, e_{Y_t}, t)\right). \tag{7}$$

The head embeds the categorical state, combines it elementwise with the image feature, and uses a learned timestep embedding to modulate its hidden layers. The timestep provides side information about how the auxiliary categorical state was generated. At early corruption levels, the state is more strongly coupled to the sampled candidate; at later levels, it increasingly reflects the terminal prior. Providing t allows the shared head to distinguish these training conditions.

The reconstruction objective uses the full candidate distribution as a soft cross-entropy target:

$$\mathcal{L}_{\mathrm{rec}} = -\sum_{c=1}^{C} s_{ic} \log p_\theta(Y_0 = c \mid z_i, Y_t, t). \tag{8}$$

Sampling supplies structured state augmentation; the full distribution supplies a stable uncertainty-aware target. In information-theoretic terms, the timestep-conditioned predictor targets $p(Y_0 \mid X, Y_t, t)$ rather than $p(Y_0 \mid X, Y_t)$, and conditioning cannot increase uncertainty:

$$H(Y_0 \mid X, Y_t, t) \le H(Y_0 \mid X, Y_t). \tag{9}$$

This inequality does not by itself imply improved classification accuracy for a finite-capacity model. Instead, it provides a simple information-theoretic motivation for exposing the corruption level: the timestep removes ambiguity about the regime under which the auxiliary categorical state was generated. Whether this additional conditioning is useful in practice is evaluated empirically in Section 4.4.

3.5 One-Step Terminal Prediction

At inference, we sample K independent terminal states: $Y_T^{(k)} \sim \mathrm{Cat}(q_\eta), k = 1, \ldots, K$. Each state is denoised directly at T and the probabilities are averaged:

$$\widehat{p}(y \mid x) = \frac{1}{K} \sum_{k=1}^{K} p_\theta(Y_0 = y \mid z, Y_T^{(k)}, T). \tag{10}$$

and predict by $\arg\max_y \widehat{p}(y \mid x)$. The K predictions form a Monte Carlo estimate of the terminal-prior expectation. They are independent and evaluated in one parallel batch, giving a single inference step without sequential reverse transitions. The image encoder is evaluated once per image. Thus K controls the number of terminal samples being averaged, rather than the depth of a reverse chain.

This prediction rule uses the same reconstruction head as training. The image feature supplies query-specific visual information, and terminal states provide categorical inputs without requiring a query label. The output is a class distribution, so it can be read directly from the head at T. Intermediate timesteps retain their role in learning: they expose that shared head to states generated under different corruption conditions.

4 Experiments

4.1 Experimental Setup

Data and protocol. We conduct our analysis on CIFAR-100 (Krizhevsky, 2009). Following the PMD protocol, we use 45,000 images with noisy labels for training and reserve the remaining 5,000 clean images for validation. The PMD-70 training labels have a measured corruption rate of 69.87%. Clean labels of the noisy training subset are never exposed to the training pipeline or used for retrieval and candidate construction. Beyond PMD-70, we evaluate two additional synthetic noise settings on CIFAR-100 and three real-world noisy-label datasets: CIFAR-100N (Wei et al., 2022), Animal-10N, and Clothing1M (Xiao et al., 2015). Retrieval graphs are constructed exclusively from the corresponding noisy training split and never include validation or test images. For the PMD experiments, we report results on the fixed clean validation set. For synthetic noise CIFAR-100 experiments, we select checkpoints on this validation set and evaluate them once on the official test set. CIFAR-100N and Animal-10N use a fixed 15-epoch training schedule, after which the final EMA checkpoint is evaluated once on the test set. Clothing1M uses its official clean validation split for checkpoint selection and its official clean test split for final evaluation. Unless stated otherwise, we report the mean and population standard deviation over seeds 42, 123, and 2026.

Implementation The main configuration freezes CLIP ViT-L/14, uses $T = 1000$, a uniform prior, and $K = 4$ terminal samples. Candidate construction uses $k = 15$, a minimum of three reciprocal matches, five directed neighbors on fallback, $r_{\min} = 0.1$, $\gamma_r = 1$, $\tau_r = 0.07$, $\gamma_f = 0.5$, $M = 5$, and $\lambda_r = 0.8$. PMD-70 candidates support 4.63 classes on average. The 3.65M-parameter head is trained for 15 epochs with batch size 256, Adam at 10^{-3}, three warm-up epochs followed by cosine decay, gradient clipping at 1.0, and EMA decay 0.999. Where validation selection is used, accuracy is the primary criterion, followed by fewer zero-recall classes and higher macro-F1 to break ties. Experiments use a single NVIDIA GeForce RTX 5070 Ti unless stated otherwise.

Baselines. The key control is a parameter-matched MLP with identical features, candidate targets, optimizer, split, and checkpoint rule. We compare with DLD on the same PMD-70 and Animal-10N data splits to assess accuracy and computational cost relative to an iterative label-diffusion baseline. To test whether the gain over the matched MLP depends on external image–text pretraining, we repeat the comparison using a ResNet-50 encoder pretrained with SimCLR on only the 45,000 CIFAR-100 training images, without labels. The encoder is then frozen, and both heads use the same features and candidate targets. We report accuracy, macro-F1, and the geometric mean of class recalls (G-Mean). These matched controls are more diagnostic than comparisons against a weak hard-label classifier: they isolate the categorical training mechanism from retrieval and representation quality.

Table 1: Matched comparison with soft-target classification. Accuracy (%) is mean ± population standard deviation over three seeds. Both heads share features and candidate targets. Δ is the gain in percentage points. U30 and A30 add 30% uniform and asymmetric noise to PMD-35, respectively.

Dataset / noise	Matched MLP	RC-Cat (ours)	Δ
CIFAR-100 / PMD-70	65.33 ± 0.31	$\mathbf{71.29 \pm 0.15}$	$+5.96$
CIFAR-100 / PMD-35	81.27 ± 0.13	$\mathbf{82.67 \pm 0.13}$	$+1.40$
CIFAR-100 / U30	79.31 ± 0.05	$\mathbf{81.32 \pm 0.22}$	$+2.01$
CIFAR-100 / A30	75.76 ± 0.02	$\mathbf{79.29 \pm 0.39}$	$+3.53$
CIFAR-100N	76.91 ± 0.11	$\mathbf{78.07 \pm 0.04}$	$+1.16$
Animal-10N	89.64 ± 0.11	$\mathbf{89.91 \pm 0.36}$	$+0.27$

4.2 Main Results

Denoising improves matched soft-target learning. Table 1 compares RC-Cat with a parameter-matched MLP using the same frozen features and candidate targets. PMD-35 and PMD-70 are evaluated on the fixed clean validation set; the remaining settings use held-out test sets. On PMD-70, RC-Cat improves accuracy from 65.33% to 71.29%, a gain of 5.96 percentage points. Macro-F1 and G-Mean also rise from 0.6481 to 0.7080 and from 61.02% to 67.98%, respectively (Appendix A.1). The accuracy gain is 1.40 points on PMD-35, and 2.01 and 3.53 points when uniform and asymmetric noise are added to PMD-35. On the real-world noisy-label benchmarks, RC-Cat further improves CIFAR-100N by 1.16 points, while the two models are close on Animal-10N (0.27 points). Thus, the advantage is largest under severe PMD noise and remains numerically positive, though smaller, across the other matched settings. Because the features and candidate targets are shared, these comparisons isolate the benefit of categorical training from representation and retrieval quality.

The magnitude of the gain also appears to depend on the noise regime. RC-Cat provides its largest improvement under severe synthetic PMD noise, where the candidate supervision is highly uncertain and training across controlled corruption levels can provide a useful reliability-conditioning signal. The smaller gains on CIFAR-100N and Animal-10N suggest that this inductive bias becomes less pronounced under human annotation noise, whose errors need not follow the controlled corruption structure used during training. On Clothing1M, the nearly identical performance of RC-Cat and the matched classifier may additionally reflect the different large-scale regime: with roughly one million training examples and only 14 classes, strong frozen representations and candidate supervision may already make the classification problem sufficiently constrained that corruption-aware training provides little additional benefit. We therefore view the benefit of RC-Cat as noise- and data-regime dependent rather than uniform across noisy-label settings.

Class-level and representation controls. On the two composite-noise test settings, the accuracy gains are accompanied by approximately 2.3- and 4.3-point improvements in G-Mean, respectively (Appendix A.1). To examine dependence on external image–text pretraining, we replace CLIP with a SimCLR encoder trained without labels on the 45,000 optimization images. With this shared encoder and candidate cache, RC-Cat reaches 48.97% PMD-70 validation accuracy versus 46.43% for the matched MLP; all three paired head seeds favor RC-Cat. This control supports a head-level advantage without CLIP.

Context from conventional noisy-label learning. Table 2 situates our results among published noisy-label learning methods. Because training and evaluation protocols differ, this table provides context rather than a controlled ranking; Table 1 provides the matched comparison that isolates our method's contribution.

4.3 Comparison with Iterative Label Diffusion

Table 3 shows the different operating points of one-step and iterative label diffusion. On the matched PMD-70 validation split, DLD achieves $73.72 \pm 0.11\%$ accuracy, compared with $71.29 \pm 0.15\%$ for RC-Cat. The accuracy comparison is dataset-dependent. DLD leads on PMD-70, whereas RC-Cat achieves $89.91 \pm 0.36\%$ test accuracy on Animal-10N, compared with $88.39 \pm 0.08\%$ for DLD (Appendix A.1). The distinction is clearer in inference cost. Under the same raw-image protocol,

Table 2: CIFAR-100 accuracy (%). Conventional rows use published *test* results (Chen et al., 2023; Zhang et al., 2021); ours use *validation* for PMD-35/70 and *test* for U30/A30 (Table 1). These results use different representation and evaluation protocols and are reported for context only.

Method	PMD-35	PMD-70	U30	A30
Standard	57.68 ± 0.29	39.32 ± 0.43	48.86 ± 0.56	45.85 ± 0.93
Co-teaching+	56.70 ± 0.71	39.53 ± 0.28	52.33 ± 0.64	51.21 ± 0.31
GCE	58.37 ± 0.18	40.01 ± 0.71	52.90 ± 0.53	52.69 ± 1.14
SL	55.20 ± 0.33	40.02 ± 0.85	51.34 ± 0.64	50.18 ± 0.97
LRT	56.74 ± 0.34	45.29 ± 0.43	45.66 ± 1.60	52.04 ± 0.15
CC	59.44 ± 0.33	42.79 ± 1.21	56.58 ± 0.45	54.45 ± 1.22
PLC	60.01 ± 0.43	45.92 ± 0.61	60.09 ± 0.15	56.40 ± 0.34
RC-Cat (ours)	$\mathbf{82.67 \pm 0.13}$	$\mathbf{71.29 \pm 0.15}$	$\mathbf{81.32 \pm 0.22}$	$\mathbf{79.29 \pm 0.39}$

Table 3: PMD-70 accuracy and inference cost. RC-Cat and DLD accuracies are three-seed validation results; LRA accuracy is its published CLIP test result (Chen et al., 2023).

Method	Accuracy (%)	Evaluation	Images/s	Checkpoint (MiB)
RC-Cat (ours)	71.29 ± 0.15	Validation	**311.06**	**13.95**
DLD	73.72 ± 0.11	Validation	170.99	368.19
LRA (CLIP)	$\mathbf{74.52 \pm 0.12}$	Test (published)	292.38	92.11

RC-Cat processes 311.06 images/s, nearly matching the MLP's 311.38 images/s and reaching $1.82\times$ the throughput of DLD DDIM-10. It also optimizes roughly $13\times$ fewer parameters and has a $26\times$ smaller checkpoint. LRA-Diffusion provides a separate reference point. Its published CLIP result is $74.52 \pm 0.12\%$ on the PMD-70 test set (Chen et al., 2023), whereas our PMD-70 results above use validation; the accuracies should therefore not be ranked directly. Appendix A.5 shows the LRA throughput and checkpoint size from our local implementation.

4.4 Mechanism Ablation

Table 4 separates the effects of conditioning on a categorical label state and revealing its corruption level. All three variants share the same features, candidate targets, model capacity, and training schedule. A1 receives a zero state and a fixed timestep input. A2 instead receives a randomly corrupted categorical state, but the sampled timestep remains hidden; this raises PMD-70 validation accuracy from 60.29% to 64.78%. A3 uses the same corruption process and also receives the sampled timestep, further increasing accuracy to 71.29%. Both gains occur at every seed, with macro-F1 and G-Mean following the same pattern.

Table 4: Three-seed mechanism ablation on PMD-70. All configurations use the same frozen features and candidate targets.

Configuration	Accuracy (%)	Macro-F1	G-Mean (%)
A1: zero state, fixed $t = 0$	60.29 ± 0.79	0.5953 ± 0.0087	55.91 ± 1.23
A2: corruption, hidden t	64.78 ± 0.79	0.6424 ± 0.0095	60.82 ± 0.97
A3: corruption, true t	$\mathbf{71.29 \pm 0.15}$	$\mathbf{0.7080 \pm 0.0004}$	$\mathbf{67.98 \pm 0.58}$

Contribution of candidate construction. To assess the role of candidate targets, we retrain the same A3 configuration with different target constructions at seed 123. Using only the observed noisy labels yields 56.52% PMD-70 validation accuracy; plain cosine-KNN soft targets raise it to 68.20%. With reciprocal neighbors, local reliability weighting further raises accuracy from 67.32% to 71.04%, and the complete construction reaches 71.48% (Appendix A.3). In this single-seed comparison, retrieval provides most of the gain over observed labels, while reliability offers a further refinement. This complements Table 1: here we vary the targets under the same denoising setup, whereas there we fix the targets and compare the denoiser with an MLP.

4.5 Denoising Gains Without Reverse-Chain Inference

The ablations above show that conditioning on corrupted label states and their timesteps improves classification. We next ask whether realizing this gain requires iterative refinement at inference. On PMD-70, the best validation-selected one-step checkpoints achieve $71.05 \pm 0.34\%$, compared with $70.68 \pm 0.65\%$ for the best full reverse-chain checkpoints; one-step prediction is at least as accurate at every seed. Together, these results support a corruption-conditioning interpretation. Introducing corrupted categorical states improves performance over the zero-state control, and additionally revealing the corresponding timestep produces a substantially larger gain. It shows that the model benefits from distinguishing among corruption regimes rather than receiving heterogeneous auxiliary states without information about how they were generated.

5 Discussion

What Produces the Gain? The controls isolate a sequence of effects. Sharing the representation and candidate targets with the matched MLP removes the two strongest alternative explanations. In table 4 A1–A2 then isolates categorical state augmentation, while A2–A3 isolates explicit knowledge of the corruption level. Together, these results support a reliability-conditioning interpretation: the model learns how much to use the auxiliary label state, rather than committing to a single corrected target throughout training. The SimCLR control further shows that the relative head advantage persists when the representation is learned only from target images without labels.

Retrieval remains essential to the absolute operating point. The candidate ablation in Appendix A.3 shows that plain KNN supplies most of the improvement over observed labels and local reliability provides the clearest additional gain. The denoising objective and retrieval therefore play different roles: retrieval broadens the supervision, while corruption-aware training determines how that supervision is consumed.

Why Can One Step Be Enough? Iterative refinement is natural for high-dimensional generation because a model must construct many dependent variables. Here, the output is one category and the image representation already carries strong semantic information. The terminal denoiser can directly amortize the mapping from (z, Y_T, T) to a class distribution. Intermediate timesteps still matter during training: they expose the head to a continuum of label-state reliability and discourage copying any single state. This explains how diffusion-style training and one-step inference can coexist without contradiction.

Accuracy–Efficiency Trade-off. DLD attains the highest PMD-70 accuracy, whereas RC-Cat approaches the throughput and storage of an ordinary classifier. On Animal-10N, RC-Cat is both smaller and more accurate. The result is therefore not a universal ranking, but a different operating point: most of the denoising benefit is retained while the sequential reverse process and its deployment cost are removed.

6 Conclusion

We proposed RC-Cat, a retrieval-calibrated one-step categorical denoising framework, which decouples corruption-aware training from iterative reverse-chain generation. RC-Cat constructs sparse uncertainty-aware candidate targets from retrieved neighborhood evidence, trains a timestep-conditioned denoiser on categorically-corrupted label states, and performs one-step terminal prediction.

With features and candidate targets fixed, RC-Cat outperforms a matched soft-target classifier by 5.96 points under PMD-70 severe noise. Our ablation studies show that exposure to corrupted label states together with timestep conditioning delivers most of the denoising improvement, whereas iterative reverse-chain inference yields negligible extra gains. Across the evaluated representations and noise settings, these results show that useful denoising training can coexist with one-step classification, with an accuracy–efficiency trade-off relative to iterative label diffusion.

AI USE STATEMENT

In accordance with ICLR 2027 guidelines, we report the use of a large language model (LLM) during the preparation of this manuscript. Its role was strictly limited to that of a writing aid, assisting with language refinement, improving the clarity and readability of the text, and debugging code. We have not used generative AI tools for research ideation or execution, drafting substantive sections of the paper, or generating synthetic datasets. We have reviewed all AI-assisted work. Specifically, we manually reviewed and revised all AI-assisted language edits, read and verified the original sources of related work suggested by generative AI, independently checked the mathematical arguments against the AI-generated verification reports, and reviewed and tested all AI-assisted code modifications for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 4 specifies all experimental details including dataset partitions, optimization hyper-parameters, evaluation protocols, and hardware setup. Appendix B provides complete pseudocode for offline retrieval-calibrated candidate construction, categorical denoising training, and one-step terminal inference procedures. All reported numerical results are aggregated across multiple fixed random seeds (42, 123, and 2026), and we explicitly state whether validation-based checkpoint selection or fixed-epoch EMA checkpoints are used for each dataset. We report mean together with population standard deviation to quantify result variance. For real-world noisy-label benchmarks CIFAR-100N, Animal-10N and Clothing1M, we strictly follow their official data splits and evaluation rules without accessing any test-set information during training or candidate retrieval. All retrieval graphs are built exclusively from the noisy training split; validation and test images are never included in neighbor search. We will release code implementing candidate target caching, timestep-conditioned denoiser training, and one-step inference upon paper acceptance.

REFERENCES

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242, 2017. URL https://proceedings.mlr.press/v70/arpit17a.html.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Jian Chen, Ruiyi Zhang, Tong Yu, Rohan Sharma, Zhiqiang Xu, Tong Sun, and Changyou Chen. Label-retrieval-augmented diffusion models for learning from noisy labels. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d191ba4c8923ed8fd8935b7c98658b5f-Abstract.html.

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

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 4521–4534, 2023.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465, 2021.

Senyu Hou, Gaoxia Jiang, Jia Zhang, Shangrong Yang, Husheng Guo, Yaqing Guo, and Wenjian Wang. Directional label diffusion model for learning from noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25738–25748, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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. URL https://openreview.net/forum?id=HJgExaVtwr.

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, pp. 20331–20342, 2020.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Javier E. Santos, Zachary R. Fox, Nicholas Lubbers, and Yen Ting Lin. Blackout diffusion: Generative diffusion models in discrete-state spaces. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9034–9059, 2023.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153, 2023. doi: 10.1109/TNNLS.2022.3152527.

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. URL https://openreview.net/forum?id=TBWA6PLJZQm.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2691–2699, 2015.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach, 2021. URL https://arxiv.org/abs/2103.07756.

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*, volume 31, 2018.

A Additional Experimental Results

A.1 Full Classification Results

Table 5 supplements the main accuracy comparison with macro-F1 and G-Mean, and includes the three-seed DLD results. All entries retain the evaluation splits specified in the main text.

Table 5: Full accuracy and class-level results across synthetic and real-world label noise. Entries are mean ± population standard deviation over seeds 42, 123, and 2026. PMD-35 and PMD-70 use the fixed clean validation split; all other groups report test results. The matched MLP and RC-Cat share frozen features and candidate targets. Bold denotes the best result within each group.

Method	Accuracy (%)	Macro-F1	G-Mean (%)
CIFAR-100: PMD-70 (validation)			
Matched MLP	65.33 ± 0.31	0.6481 ± 0.0034	61.02 ± 0.52
DLD	$\mathbf{73.72 \pm 0.11}$	$\mathbf{0.7314 \pm 0.0025}$	$\mathbf{69.47 \pm 0.65}$
RC-Cat (ours)	71.29 ± 0.15	0.7080 ± 0.0004	67.98 ± 0.58
CIFAR-100: PMD-35 (validation)			
Matched MLP	81.27 ± 0.13	0.8089 ± 0.0014	79.50 ± 0.21
RC-Cat (ours)	$\mathbf{82.67 \pm 0.13}$	$\mathbf{0.8239 \pm 0.0016}$	$\mathbf{81.46 \pm 0.20}$
CIFAR-100: PMD-35 + 30% uniform (test)			
Matched MLP	79.31 ± 0.05	0.7930 ± 0.0010	77.67 ± 0.11
RC-Cat (ours)	$\mathbf{81.32 \pm 0.22}$	$\mathbf{0.8121 \pm 0.0026}$	$\mathbf{80.01 \pm 0.26}$
CIFAR-100: PMD-35 + 30% asymmetric (test)			
Matched MLP	75.76 ± 0.02	0.7553 ± 0.0007	73.59 ± 0.18
RC-Cat (ours)	$\mathbf{79.29 \pm 0.39}$	$\mathbf{0.7915 \pm 0.0033}$	$\mathbf{77.87 \pm 0.40}$
CIFAR-100N (test)			
Matched MLP	76.91 ± 0.11	0.7597 ± 0.0015	72.34 ± 0.22
RC-Cat (ours)	$\mathbf{78.07 \pm 0.04}$	$\mathbf{0.7713 \pm 0.0008}$	$\mathbf{73.54 \pm 0.11}$
Animal-10N (test)			
Matched MLP	89.64 ± 0.11	0.8923 ± 0.0012	89.05 ± 0.13
DLD	88.39 ± 0.08	0.8790 ± 0.0008	87.75 ± 0.09
RC-Cat (ours)	$\mathbf{89.91 \pm 0.36}$	$\mathbf{0.8951 \pm 0.0039}$	$\mathbf{89.41 \pm 0.40}$

A.2 Representation and Dataset Controls

A ResNet-50 SimCLR encoder is trained for 200 epochs using only the 45,000 optimization images and no labels. With its frozen features, RC-Cat reaches $48.97 \pm 0.36\%$ accuracy, versus $46.43 \pm 0.08\%$ for the parameter-matched MLP (mean ± sample standard deviation over three seeds). On Clothing1M, the corresponding accuracies are $68.97 \pm 0.33\%$ and $68.92 \pm 0.04\%$. These controls respectively test dependence on external pretraining and behavior at million-example scale.

A.3 Candidate Construction

Table 6: Candidate-target ablation at seed 123 with the denoiser fixed.

Candidate construction	Accuracy (%)
Observed label only	56.52
Directed cosine KNN	68.20
Reciprocal KNN	67.32
Reciprocal + reliability	71.04
Directed + reliability	71.30
Directed + reliability + frequency	71.46
Frozen full target	71.48

Retrieval supplies most of the improvement over observed labels, and local reliability gives the clearest additional gain. The main configuration remains frozen to the reciprocal construction selected before this single-seed audit.

A.4 Additional Training Details

The timestep-modulated head maps the categorical state to the feature dimension, combines it elementwise with the normalized image feature, applies two further conditional layers, and predicts C logits. We also evaluated the exact D3PM posterior KL; its weighted magnitude was approximately 8×10^{-5} and it did not change the selected predictions, so the final objective uses only Eq. (8). Extending a representative run from 15 to 50 epochs did not improve validation accuracy, supporting the fixed 15-epoch schedule.

A.5 Local LRA Reproduction

LRA-Diffusion (Chen et al., 2023) is closely related through its use of retrieved label evidence. Our local LRA DDIM-10 implementation was included in the PMD-70 raw-image profiling study. It obtained 54.56% validation accuracy, 292.38 images/s throughput, 1,891.42 MiB peak allocated memory, 2.56M optimized parameters, and a 92.11 MiB checkpoint. These are measurements of the profiled implementation, not three-seed averages. This reproduction did not recover the published accuracy and is retained as an implementation-level record; it does not support a definitive accuracy comparison with LRA-Diffusion.

Table 7: Complete local PMD-70 raw-image profiling at batch size 100. Accuracy is measured for the profiled checkpoint, not averaged over seeds; optimized parameters exclude the frozen encoder.

Method	Acc. (%)	Images/s	Peak MiB	Optimized	Checkpoint MiB
Matched MLP	65.26	311.38	1813.44	3.65M	13.97
DLD DDIM-10	73.74	170.99	1984.84	48.21M	368.19
LRA DDIM-10	54.56	292.38	1891.42	2.56M	92.11
RC-Cat (ours)	71.50	311.06	1813.04	3.65M	13.95

B Candidate Construction and Learning Algorithms

Algorithm 1 constructs all neighbor sets before computing local reliability, so each neighbor's agreement is evaluated under the same graph rule. It uses only noisy labels from the optimization partition. The candidate support includes the observed annotation even when that class is not among the largest neighbor votes.

Algorithm 1 Offline retrieval-calibrated candidate construction

Require: Noisy optimization set $\widetilde{\mathcal{D}}$; frozen encoder g; $k, m_{\min}, k_f, M$; $\tau_r, r_{\min}, \gamma_r, \gamma_f, \lambda_r$
Ensure: Cached normalized features $\{\mathbf{z}_i\}$ and candidate targets $\{\mathbf{s}_i\}$

for $i = 1, \dots, N$ **do**
  $\mathbf{z}_i \leftarrow g(x_i)/\|g(x_i)\|_2$
end for
for $i = 1, \dots, N$ **do**
  $N_k(i) \leftarrow$ indices of the k largest $\mathbf{z}_i^\top \mathbf{z}_j$, excluding i
  $\mathcal{M}_i \leftarrow \{j \in N_k(i) : i \in N_k(j)\}$
  if $|\mathcal{M}_i| < m_{\min}$ **then**
    $\mathcal{M}_i \leftarrow$ first k_f elements of $N_k(i)$
  end if
end for
for $j = 1, \dots, N$ **do**
  $a_j \leftarrow |\mathcal{M}_j|^{-1} \sum_{\ell \in \mathcal{M}_j} \mathbb{I}(\widetilde{y}_\ell = \widetilde{y}_j)$
  $r_j \leftarrow [\max(a_j, r_{\min})]^{\gamma_r}$
end for
$n_c \leftarrow \sum_i \mathbb{I}(\widetilde{y}_i = c)$ for all c; $\bar{n} \leftarrow N/C$
for $i = 1, \dots, N$ **do**
  $v_{ic} \leftarrow 0$ for all c
  for $j \in \mathcal{M}_i$ **do**
    $w_{ij} \leftarrow \exp(\mathbf{z}_i^\top \mathbf{z}_j/\tau_r) r_j$
    $v_{i,\widetilde{y}_j} \leftarrow v_{i,\widetilde{y}_j} + w_{ij}$
  end for
  $v_{ic} \leftarrow v_{ic}/(n_c/\bar{n})^{\gamma_f}$ for all c
  $S_i \leftarrow \mathrm{TopM}(\mathbf{v}_i) \cup \{\widetilde{y}_i\}$
  $Z_i \leftarrow \sum_d \mathbb{I}(d \in S_i) v_{id}$
  if $Z_i > 0$ **then**
    $\rho_{ic} \leftarrow \mathbb{I}(c \in S_i) v_{ic}/Z_i$ for all c
  else
    $\boldsymbol{\rho}_i \leftarrow \mathbf{e}_{\widetilde{y}_i}$
  end if
  $\mathbf{s}_i \leftarrow (1 - \lambda_r)\mathbf{e}_{\widetilde{y}_i} + \lambda_r \boldsymbol{\rho}_i$
end for
Cache $\{\mathbf{z}_i, \mathbf{s}_i\}_{i=1}^N$

Algorithm 2 samples the cumulative marginal directly. Validation-based selection applies to the PMD, composite-noise, and Clothing1M protocols. CIFAR-100N and Animal-10N instead retain the final EMA checkpoint after 15 epochs. For validation-based selection, the terminal sampling seed is fixed across epochs, to keep terminal sampling consistent during checkpoint comparison.

Algorithm 2 Categorical training and direct terminal prediction

Require: Cached pairs $\{(\mathbf{z}_i, \mathbf{s}_i)\}$; prior q_η; cumulative schedule $\{\bar{\alpha}_t\}_{t=1}^T$; epochs E; terminal samples K; EMA decay μ; gradient limit G; dataset-specific selection rule
Ensure: Selected EMA parameters $\theta^\star$

Initialize θ and $\bar{\theta} \leftarrow \theta$; best validation key $b \leftarrow (-\infty, -\infty, -\infty)$
for $e = 1, \dots, E$ **do**
  for mini-batch B **do**
    for $i \in B$ **do**
      Sample $t_i \sim \mathrm{Uniform}\{1, \dots, T\}$
      $\mathbf{u}_i \leftarrow \bar{\alpha}_{t_i} \mathbf{s}_i + (1 - \bar{\alpha}_{t_i}) q_\eta$
      Sample $Y_{t_i,i} \sim \mathrm{Cat}(\mathbf{u}_i)$
      $\mathbf{p}_i \leftarrow \mathrm{softmax}\, f_\theta(\mathbf{z}_i, \mathbf{e}_{Y_{t_i,i}}, t_i)$
    end for
    $\mathcal{L} \leftarrow -|B|^{-1} \sum_{i \in B} \sum_c s_{ic} \log p_{ic}$
    Reset gradients; backpropagate $\mathcal{L}$; clip gradient norm to G
    Update θ using Adam and the scheduled learning rate
    $\bar{\theta} \leftarrow \mu \bar{\theta} + (1 - \mu)\theta$
  end for
  if the protocol uses validation selection and e is a validation epoch **then**
    Evaluate OneStepPredict$(\bar{\theta}, K, \omega_{\mathrm{eval}})$ on validation data
    $b_e \leftarrow (\mathrm{accuracy}, -\mathrm{zerorecall\ classes}, \mathrm{macroF1})$
    if b_e is lexicographically greater than b **then**
      $(b, \theta^\star) \leftarrow (b_e, \bar{\theta})$
    end if
  end if
end for
if the protocol uses the final checkpoint **then**
  $\theta^\star \leftarrow \bar{\theta}$
end if

procedure OneStepPredict(θ', K, ω)
  for input image x **do**
    $\mathbf{z} \leftarrow g(x)/\|g(x)\|_2$
    Sample $Y_T^{(1)}, \dots, Y_T^{(K)} \overset{\mathrm{iid}}{\sim} \mathrm{Cat}(q_\eta)$ using generator ω
    Evaluate $f_{\theta'}(\mathbf{z}, \mathbf{e}_{Y_T^{(k)}}, T)$ for all k in one batch
    $\widehat{p}(\cdot \mid x) \leftarrow K^{-1} \sum_k \mathrm{softmax}\, f_{\theta'}(\mathbf{z}, \mathbf{e}_{Y_T^{(k)}}, T)$
    $\widehat{y}(x) \leftarrow \arg\max_c \widehat{p}(c \mid x)$
  end for
  return $\{\widehat{y}(x), \widehat{p}(\cdot \mid x)\}$
end procedure