Reflection as a Criterion for Active Learning

Anonymous authors
Paper under double-blind review

Abstract

Active learning algorithms are commonly evaluated by classification error, but directly evaluating data acquisitions by their effect on expected classification error is generally difficult. Many principled acquisition criteria are therefore derived for cross-entropy, information gain, or model uncertainty. We propose reflection as an intermediate criterion for this modeling problem: it scores a candidate batch by holding the previously acquired and candidate instances fixed and assessing their usefulness under alternative plausible label realizations. Given the current posterior over model parameters, reflection samples plausible models, resamples labels for both previously selected and candidate instances, retrains on the relabeled set, and evaluates the resulting classifier under the sampled model. This formulation is closely aligned with the downstream objective, remains meaningful under point-estimate posteriors and is amenable to asymptotic analysis. Under cross-entropy, reflection provides a derivation of the Fisher-information-ratio objective as used by BAIT, explaining why BAIT evaluates this ratio on the entire selected set comprising both previous and candidate instances. To analyze reflection under classification error, we derive the asymptotic expected zero-one loss for histogram classifiers. The formula yields a budget-dependent shift from coverage-oriented to difficulty-driven acquisition. A kernelized relaxation turns this structure into a practical strategy that is competitive with strong baselines, including Margin, BADGE and TypiClust and robust to large batches, label noise, and duplicates.

1 Introduction

In pool-based active learning, a query strategy selects unlabeled instances for annotation over several rounds, with the goal of achieving high predictive performance under a fixed labeling budget (Cohn et al., 1994; Settles & Craven, 2008; Herde et al., 2021). Although such strategies are commonly evaluated by classification error, many principled acquisition criteria are derived from cross-entropy, information gain, uncertainty, or other surrogate objectives; see Kirsch & Gal (2022) for an overview.

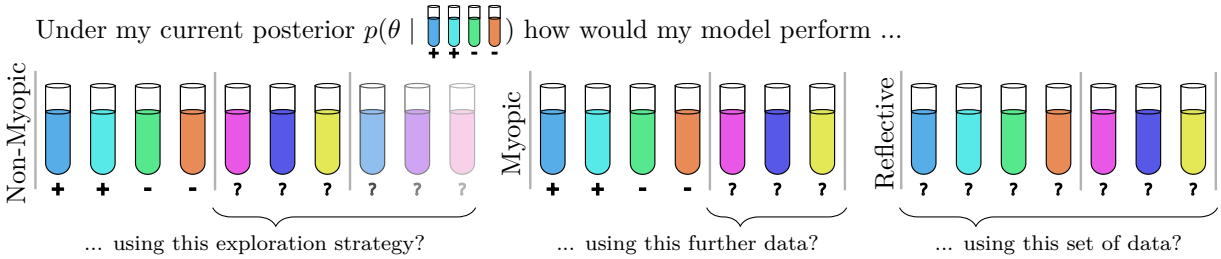


Figure 1: Comparison of non-myopic, myopic, and reflective decision making. Test tubes represent instances, $+/-$ signs represent observed labels and ? signs represent unobserved or hypothetically resampled labels, fading test tubes represent potential future queries. Non-myopic policies evaluate a candidate batch considering potential future query decisions, myopic policies evaluate the immediate effect of the candidate batch, and reflection evaluates the selected and candidate instances under hypothetical relabelings based on the current posterior.

A decision-theoretic optimum for minimizing final generalization error is non-myopic active learning, which evaluates an acquisition by its expected value within a future sequence of querying decisions. This formulation directly targets final generalization performance, but solving the resulting Bellman equation (Bellman, 1954) scales exponentially with the planning horizon (Low et al., 2009; Hoang et al., 2014; Huan et al., 2024) and is infeasible for most applied active learning settings. Myopic criteria are comparatively cheaper (Kottke et al., 2021; Huan et al., 2024). They only evaluate the next acquisition relative to the currently observed labeled data and do not plan over future acquired labels (Cohn et al., 1996; Settles et al., 2007; Roy & McCallum, 2001). Among the myopic methods, those that estimate performance improvement most directly usually require retraining for many possible label outcomes, making them costly in batch settings (Huseljic et al., 2026).

Reflection addresses the problem from a different angle while remaining myopic in the acquisition horizon. It fixes the previously acquired and candidate instances, but evaluates them under plausible alternative label realizations. Thereby the observed labels influence the score through the posterior, rather than as fixed outcomes. Thus, reflection asks whether the selected instances would be useful beyond the particular labels observed so far. Figure 1 visualizes this distinction.

Reflection is not meant to be implemented by exhaustively relabeling, retraining, and evaluating every candidate batch. Rather, it exposes a design problem that can be approximated analytically. We use *design* to denote the selected instances for which labels are acquired, independent of the realized label values. In our setting, the design consists of the previously acquired and the candidate instances.

The criterion remains useful when the posterior must be approximated crudely, as is common for neural networks. Even if the posterior is collapsed to a single maximum-a-posteriori (MAP) estimate, reflection remains nontrivial: it compares candidate designs by their expected training value under labels generated from the current predictive model, rather than relying on posterior variability alone. Under cross-entropy, the point-estimate version of reflection provides a criterion-level rationale for the full-design Fisher-information-ratio (FIR) objective used by BAIT (Ash et al., 2021), where the design includes both previously acquired and newly queried instances rather than only the candidate batch. We then ask what reflection implies under zero-one loss for histogram classifiers (Lugosi & Nobel, 1996).

Contributions

- **Reflection as a criterion.** We introduce *reflection*, a criterion that evaluates candidate batches under posterior-consistent relabelings of the selected design. For cross-entropy with a MAP posterior approximation, this gives a criterion-level derivation of BAIT's full-design FIR objective and clarifies its distinction from related FIR-based strategies.
- **Reflection as a criterion under zero-one loss.** We analyze reflection for histogram classifiers and derive asymptotic expressions for the expected zero-one loss.
- **A practical kernelized query strategy.** We use the resulting local-counting principle to construct *Reflective Binning*, a kernelized relaxation in representation space, and show its robustness to label noise, duplicates, and large batches.

2 Related Work

Representation-based strategies. Many query strategies select instances that represent the geometry or density of the unlabeled pool. Coreset (Sener & Savarese, 2018) methods minimize covering radii, density-aware variants such as ProbCover (Yehuda et al., 2022) select points that cover high-probability regions, and TypiClust (Hacohen et al., 2022) combines clustering with local typicality. Kernelized variants replace hard geometric neighborhoods by smoother similarity measures (Bae et al., 2025). These geometric criteria are often motivated as proxies or upper-bound surrogates for downstream generalization error. Yet the corresponding objective is usually label-agnostic, depending on the feature distribution or representation geometry rather than on labels observed during active learning. In our framework, this is analogous to a prior-only form of reflection, before the posterior is updated by label information.

Informativeness-based strategies. A complementary line of work queries instances whose labels are expected to be informative for the current model. Bayesian approaches include Query by Committee (Seung et al., 1992), which selects instances on which posterior-sampled models disagree, and information-gain criteria such as BALD (Houlsby et al., 2011) and BatchBALD (Kirsch et al., 2019). A more heuristic but widely used family is uncertainty sampling, including least-confidence, entropy, and margin (Culotta & McCallum, 2005; Settles & Craven, 2008). These methods are computationally attractive, but their scores are usually surrogate notions of informativeness rather than direct estimates of downstream classification performance. More directly outcome-oriented criteria score candidates by their anticipated effect on the trained model. Expected error reduction (Roy & McCallum, 2001) estimates the reduction in future prediction error, while expected model change (Freytag et al., 2014) measures how strongly retraining would alter the current model. In deep learning, BADGE (Ash et al., 2020) combines informativeness and diversity through gradient embeddings. These methods expose a tradeoff. Criteria close to downstream model improvement are attractive, but can be expensive when they require candidate-wise retraining. Reflection keeps the outcome-oriented motivation, while shifting the evaluation from individual retraining effects to the selected set as a training design. This design view lets us replace retraining simulations by asymptotic design scores.

Fisher-information-based selection. FIR objectives have a long connection to active learning (Zhang & Oles, 2000; Hoi et al., 2006; Chaudhuri et al., 2015). They are typically motivated through asymptotic efficiency arguments for the cross-entropy, where the selected design should provide Fisher information that is useful for the evaluation distribution. BAIT (Ash et al., 2021) is the most relevant practical instance for our work, because it evaluates the FIR objective on the full selected design, including both previously acquired and newly queried instances. In Section 3.4, we show that this full-design structure arises naturally as a tractable approximation to reflection under cross-entropy with a MAP posterior approximation.

Allocation views and bandit active learning. A related line of work studies active learning in multi-armed bandits, where a finite sampling budget is allocated across arms in order to estimate arm-specific quantities accurately (Antos et al., 2008; Carpentier et al., 2011). These methods are close in spirit to our later histogram analysis: bins play the role of arms, and the query strategy determines how many labels are collected from each bin. In both cases, one can ask what allocation would be optimal if the relevant distributional quantities were known, and then design adaptive procedures that estimate these quantities from data. The main difference is the loss. Bandit active learning work typically focuses on mean estimation or regression-type objectives, for which optimal allocations are governed by variance. In our zero-one setting, the relevant quantity is instead the probability that the empirical majority label in a bin differs from the Bayes-optimal label.

3 Active Learning as a Decision Process

We first formalize pool-based active learning as a sequential decision problem and relate the resulting notation to static optimal experimental design. We then introduce reflection as an intermediate criterion between non-myopic and myopic decision making, before showing how the cross-entropy case leads to the full-design FIR objective used by BAIT.

3.1 Setup and Formalization

We consider a pool-based active learning setting for classification with feature space $\mathcal{X}$, finite label space $\mathcal{Y}$, and a model class Θ that defines conditional probabilities $p_\theta(y \mid x)$ and predictions $\mathrm{clf}_\theta(x) = \arg\max_{y\in\mathcal{Y}} p_\theta(y \mid x)$. We assume realizability, i.e., there exists $\theta^\star \in \Theta$ such that the true conditional distribution satisfies $p(y \mid x) = p_{\theta^\star}(y \mid x)$.

Active learning starts from an initial batch $B_0 = (x_{0,1}, \ldots, x_{0,b_0})$ and proceeds in acquisition cycles. At cycle $i \geq 1$, a query strategy selects a batch $B_i = (x_{i,1}, \ldots, x_{i,b_i}) \in \mathcal{X}^{b_i}$ of b_i instances to be labeled. After B_i is selected, labels $Z_i = (y_{i,1}, \ldots, y_{i,b_i})$ are drawn independently, conditional on the selected batch and the previously observed data, with $y_{i,j} \sim p_{\theta^\star}(\cdot \mid x_{i,j})$. Using $\oplus$ for concatenation, we denote by $X_i = B_0 \oplus \cdots \oplus B_i$ the sequence of selected instances, by $Y_i = Z_0 \oplus \cdots \oplus Z_i$ the corresponding sequence of observed labels, and by

$D_i = (X_i, Y_i)$ the labeled data available after cycle i. The number of labeled instances after cycle i is denoted by $n_i = \sum_{r=0}^{i} b_r$. Formally, a deterministic query strategy at cycle i is a map $q_i : \mathcal{X}^{n_{i-1}} \times \mathcal{Y}^{n_{i-1}} \to \mathcal{X}^{b_i}$, with $B_i = q_i(D_{i-1})$. For simplicity, we omit explicit pool constraints in the formal objectives and write candidate batches as elements of $\mathcal{X}^{b_i}$. Given labeled data D, we denote by $\mathrm{train}(D)$ the parameter obtained by training on D, for example a maximum-likelihood or MAP estimate depending on the setting. After the final cycle k, the learned parameter is $\hat{\theta} = \mathrm{train}(D_k)$. For a data-generating parameter θ and a trained parameter $\hat{\theta}$, we write $R_\theta(\hat{\theta})$ for the evaluation risk under a fixed evaluation distribution, with cross-entropy and zero-one loss as the main examples. The active learning goal is to choose query strategies that minimize the expected final risk $R_{\theta^\star}(\mathrm{train}(D_k))$ under the labeling budget n_k.

3.2 Relation to Optimal Experimental Design

This formulation connects to optimal experimental design (OED), where the goal is to choose data points that make training statistically efficient. OED typically considers a single-stage acquisition process (Huan et al., 2024), corresponding in our notation to the case $k = 1$. Under this reduction, active learning becomes a non-sequential design problem. This static reduction was studied by Chaudhuri et al. (2015), who show that asymptotic optimality for cross-entropy active learning can be achieved by a two-stage procedure: first estimate the parameter from the initial labeled batch, then solve the static design problem for the remaining budget. Here, asymptotic optimality means asymptotically matching the loss of an oracle that selects all instances in advance knowing $\theta^\star$. We discuss related notions of optimality in App. A. A key tool behind this perspective is the following classical asymptotic characterization of the cross-entropy risk.

Theorem 1. *Let $p_{\theta^\star}(y \mid x)$ be the ground truth conditional distribution and $p_{\mathrm{eval}}(x)$ the distribution of the instances during evaluation, $p_{\mathrm{train}}(x)$ the distribution of the instances used for training, $\hat{\theta} = \arg\min_\theta \sum_{i=1}^{n} -\ln p_\theta(y_i \mid x_i)$, where $x_i \sim p_{\mathrm{train}}(x)$ and $y_i \sim p_{\theta^\star}(y \mid x_i)$ independently for all i. Then*

$$\mathbb{E}_{\hat{\theta},x}[D_{\mathrm{KL}}(p_{\theta^\star}(\cdot \mid x) \parallel p_{\hat{\theta}}(\cdot \mid x))] \sim \frac{\mathrm{tr}(I_{\mathrm{train}}^{-1} I_{\mathrm{eval}})}{n}, \tag{1}$$

where D_{KL} denotes the Kullback-Leibler divergence and I_{train} and I_{eval} are the Fisher-information matrices with respect to the distributions $p_{\theta^\star}(y \mid x) \cdot p_{\mathrm{train}}(x)$ and $p_{\theta^\star}(y \mid x) \cdot p_{\mathrm{eval}}(x)$.

Theorem 1 gives a static design principle: the expected excess cross-entropy risk decays as $O(1/n)$, with a constant determined by the FIR $\mathrm{tr}(I_{\mathrm{train}}^{-1} I_{\mathrm{eval}})$. Thus, the training distribution directly controls statistical efficiency. However, active learning is sequential: later batches depend on labels observed earlier, whereas the theorem assumes a fixed training distribution. This creates a mismatch between the single-stage design setting and the adaptive setting in which query strategies are used. This motivates reflection, defined next, which applies the design perspective inside the active learning loop by evaluating the full selected design under the current posterior.

3.3 Reflection

Suppose we are at cycle $i \geq 1$, selecting the next batch, with labeled data $D_{i-1} = (X_{i-1}, Y_{i-1})$. From a decision-theoretic perspective, the optimal action would minimize the expected final risk after the current and all future acquisitions. A non-myopic policy would choose the current batch jointly with future query strategies so as to minimize the expected final risk. This is generally infeasible because it requires optimizing over future query rules and integrating over their possible label outcomes.

A myopic policy avoids this difficulty by evaluating only the immediate acquisition step, keeping the already observed labels fixed and averaging only over the unknown labels of the candidate batch (Kottke et al., 2021; Kottke, 2021). Reflection modifies this view: instead of treating past labels as fixed, it evaluates the enlarged selected design $X_{i-1} \oplus B$ under hypothetical relabelings of both previously selected and newly proposed instances. This distinction is illustrated in Figure 1, where test tubes represent instances and $+$, $-$, and ? indicate observed, unknown, or hypothetically resampled labels. Formally, reflection selects

$$\arg\min_{B \in \mathcal{X}^{b_i}} \mathbb{E}_{\theta \sim p(\theta \mid D_{i-1})} \mathbb{E}_{\widetilde{Y} \sim p_\theta(\cdot \mid X_{i-1} \oplus B)} \left[R_\theta\Big(\mathrm{train}(X_{i-1} \oplus B, \widetilde{Y})\Big) \right]. \tag{2}$$

The observed labels Y_{i-1} affect this score only through the posterior $p(\theta \mid D_{i-1})$. They are not reused as fixed labels in the hypothetical training step. Although this departs from the standard Bayesian perspective, where observed data are fixed by conditioning, it is natural from an estimation viewpoint: the selected instances should be useful under the data-generating process, not only under the particular labeling observed so far.

This criterion is motivated by the corresponding oracle design problem. If $\theta^\star$ were known, one could choose a design $X \in \mathcal{X}^n$ by solving

$$\arg\min_{X \in \mathcal{X}^n} \mathbb{E}_{Y \sim p_{\theta^\star}(\cdot \mid X)} \left[R_{\theta^\star}(\mathrm{train}(X, Y)) \right]. \tag{3}$$

This oracle design problem separates two ingredients. First, one needs to understand how the expected risk depends on the selected design when the data-generating parameter is fixed. For cross-entropy, this dependence is characterized by Theorem 1 through the FIR. If the relevant estimator is consistent under the sampling process, this static criterion can be evaluated at an estimated parameter, as in two-stage analyses such as Chaudhuri et al. (2015). Second, one needs to turn the resulting design criterion into an active learning rule by evaluating it under the information available at the current cycle. Reflection does this by averaging over the current posterior, or by using approximations such as a MAP point estimate.

In the cross-entropy case, the first ingredient is already provided by classical asymptotics, leading to the FIR approximation discussed below. The main new theoretical question in this paper is the corresponding one for zero-one loss: how does the expected zero-one loss depend on the selected design? We address this question for histogram classifiers in Section 4, and then use the resulting structure to motivate a kernelized query strategy.

3.4 Cross-Entropy Approximation and BAIT

For cross-entropy loss, we approximate equation 2 by replacing the posterior with a point mass at the MAP estimate $\hat{\theta}_{i-1}$ and applying Theorem 1 to the relabeled design $X_{i-1} \oplus B$. This yields

$$\arg\min_{B \in \mathcal{X}^{b_i}} \mathrm{tr}\Big(I_{X_{i-1} \oplus B}(\hat{\theta}_{i-1})^{-1} I_{\mathrm{eval}}(\hat{\theta}_{i-1}) \Big), \tag{4}$$

where $I_{X_{i-1} \oplus B}(\hat{\theta}_{i-1})$ denotes the Fisher information induced by the selected design and $I_{\mathrm{eval}}(\hat{\theta}_{i-1})$ the Fisher information induced by the evaluation distribution. This is precisely the design structure used by BAIT (Ash et al., 2021). Reflection therefore gives a criterion-level explanation of why BAIT evaluates the FIR objective on the full selected design, rather than only on the newly queried batch as in related FIR-based approaches (Chaudhuri et al., 2015; Zhang & Oles, 2000; Hoi et al., 2006). This means we can subsume BAIT as a method using the reflection approach.

4 Reflection for the Zero-One Loss of Histogram Classifiers

We now study the oracle-design question for zero-one loss in a tractable histogram model. Here, the selected design is summarized by the number of queried instances in each bin, yielding a bin-allocation problem analogous to pure-exploration bandit allocation but governed by classification error rather than mean-estimation error.

Histogram model and risks. Let $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_m$ be a partition of the feature space. A histogram model assigns to each bin r a class distribution $\rho_r \in \Delta_c$, where Δ_c is the probability simplex over the c labels, equivalently, $\Theta = \Delta_c^m$. For each bin, let $w_r = P_{\mathrm{eval}}(X \in \mathcal{X}_r)$ denote its evaluation weight. Within each bin, we relabel classes in decreasing order of their true probability and write

$$\rho_{r,1} > \rho_{r,2} > \cdots > \rho_{r,c} > 0.$$

The ordering is a notational convention and may differ across bins. The strict inequalities are assumed to avoid tie cases in the asymptotic analysis.

For a selected design $X = (x_1, \ldots, x_n)$, let $\nu_r = \sum_{t=1}^n \mathbf{1}\{x_t \in \mathcal{X}_r\}$ be the number of selected instances in bin r. Training is binwise: the empirical class distribution $\hat{\rho}_r$ is estimated from the labels observed in bin r,

and the induced classifier predicts the empirical majority class $\hat{y}_r = \arg\max_j \hat{\rho}_{r,j}$. Thus, conditional on the design, the expected risk depends on X only through the bin counts $\nu_1, \ldots, \nu_m$.

We consider two error criteria. The expected stochastic zero-one risk is

$$\mathbb{E}[R_{\text{soft}}(\hat{\rho})] = \sum_{r=1}^{m} w_r \sum_{j=1}^{c} (1 - \rho_{r,j})\, P(\hat{y}_r = j). \tag{5}$$

Indeed, if the learned classifier predicts label j in bin r, then its error probability under the stochastic label distribution is $1 - \rho_{r,j}$. We also consider the hard decision error

$$\mathbb{E}[R_{\text{hard}}(\hat{\rho})] = \sum_{r=1}^{m} w_r\, P(\hat{y}_r \neq 1). \tag{6}$$

The hard risk ignores irreducible label noise within a bin and measures only whether the histogram classifier identifies the Bayes-optimal class. This gives a sharper signal for decision uncertainty and is the version used later for the practical query strategy.

Asymptotic zero-one risk. The risks above are controlled by the binwise probabilities $P(\hat{y}_r = j)$. These probabilities correspond to large-deviation events for the empirical class distributions $\hat{\rho}_r$. A large deviation analysis based on Sanov's theorem shows that the dominant term for each bin is given by the probability that class 2 is observed to be most frequent. This probability in turn is determined by the event that the observed frequency of class 2 is larger than that of class 1. We can then apply the Bahadur–Rao formula for this event to obtain the following expansion. The full derivation is given in App. B.

Theorem 2. *Assume that $\rho_{r,1} > \rho_{r,2} > \cdots > \rho_{r,c}$ for every bin r and that $\nu_r \to \infty$. For each bin,*

$$P(\hat{y}_r = 2) \sim P(\hat{y}_r \neq 1) \sim P(\hat{\rho}_{r,1} \leq \hat{\rho}_{r,2}) \sim \frac{C_r}{\sqrt{\nu_r}} e^{-\kappa_r \nu_r}, \tag{7}$$

while the contributions from labels $j > 2$ are exponentially smaller, with

$$\kappa_r = -\log\Big(1 - \big(\sqrt{\rho_{r,1}} - \sqrt{\rho_{r,2}}\big)^2\Big), \qquad C_r = \left(1 - \sqrt{\rho_{r,2}/\rho_{r,1}}\right) \sqrt{\frac{4\pi\sqrt{\rho_{r,1}\rho_{r,2}}}{1 - \big(\sqrt{\rho_{r,1}} - \sqrt{\rho_{r,2}}\big)^2}}.$$

Consequently,

$$\mathbb{E}[R_{\text{soft}}(\hat{\rho})] - R^{\star}_{\text{soft}} \sim \sum_{r=1}^{m} w_r (\rho_{r,1} - \rho_{r,2}) \frac{C_r}{\sqrt{\nu_r}} e^{-\kappa_r \nu_r}, \tag{8}$$

$$\mathbb{E}[R_{\text{hard}}(\hat{\rho})] \sim \sum_{r=1}^{m} w_r \frac{C_r}{\sqrt{\nu_r}} e^{-\kappa_r \nu_r}, \tag{9}$$

where $R^{\star}_{\text{soft}} = \sum_{r=1}^{m} w_r (1 - \rho_{r,1})$ is the Bayes risk of the optimal histogram classifier. Ties in the empirical majority vote are counted as errors. Random tie-breaking would slightly change the prefactor.

The rate κ_r is the main difficulty parameter of bin r. It is small when the two most likely classes are hard to distinguish and large when identifying the most likely class is easy. The constant C_r is the Bahadur–Rao prefactor. Thus, the expected zero-one loss is governed both by the evaluation weight w_r and by the top-two class probabilities that determine the bin's decision difficulty. Theorem 2 is the analogue of the cross-entropy asymptotic formula in Theorem 1 for the zero-one loss in the histogram setting.

Implications for allocation. Theorem 2 implies two qualitatively different ways of allocating the label queries across bins depending on the total budget n. In the late-stage regime, i.e., for large n, the exponential terms dominate. For a fixed large budget $\sum_{r=1}^{m} \nu_r = n$, the leading allocation balances the exponents $\kappa_r \nu_r$ across bins and therefore satisfies

$$\nu_r \propto \frac{1}{\kappa_r}.$$

Thereby the late-stage allocation is governed by decision difficulty rather than bin density. The weights w_r and prefactors C_r enter only through lower-order corrections. The corresponding exponential decay rate of the loss is

$$\frac{1}{\sum_{r=1}^{m} 1/\kappa_r}.$$

In comparison a fixed uniform allocation would correspond to a smaller decay rate of $\min_{1\le r\le m} \frac{\kappa_r}{m}$.

In a low-budget approximation, the exponential factors vary less strongly and the polynomial prefactors dominate. Ignoring the exponential variation gives the hard-risk allocation

$$\nu_r \propto (w_r C_r)^{2/3},$$

and the soft-risk allocation

$$\nu_r \propto \big(w_r(\rho_{r,1} - \rho_{r,2})C_r\big)^{2/3}.$$

Note that the density enters sublinearly. Thereby high-weight bins receive more queries, but the allocation is more diverse than density-proportional sampling. This budget-dependent shift is consistent with the observation of Hacohen et al. (2022) that the query behavior required in active learning can change substantially between low- and high-budget regimes. For comparison, in the same histogram model the leading cross-entropy excess risk yields an allocation proportional to $w_r^{1/2}$, which does not depend on the binwise class probabilities and not on the budget.

The rate κ_r is governed by the Hellinger-type separation $\sqrt{\rho_{r,1}} - \sqrt{\rho_{r,2}}$ between the two most likely classes. Thus, in the histogram setting, a margin-like top-two separation is not merely a heuristic uncertainty score: it arises directly as the large-deviation difficulty of identifying the Bayes-optimal label from noisy samples. This provides an aleatoric interpretation of the intuition behind margin-based uncertainty sampling (Scheffer et al., 2001; Roth & Small, 2006; Settles & Craven, 2008). Exact characterizations of the allocation asymptotics are given in Prop. 5 in App. C.

Posterior evaluation. The analysis above gives the oracle design criterion for fixed bin distributions ρ_r. Reflection turns this into an active learning criterion by evaluating the binwise quantities under the information available at the current cycle. A plug-in version replaces ρ_r by a point estimate, while a Bayesian version averages over the posterior. For Dirichlet posteriors, the exponential part of the binwise error can still be characterized by a large-deviation rate, so the Bayesian objective has the same structural form as the plug-in criterion: a sum of binwise terms that decay exponentially with the allocated count ν_r. The rate itself is different, because posterior uncertainty is integrated rather than collapsed to a point estimate. We defer the corresponding large-deviation result to App. D and use this shared structure in the allocation objective below.

5 From Histogram Asymptotics to a Practical Query Strategy

The histogram analysis gives a local-counting principle for zero-one reflection: a selected design is useful when it places enough labels in regions where the empirical majority decision is difficult. We now turn this principle into a practical query strategy. We first formulate the finite-bin allocation problem suggested by the asymptotics, then discuss how the binwise quantities are estimated, and finally replace hard bins by soft kernel neighborhoods in representation space.

5.1 From Rates to a Bin-Allocation Objective

Suppose first that the feature space is partitioned into bins as in Section 4. At acquisition cycle i, let l_r denote the current number of selected instances in bin r, and let u_r denote the largest final count that can be assigned to this bin under the available pool. The next batch has size b_i, so the final count vector after the batch must satisfy

$$l_r \le \nu_r \le u_r, \qquad \sum_{r=1}^{m} \nu_r = n_{i-1} + b_i.$$

The hard-risk asymptotic from Theorem 2 suggests binwise terms

$$e^{-\phi_r(\nu)} := \widehat{A}_r \nu^{-1/2} e^{-\widehat{\kappa}_r \nu},$$

where $\widehat{\kappa}_r$ is the estimated decision-difficulty rate and $\widehat{A}_r$ contains the bin weight and prefactor. In the plug-in hard-risk criterion, $\widehat{A}_r = \widehat{w}_r \widehat{C}_r$. The Bayesian variant changes how these binwise quantities are computed, but keeps the same allocation layer. Thus the bin target allocation is

$$\min_{\nu_1,\ldots,\nu_m \in \mathbb{N}} \sum_{r=1}^{m} e^{-\phi_r(\nu_r)} \qquad \text{s.t.} \qquad l_r \le \nu_r \le u_r, \quad \sum_{r=1}^{m} \nu_r = n_{i-1} + b_i. \tag{10}$$

We solve equation 10 by its continuous relaxation, similar to Chaudhuri et al. (2015). This allows $\nu_r \in [l_r, u_r]$. For fixed positive $\widehat{A}_r$ and $\widehat{\kappa}_r$, each binwise term is decreasing and convex in $\nu_r > 0$, so the relaxation is a separable convex resource-allocation problem with box constraints and one equality constraint. This structure makes the relaxation efficient to solve: the Karush-Kuhn-Tucker (KKT) conditions reduce the problem to finding a single Lagrange multiplier together with the active lower and upper bounds. We use an active-set procedure for this inner problem and give the derivation and implementation details in App. E.

5.2 Plug-In and Bayesian Bin Scores

The allocation objective in equation 10 requires binwise estimates of the rate and prefactor. In the plug-in variant, we replace the unknown bin distribution ρ_r by a point estimate $\widehat{\rho}_r$ and compute the quantities $\widehat{\kappa}_r$ and $\widehat{C}_r$ from Theorem 2. For the hard-risk criterion this gives

$$\phi_r(\nu) = \widehat{\kappa}_r \nu + \frac{1}{2} \log \nu - \log(\widehat{w}_r \widehat{C}_r).$$

The Bayesian variant keeps the same allocation layer but averages the binwise quantities over the posterior. App. D shows that posterior averaging preserves the exponential-in-count structure of the objective, although the exact rate differs from the plug-in rate. In practice, instead of solving this large-deviation problem in every bin, we use a stable Monte Carlo surrogate heurisitc. For posterior samples $\rho^{(1)}, \ldots, \rho^{(S)} \sim \Pi_r$, let $\kappa(\rho^{(s)})$ and $C(\rho^{(s)})$ be the plug-in quantities computed from the top two class probabilities of the sample. We set

$$\widehat{\kappa}_r^{\mathrm{B}} = -\log\left(\frac{1}{S}\sum_{s=1}^{S} e^{-\kappa(\rho^{(s)})}\right), \qquad \widehat{C}_r^{\mathrm{B}} = \left(\frac{1}{S}\sum_{s=1}^{S} C(\rho^{(s)})^{-1}\right)^{-1}.$$

The Bayesian score is then obtained by replacing $\widehat{\kappa}_r, \widehat{C}_r$ with $\widehat{\kappa}_r^{\mathrm{B}}, \widehat{C}_r^{\mathrm{B}}$ in ϕ_r. Thus, plug-in and Bayesian variants differ only in the binwise scores, not in the optimization problem. We use the harmonic mean for the prefactor due to its higher stability.

5.3 Kernelized Relaxation in Representation Space

The finite-bin allocation objective still depends on a hard partition of the feature space. For neural networks, we instead work in representation space. Write the model as $f_\theta \circ h_\omega$, where h_ω is the representation map and f_θ the classification head. As is common in deep active learning, we use the current representation to define local neighborhoods and apply the acquisition criterion at the representation level (Ash et al., 2020).

Let $\mathcal{U}_i = (x_1^{\mathrm{u}}, \ldots, x_s^{\mathrm{u}})$ denote the candidate sequence at cycle i, and write $z_v = h_\omega(x_v^{\mathrm{u}})$ for the corresponding representation. For each anchor z_v, define

$$T_v = \sum_{u=1}^{s} K_\ell(z_v, z_u), \qquad l_v = \sum_{t=1}^{n_{i-1}} K_\ell(z_v, h_\omega(x_t)), \qquad n_{i-1}^{\mathrm{ker}} = \sum_{v=1}^{s} \frac{l_v}{T_v},$$

where $X_{i-1} = (x_1, \ldots, x_{n_{i-1}})$ is the sequence of previously selected instances. Here T_v is the available local candidate mass around z_v, l_v is the already selected local mass, and n_{i-1}^{ker} is the corresponding effective previous count in the kernelized representation.

We replace the hard-bin count by the local kernel count $\nu_v T_v$ and solve

$$\min_{\nu} \frac{1}{s} \sum_{v=1}^{s} e^{-\phi_v(\nu_v T_v)} \qquad \text{s.t.} \qquad l_v \le \nu_v T_v \le l_v + T_v, \quad \sum_{v=1}^{s} \nu_v = n_{i-1}^{\text{ker}} + b_i. \tag{11}$$

The function ϕ_v is obtained from the plug-in or Bayesian bin score at anchor z_v, using the predictive class probabilities of the current model. Thus, $\nu_v T_v$ plays the role of the final local count around z_v, while the equality constraint ensures that the total additional normalized mass equals the batch size b_i.

This relaxation recovers the finite-bin formulation when K_ℓ is the bin-indicator kernel,

$$K_\ell(z, z') = \begin{cases} 1, & z, z' \text{ lie in the same bin,} \\ 0, & \text{otherwise.} \end{cases}$$

In that case, all instances in the same bin share the same soft count, $n_{i-1}^{\text{ker}} = n_{i-1}$, and solving equation 11 is equivalent to solving the corresponding bin-allocation problem. For general kernels, equation 11 is a soft local-count analogue of the histogram objective. We give the formal change-of-variables argument in App. F.

5.4 Batch Construction and Kernel Scale

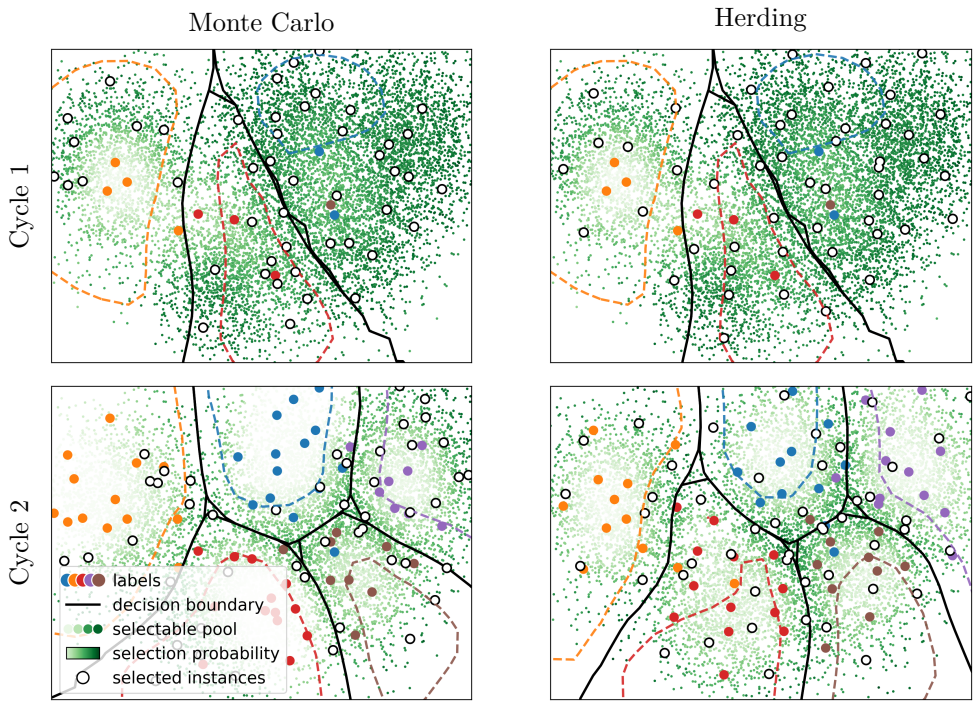


Figure 2: Selection behavior induced by the kernelized assignment distribution for a Parzen-window classifier (Chapelle, 2005) with batch size 50. Both variants shift from broad coverage toward boundary regions and outliers across cycles. Monte Carlo sampling is more clustered, whereas herding gives a more even deterministic selection.

For the kernelized local counts, we use an exponential cosine kernel (Bae et al., 2025) on normalized representations. Writing $z_v = h_\omega(x_v)$ and denoting cosine similarity by $k_{\cos}(z, z') = \frac{\langle z, z' \rangle}{\|z\|\|z'\|}$, we set

$$K_\ell(z, z') = \exp(\ell k_{\cos}(z, z') - \ell)\,.$$

The length scale ℓ is chosen by an effective-bin heuristic

$$\frac{1}{n_{\text{bins}}} = \frac{1}{s^2} \sum_{u,v=1}^{s} K_\ell(z_u, z_v), \tag{12}$$

so that the average kernel similarity matches that of an equal-size hard partition with n_{bins} bins. After solving equation 11, we convert the target local counts into an instance-selection distribution. The current normalized local mass around anchor z_v is l_v/T_v, while the optimized target is ν_v. We therefore define the residual assignment weight

$$\delta_v = \nu_v - \frac{l_v}{T_v}, \qquad q_v = \frac{\delta_v}{\sum_{u=1}^{s} \delta_v}.$$

In the hard-bin case, this assigns probability to each bin in proportion to its target additional count.

The final batch over the candidates with non zero q can be drawn by Monte Carlo sampling or by kernel herding (Chen et al., 2010). Monte Carlo sampling follows the target distribution only in expectation and may create holes or clusters. Kernel herding instead acts as a deterministic rounding of q by matching its kernel mean, and can therefore be viewed as a soft analogue of filling bins according to their target counts. Figure 2 visualizes this difference.

6 Experiments

We evaluate the practical query strategy derived in Section 5. Our goal is threefold. First, we test whether the method is competitive with strong active learning baselines. Second, we examine whether its behavior reflects the budget-dependent exploration–exploitation shift suggested by the histogram analysis. Third, we evaluate robustness in challenging settings: duplicate instances, label noise, and few-cycle large-batch acquisition. Unless stated otherwise, we use the hard-risk formulation, $n_{\text{bins}} = 10$ for the effective number of bins and a Jeffreys prior for the Dirichlet scores. An ablation in App. G shows the importance of the hard risk formulation, a low value for n_{bins} and stability of the method with respect to the chosen kernel. Run-time comparisons in App. H shows speedups over BADGE on datasets with many classes.

Experimental setup. Tables 1 and 2 summarizes the experimental setup. We evaluate linear probing with two frozen backbones and compare against four query strategies on six image-classification datasets from different domains, with the number of classes ranging from 4 to 200. Each run starts from a random initial labeled set and proceeds for 100 active learning cycles. Batch sizes and total labeling budgets are chosen per dataset. At each cycle, every query strategy receives a random candidate subset of at most 10,000 unlabeled instances. Each configuration is repeated with the same 20 random seeds.

Table 1: Compared compounds.

Factor	Method
Backbone	DINOv3 (2025)
	CLIP (2021)
Query strategy	TypiClust (2022)
	BADGE (2020)
	Margin (2008)
	Random

Table 2: Dataset statistics.

Dataset	Classes	Pool	Batch	Budget
OCTMNIST (2021)	4	97,477	50	5,000
CIFAR-10 (2009)	10	50,000	50	5,000
SVHN (2011)	10	73,257	50	5,000
DopAnim (2024)	15	10,484	30	3,000
CIFAR-100 (2009)	100	50,000	50	5,000
Tiny ImageNet (2015)	200	100,000	100	10,000

6.1 Competitiveness

As discussed in Section 3.1, active learning aims to achieve low risk under a fixed labeling budget. To aggregate performance across budgets, we use the area under the learning curve (AULC), computed from accuracy across active learning cycles (Guyon et al., 2011; Reyes et al., 2018). This corresponds to one minus the average zero-one loss across active learning cycles. Figure 3 reports pairwise wins under this metric, aggregated across datasets, backbones, and seeds.

We also report the relative reduction in zero-one loss compared with random sampling,

$$1 - \frac{\text{Avg. zero-one loss of query strategy}}{\text{Avg. zero-one loss of random sampling}}. \tag{13}$$

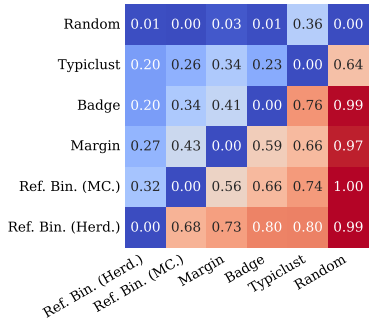


Figure 3: Entry (i, j) shows the average number of wins of strategy i over strategy j with respect to the AULC.

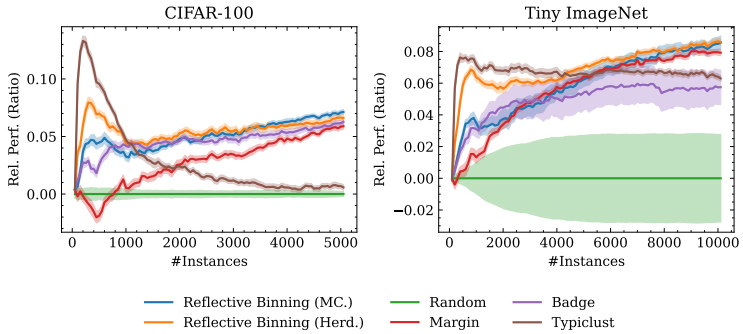


Figure 4: Relative reduction in zero-one loss over random sampling on CIFAR-100 and Tiny ImageNet using CLIP as backbone. Curves show means over seeds and shaded regions indicate standard errors.

This normalization accounts for differences in dataset difficulty and makes improvements across stages more comparable. Figure 4 shows representative learning curves for CIFAR-100 and Tiny ImageNet using CLIP as backbone. Overall, both Monte Carlo (MC) and herding (Herd.) variants of our method are competitive with the selected baselines. The herding variant is usually more stable, consistent with its role as a deterministic rounding of the assignment distribution. The full collection of learning curves is given in App. I.

6.2 Exploration–Exploitation Tradeoff

Active learning balances exploration of underrepresented regions against exploitation of model uncertainty. TypiClust and Margin represent opposite ends of this tradeoff. Section 4 suggests that the balance should shift with budget: early allocations favor coverage, whereas late allocations concentrate on difficult regions, see also Hacohen et al. (2022); Huseljic et al. (2026).

For our method, this behavior can be visualized through the assignment distribution returned by equation 11. Let q_v denote the normalized assignment probability of candidate instance z_v in representation space. We use the entropy

$$H(q) = -\sum_{v=1}^{s} q_v \log q_v$$

as a proxy for exploration. High entropy indicates that mass is spread across many candidates, whereas low entropy indicates concentration on a smaller set of candidates. Equivalently, $\exp(H(q))$ is the effective support size of the assignment distribution.

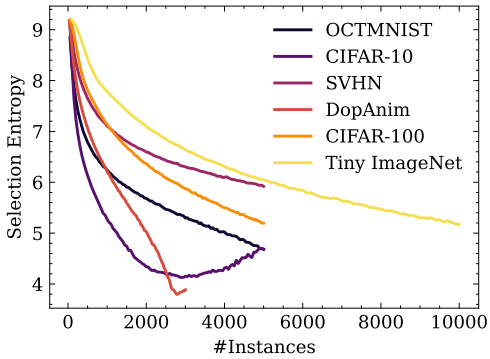


Figure 5: Selection entropy across active learning cycles for the Monte Carlo variant. The entropy is computed from the assignment distribution returned by equation 11.

Under the constraint that each instance can be selected at most once, the entropy lies between $\log(b_i)$ and $\log(s)$, where b_i is the batch size and s the candidate-pool size. Figure 5 shows this quantity for the Monte Carlo variant. The herding variant yields similar trends. For easier datasets such as CIFAR-10, the entropy decreases rapidly, indicating an early shift toward exploitation. For harder datasets such as Tiny ImageNet, it decreases more gradually, suggesting that the method remains exploratory for longer. On CIFAR-10, the entropy eventually plateaus and increases again. This may reflect saturation of the most difficult pool regions. Once few boundary-near candidates remain, the pool constraints force the strategy to shift mass to easier regions, spreading the assignment over more candidates. For a similar conclusion from a different perspective see Huseljic et al. (2026).

6.3 Robustness

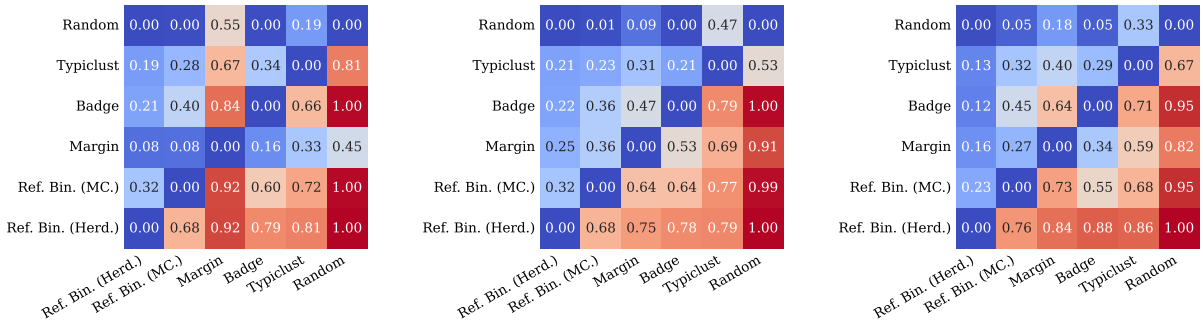
We evaluate robustness in three practically relevant scenarios.

Duplicates. Following the motivation of robustness to repeated samples in active learning (Kirsch et al., 2023), we inject duplicates into the unlabeled pool. Concretely, we duplicate each instance twice and then randomly subsample the enlarged pool back to its original size.

Label noise. To test robustness to noisy labels, we first train a probing head on the full dataset and apply temperature scaling so that the average confidence is 80%. We then relabel 50% of the training instances by sampling from the resulting temperature-scaled predictive distribution. This corrupts roughly 10% of labels in expectation. This corresponds to xy-dependent noise (Algan & Ulusoy, 2021). For a similar simulation technique see Algan & Ulusoy (2020) or Gu et al. (2023).

Few-cycle / large-batch acquisition. In annotation campaigns long turnaround times allow only a small number of acquisition rounds. To study this regime, we repeat the linear-probing experiments with fewer active learning cycles and proportionally larger batches. Specifically, we increase the candidate subset size to 30,000, multiply the batch size by 10, and reduce the number of cycles by the same factor.

Figure 6 summarizes the pairwise comparisons. Some baselines, especially margin, degrade substantially under these perturbations. Both variants of our method remain competitive, with the herding variant showing its largest advantage in the few-cycle / large-batch regime. The full learning curves are given in App. I.



(a) Duplicates. (b) Label noise. (c) Few-cycle / large-batch regime.

Figure 6: Competitiveness of the compared query strategies under challenging active learning scenarios. Each panel shows pairwise win rates with respect to the AULC under a different type of perturbation.

7 Conclusion

Reflection treats active learning as a design problem: candidate queries are evaluated through hypothetical relabelings of both previously selected and newly selected instances. For cross-entropy, a MAP approximation recovers the full-design FIR objective used by BAIT, explaining why previously selected and candidate instances should enter the same design criterion. For zero-one loss, our histogram analysis gives explicit expected-risk asymptotics and shows how allocation shifts with budget: early behavior is more density- and coverage-sensitive, while late-stage allocation is governed by local decision difficulty. This also gives an aleatoric interpretation of margin-like top-two separations and motivates a kernelized local-counting strategy in representation space. Empirically, the resulting method is competitive with strong baselines and robust to duplicates, label noise, and few-cycle large-batch acquisition.

For future work, we want to extend the theory to richer model classes and study reflection beyond frozen representations.

References

Görkem Algan and Ilkay Ulusoy. Label noise types and their effects on deep learning, 2020.

Görkem Algan and Ilkay Ulusoy. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, 215:106771, 2021. doi: 10.1016/j.knosys.2021.106771.

András Antos, Varun Grover, and Csaba Szepesvári. Active learning in multi-armed bandits. In *Algorithmic Learning Theory*, volume 5254 of *Lecture Notes in Computer Science*, pp. 287–302, 2008.

Jordan Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham M. Kakade. Gone fishing: Neural active learning with Fisher embeddings. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8927–8939, 2021.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

Wonho Bae, Junhyug Noh, and Danica J. Sutherland. Generalized coverage for more robust low-budget active learning. In *Computer Vision – ECCV 2024*, volume 15141 of *Lecture Notes in Computer Science*, pp. 318–334, 2025.

Raghu Raj Bahadur and R. Ranga Rao. On deviations of the sample mean. *The Annals of Mathematical Statistics*, 31(4):1015–1027, 1960. doi: 10.1214/aoms/1177705674.

Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60 (6):503–515, 1954. doi: 10.1090/S0002-9904-1954-09848-8.

Alexandra Carpentier, Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos, and Peter Auer. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *Algorithmic Learning Theory*, volume 6925 of *Lecture Notes in Computer Science*, pp. 189–203, 2011.

Olivier Chapelle. Active learning for parzen window classifier. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pp. 49–56, 2005.

Kamalika Chaudhuri, Sham M. Kakade, Praneeth Netrapalli, and Sujay Sanghavi. Convergence rates of active learning for maximum likelihood estimation. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1090–1098, 2015.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 109–116, 2010.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. doi: 10.1007/BF00993277.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. doi: 10.1613/jair.295.

Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *Proceedings of the AAAI Conference on Artificial Intelligencee*, pp. 746–751, 2005.

Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*. Springer, 2 edition, 2009.

Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In *Computer Vision – ECCV 2014*, volume 8692 of *Lecture Notes in Computer Science*, pp. 562–577, 2014.

Keren Gu, Xander Masotto, Vandana Bachani, Balaji Lakshminarayanan, Jack Nikodem, and Dong Yin. An instance-dependent simulation framework for learning with label noise. *Machine Learning*, 112(6): 1871–1896, 2023. doi: 10.1007/s10994-022-06207-7.

Isabelle Guyon, Gavin C. Cawley, Gideon Dror, and Vincent Lemaire. Results of the active learning challenge. In *Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pp. 19–45, 2011.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8175–8195, 2022.

Marek Herde, Denis Huseljic, Bernhard Sick, and Adrian Calma. A survey on cost types, interaction schemes, and annotator performance models in selection algorithms for active learning in classification. *IEEE Access*, 9:166970–166989, 2021. doi: 10.1109/ACCESS.2021.3135514.

Marek Herde, Denis Huseljic, Lukas Rauch, and Bernhard Sick. dopanim: A dataset of doppelganger animals with noisy annotations from multiple humans. In *Advances in Neural Information Processing Systems*, volume 37, pp. 51085–51117, 2024.

Trong Nghia Hoang, Bryan Kian Hsiang Low, Patrick Jaillet, and Mohan Kankanhalli. Nonmyopic ε-Bayes-optimal active learning of gaussian processes. In *Proceedings of the 31st International Conference on Machine Learning*, volume 31 of *Proceedings of Machine Learning Research*, pp. 739–747, 2014.

Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the Fifteenth International Conference on World Wide Web*, pp. 633–642, 2006.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011.

Xun Huan, Jayanth Jagalur, and Youssef Marzouk. Optimal experimental design: Formulations and computations. *Acta Numerica*, 33:715–840, 2024. doi: 10.1017/S0962492924000023.

Denis Huseljic, Paul Hahn, Marek Herde, Christoph Sandrock, and Bernhard Sick. BoSS: A best-of-strategies selector as an oracle for deep active learning. *Transactions on Machine Learning Research*, 2026.

Andreas Kirsch and Yarin Gal. Unifying approaches in active learning and active sampling via Fisher information and information-theoretic quantities. *Transactions on Machine Learning Research*, 2022.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7026–7037, 2019.

Andreas Kirsch, Sebastian Farquhar, Parmida Atighehchian, Andrew Jesson, Frédéric Branchaud-Charron, and Yarin Gal. Stochastic batch acquisition: A simple baseline for deep active learning. *Transactions on Machine Learning Research*, 2023.

Daniel Kottke. *A Holistic, Decision-Theoretic Framework for Pool-Based Active Learning*. Kassel University Press, 2021.

Daniel Kottke, Marek Herde, Christoph Sandrock, Denis Huseljic, Georg Krempl, and Bernhard Sick. Toward optimal probabilistic active learning using a Bayesian approach. *Machine Learning*, 110(6):1199–1231, 2021. doi: 10.1007/s10994-021-05986-9.

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

Ya Le and Xuan S. Yang. Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Kian Hsiang Low, John Dolan, and Pradeep Khosla. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 19, pp. 233–240, 2009.

Gábor Lugosi and Andrew Nobel. Consistency of data-driven histogram methods for density estimation and classification. *The Annals of Statistics*, 24(2):687–706, 1996. doi: 10.1214/aos/1032894460.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

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.

Oscar Reyes, Abdulrahman H. Altalhi, and Sebastián Ventura. Statistical comparisons of active learning strategies over multiple datasets. *Knowledge-Based Systems*, 145:274–288, 2018. doi: 10.1016/j.knosys.2018.01.033.

Christian P. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, 2 edition, 2007.

Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pp. 413–424, 2006.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 441–448, 2001.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden Markov models for information extraction. In *Advances in Intelligent Data Analysis*, volume 2189 of *Lecture Notes in Computer Science*, pp. 309–318, 2001.

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

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079, 2008.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, volume 20, pp. 1289–1296, 2007.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 287–294, 1992.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025.

Jiancheng Yang, Rui Shi, and Bingbing Ni. MedMNIST classification decathlon: A lightweight AutoML benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195, 2021.

Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22354–22367, 2022.

Tong Zhang and Frank J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 1191–1198, 2000.

A Optimality in Active Learning

There are several natural notions of optimality in active learning. From a Bayesian decision-theoretic perspective, the natural optimum is a non-myopic Bayes policy: a query strategy that minimizes posterior expected final risk over the full sequence of acquisition decisions. Bayes rules are a central class of admissible decision rules under standard conditions (Robert, 2007), but in active learning the corresponding non-myopic optimization is generally computationally intractable.

A different notion of optimality is obtained by giving the learner access to the true data-generating parameter $\theta^\star$. Two oracle baselines are common. A *myopic oracle* evaluates each acquisition using the true conditional distribution (Huseljic et al., 2026). An *a priori oracle design* chooses all instances in advance so as to minimize the expected risk after training. In the notation of Section 3.1, the latter is

$$\min_{X \in \mathcal{X}^n} \mathbb{E}_{Y \sim p_{\theta^\star}(\cdot|X)} \left[R_{\theta^\star}(\mathrm{train}(X, Y)) \right]. \tag{14}$$

Here $X = (x_1, \ldots, x_n)$ is a selected design and $Y = (y_1, \ldots, y_n)$ denotes fresh labels drawn independently according to $y_t \sim p_{\theta^\star}(\cdot \mid x_t)$.

To compare sequential query strategies with the a priori design objective in equation 14, it is useful to evaluate the selected design rather than the particular realized labeled dataset. An active learning strategy is run in the usual way: it selects instances sequentially, observes labels, and uses those labels to choose later batches. After the final sequence of selected instances X_k has been obtained, however, we evaluate the design by drawing fresh labels $Y \sim p_{\theta^\star}(\cdot \mid X_k)$ and training on (X_k, Y). This relabeling protocol should be understood as a design-based evaluation device: it removes the accidental effect of the particular labels observed during acquisition, while preserving their role in determining which instances were selected.

Under this design-based evaluation, every query strategy induces a possibly random final design X_k. Its expected risk cannot be smaller than the value of the a priori oracle design problem in equation 14, which is allowed to choose the entire design in advance with knowledge of $\theta^\star$. Thus, the oracle objective provides a lower bound on the achievable risk under this evaluation protocol.

This perspective is closely aligned with reflection. Reflection also evaluates selected instances as a design under hypothetical relabelings. The difference is that the unknown ground-truth parameter $\theta^\star$ is replaced by the current posterior, or by an approximation such as a point estimate. In this sense, reflection can be viewed as a posterior version of the a priori design objective.

For maximum-likelihood training, this design-based comparison is especially useful because consistency of the estimator allows the oracle design criterion to be approached from data. This is the route taken by two-stage analyses such as Chaudhuri et al. (2015). An initial labeled sample is used to estimate the data-generating parameter, and the remaining budget is allocated according to an approximately optimal static design.

B Derivation of the Histogram Zero-One Asymptotics

We derive the binwise asymptotic expansion used in Theorem 2. It suffices to consider a single bin with class distribution $\rho = (\rho_1, \ldots, \rho_c) \in \Delta_c$, where classes are relabeled so that

$$\rho_1 > \rho_2 > \cdots > \rho_c > 0.$$

This ordering is only a notational convention. Let $Y_1, \ldots, Y_n$ be i.i.d. labels drawn from ρ, and let

$$\hat{\rho}_j = \frac{1}{n} \sum_{\ell=1}^{n} \mathbf{1}\{Y_\ell = j\}$$

be the empirical class proportions. The histogram classifier predicts the empirical majority label

$$\hat{y} = \operatorname*{arg\,max}_{j \in \{1, \ldots, c\}} \hat{\rho}_j.$$

We count ties against the Bayes-optimal class as errors.

The proof has three steps. First, we use Sanov's theorem to show that the dominant incorrect decision is the top-two crossing event $\hat{\rho}_2 \geq \hat{\rho}_1$. Second, we apply a Bahadur–Rao refinement to obtain the prefactor of this crossing probability. Third, we translate the binwise crossing asymptotic into the soft and hard histogram risks.

B.1 Reduction to the Top-Two Crossing Event

We first recall Sanov's theorem in the form used below. We write $D_{\mathrm{KL}}(q \parallel \rho) = \sum_{j=1}^{c} q_j \log \frac{q_j}{\rho_j}$, for the KL divergence from a probability distribution q to ρ.

Theorem 3 (Sanov's theorem; Dembo & Zeitouni, 2009)**.** *Let $\hat{\rho}_n$ be the empirical distribution of n i.i.d. samples from $\rho \in \Delta_c$. If $A \subseteq \Delta_c$ satisfies $A \subseteq \overline{\mathrm{int}(A)}$, then*

$$\lim_{n\to\infty} \frac{1}{n} \log P(\hat{\rho}_n \in A) = -\inf_{q \in A} D_{\mathrm{KL}}(q \parallel \rho).$$

The decision regions and crossing events used below satisfy this condition.

For $j \in \{1, \ldots, c\}$, define the closed decision region

$$A_j = \{q \in \Delta_c : q_j \geq q_\ell \text{ for all } \ell\}.$$

Then $P(\hat{y} = j) = P(\hat{\rho} \in A_j)$ up to the chosen tie-breaking convention. Sanov's theorem shows that these probabilities decay exponentially with rates

$$d_j := \inf_{q \in A_j} D_{\mathrm{KL}}(q \parallel \rho).$$

Since $\rho \in A_1$, we have $d_1 = 0$. For $j \geq 2$, the event A_j is a large-deviation event and $d_j > 0$.

The following lemma shows that among all incorrect labels, class 2 has the smallest large-deviation rate.

Lemma 4. *For every $j \geq 3$,*

$$d_j > d_2.$$

Proof. Fix $j \geq 3$, and let $q \in A_j$ be a minimizer of $D_{\mathrm{KL}}(q \parallel \rho)$ over A_j. Such a minimizer exists by compactness. Construct $z \in \Delta_c$ by swapping the second and jth coordinates of q:

$$z_i = \begin{cases} q_j, & i = 2, \\ q_2, & i = j, \\ q_i, & \text{otherwise.} \end{cases}$$

Since $q \in A_j$, we have $q_j \geq q_i$ for all i, and hence $z_2 \geq z_i$ for all i. Thus $z \in A_2$. Moreover,

$$\begin{aligned} D_{\mathrm{KL}}(q \parallel \rho) - D_{\mathrm{KL}}(z \parallel \rho) &= q_2 \log\left(\frac{q_2}{\rho_2}\right) + q_j \log\left(\frac{q_j}{\rho_j}\right) - q_j \log\left(\frac{q_j}{\rho_2}\right) - q_2 \log\left(\frac{q_2}{\rho_j}\right) \\ &= (q_j - q_2) \log\left(\frac{\rho_2}{\rho_j}\right). \end{aligned}$$

Because $\rho_2 > \rho_j$, the logarithm is positive. If $q_j > q_2$, then $D_{\mathrm{KL}}(q \parallel \rho) > D_{\mathrm{KL}}(z \parallel \rho) \geq d_2$, and hence $d_j > d_2$.

It remains to consider the case $q_j = q_2$, i.e., $z = q$. For sufficiently small $\varepsilon > 0$, define

$$z(\varepsilon)_i = \begin{cases} q_2 + \varepsilon, & i = 2, \\ q_j - \varepsilon, & i = j, \\ q_i, & \text{otherwise.} \end{cases}$$

Then $z(\varepsilon) \in A_2$. The derivative of the divergence at $\varepsilon = 0$ is

$$\frac{d}{d\varepsilon} D_{\mathrm{KL}}(z(\varepsilon) \parallel \rho)\bigg|_{\varepsilon=0} = \log\left(\frac{q_2}{\rho_2}\right) - \log\left(\frac{q_j}{\rho_j}\right) = \log\left(\frac{\rho_j}{\rho_2}\right) < 0.$$

Thus, for sufficiently small $\varepsilon > 0$, $D_{\mathrm{KL}}(q \parallel \rho) > D_{\mathrm{KL}}(z(\varepsilon) \parallel \rho) \geq d_2$, and again $d_j > d_2$. □

By Sanov's theorem and Lemma 4, the events $\{\hat{y} = j\}$ for $j \geq 3$ are exponentially smaller than the event $\{\hat{y} = 2\}$. Hence

$$P(\hat{y} \neq 1) \sim P(\hat{y} = 2).$$

It remains to identify $P(\hat{y} = 2)$ with the simpler top-two crossing event $P(\hat{\rho}_2 \geq \hat{\rho}_1)$. Decompose

$$\begin{aligned} P(\hat{\rho}_2 \geq \hat{\rho}_1) = {} & P(\hat{\rho}_2 \geq \hat{\rho}_1,\ \hat{\rho}_2 \geq \hat{\rho}_\ell \text{ for all } \ell \geq 3) \\ & + P(\hat{\rho}_2 \geq \hat{\rho}_1,\ \exists \ell \geq 3 : \hat{\rho}_\ell \geq \hat{\rho}_2). \end{aligned}$$

The first term is $P(\hat{y} = 2)$ under the pessimistic tie convention. The second term is exponentially smaller. Indeed, let

$$B = \{q \in \Delta_c : q_2 \geq q_1 \text{ and } q_\ell \geq q_2 \text{ for some } \ell \geq 3\}.$$

The minimizer of $D_{\mathrm{KL}}(q \parallel \rho)$ over the set $\{q \in \Delta_c : q_2 \geq q_1\}$ is obtained at the top-two tie and is given by

$$q_1 = q_2 = s\sqrt{\rho_1 \rho_2}, \qquad q_\ell = s\rho_\ell \quad (\ell \geq 3),$$

where

$$s = \frac{1}{1 - \rho_1 - \rho_2 + 2\sqrt{\rho_1 \rho_2}}.$$

The minimizer can be verified by the KKT conditions for minimizing $D_{\mathrm{KL}}(q \parallel \rho)$ over the convex set $\{q \in \Delta_c : q_2 \geq q_1\}$. The displayed point satisfies the simplex constraint, nonnegativity, and the active constraint $q_1 = q_2$. This point satisfies $q_\ell < q_2$ for all $\ell \geq 3$, and therefore does not lie in B. Consequently,

$$\inf_{q \in B} D_{\mathrm{KL}}(q \parallel \rho) > d_2,$$

so Sanov's theorem gives

$$P(\hat{\rho} \in B) = o(P(\hat{y} = 2)).$$

We conclude that

$$P(\hat{y} \neq 1) \sim P(\hat{y} = 2) \sim P(\hat{\rho}_2 \geq \hat{\rho}_1).$$

B.2 Bahadur–Rao Refinement

We now compute the sharp asymptotic behavior of the top-two crossing probability. Let

$$Z_\ell = \begin{cases} 1, & Y_\ell = 2, \\ -1, & Y_\ell = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$\hat{\rho}_2 \geq \hat{\rho}_1 \quad \Longleftrightarrow \quad \frac{1}{n} \sum_{\ell=1}^{n} Z_\ell \geq 0.$$

The cumulant generating function of Z_ℓ is

$$K(t) = \log\left(\rho_1 e^{-t} + \rho_2 e^{t} + 1 - \rho_1 - \rho_2\right).$$

The saddle point $t^\star > 0$ satisfying $K'(t^\star) = 0$ is

$$t^\star = \log \sqrt{\frac{\rho_1}{\rho_2}}.$$

At this point,

$$K(t^\star) = \log(1 - \rho_1 - \rho_2 + 2\sqrt{\rho_1\rho_2}),$$

and

$$K''(t^\star) = \frac{2\sqrt{\rho_1\rho_2}}{1 - \rho_1 - \rho_2 + 2\sqrt{\rho_1\rho_2}}.$$

We use the Bahadur–Rao theorem for lattice random variables (Bahadur & Rao, 1960; Dembo & Zeitouni, 2009). For the event $\frac{1}{n}\sum_{\ell=1}^{n} Z_\ell \geq 0$, it gives

$$P(\hat{\rho}_2 \geq \hat{\rho}_1) \sim \frac{1}{\sqrt{2\pi n K''(t^\star)}} \frac{1}{1 - e^{-t^\star}} e^{-n\kappa},$$

where

$$\kappa = -K(t^\star) = -\log\Big(1 - (\sqrt{\rho_1} - \sqrt{\rho_2})^2\Big).$$

Substituting $t^\star$ and $K''(t^\star)$ yields

$$P(\hat{\rho}_2 \geq \hat{\rho}_1) \sim \frac{C}{\sqrt{n}} e^{-\kappa n},$$

with

$$C = \Big(1 - \sqrt{\rho_2/\rho_1}\Big) \sqrt{\frac{4\pi\sqrt{\rho_1\rho_2}}{1 - \big(\sqrt{\rho_1} - \sqrt{\rho_2}\big)^2}}.$$

B.3 From Binwise Crossings to Histogram Risks

Applying the single-bin result to bin r, with $\rho = \rho_r$ and $n = \nu_r$, gives

$$P(\hat{y}_r \neq 1) \sim \frac{C_r}{\sqrt{\nu_r}} e^{-\kappa_r \nu_r}.$$

This immediately gives the hard-risk expansion:

$$\mathbb{E}[R_{\text{hard}}(\hat{\rho})] = \sum_{r=1}^{m} w_r P(\hat{y}_r \neq 1) \sim \sum_{r=1}^{m} w_r \frac{C_r}{\sqrt{\nu_r}} e^{-\kappa_r \nu_r}.$$

For the soft risk, recall that

$$\mathbb{E}[R_{\text{soft}}(\hat{\rho})] = \sum_{r=1}^{m} w_r \sum_{j=1}^{c} (1 - \rho_{r,j}) P(\hat{y}_r = j).$$

The Bayes risk in bin r is $1 - \rho_{r,1}$. Therefore the excess soft risk is

$$\mathbb{E}[R_{\text{soft}}(\hat{\rho})] - R^\star_{\text{soft}} = \sum_{r=1}^{m} w_r \sum_{j=2}^{c} (\rho_{r,1} - \rho_{r,j}) P(\hat{y}_r = j).$$

By the top-two dominance result, only $j = 2$ contributes at leading order. Thus

$$\mathbb{E}[R_{\text{soft}}(\hat{\rho})] - R^\star_{\text{soft}} \sim \sum_{r=1}^{m} w_r (\rho_{r,1} - \rho_{r,2}) \frac{C_r}{\sqrt{\nu_r}} e^{-\kappa_r \nu_r}.$$

This proves Theorem 2.

C Allocations Induced by the Histogram Asymptotics

Theorem 2 leads to binwise terms of the form

$$A_r \nu_r^{-1/2} e^{-\kappa_r \nu_r},$$

where ν_r is the number of labels allocated to bin r. For the hard decision risk, $A_r = w_r C_r$, while for the soft risk,

$$A_r = w_r(\rho_{r,1} - \rho_{r,2})C_r.$$

We assume throughout this section that $A_r > 0$ and $\kappa_r > 0$ for all bins.

C.1 Late-Stage Allocation

In the late-stage regime, the exponential rates in Theorem 2 dominate the polynomial prefactors and bin weights. The following proposition makes this precise for the exact expected histogram risk: optimal integer allocations asymptotically balance the binwise exponents.

Proposition 5 (Late-stage allocation)**.** *Assume that $m < \infty$, $w_r > 0$, and $\rho_{r,1} > \rho_{r,2} > \cdots > \rho_{r,c} > 0$ for all bins r. Let $\nu^{(n)} = (\nu_1^{(n)}, \ldots, \nu_m^{(n)})$ be a minimizer of the exact soft excess risk*

$$S_n(\nu) = \mathbb{E}[R_{\text{soft}}(\hat{\rho}_\nu)] - R^\star_{\text{soft}}$$

over positive integer allocations satisfying $\sum_{r=1}^m \nu_r = n$. Then

$$\frac{\nu_r^{(n)}}{n} \to \frac{1/\kappa_r}{\sum_{s=1}^m 1/\kappa_s}.$$

The corresponding exponential decay rate is

$$\left(\sum_{s=1}^m \frac{1}{\kappa_s}\right)^{-1}.$$

The same conclusion holds for the exact hard-risk objective $\mathbb{E}[R_{\text{hard}}(\hat{\rho}_\nu)]$.

Proof. We give the proof for the soft excess risk. The hard-risk case is identical. Write the objective as

$$S_n(\nu) = \sum_{r=1}^m g_r(\nu_r),$$

where

$$g_r(t) = w_r \sum_{j=2}^c (\rho_{r,1} - \rho_{r,j}) P_t(\hat{y}_r = j)$$

is the exact excess-risk contribution of bin r when t labels are allocated to that bin. By Theorem 2,

$$g_r(t) \sim A_r t^{-1/2} e^{-\kappa_r t}, \qquad A_r = w_r(\rho_{r,1} - \rho_{r,2})C_r > 0.$$

Therefore, for any sequence t_n with $t_n/n \to x_r$,

$$-\frac{1}{n} \log g_r(t_n) \to \kappa_r x_r,$$

with limiting rate 0 if $x_r = 0$.

Now let $\nu^{(n)}$ be a sequence of optimal allocations and consider any convergent subsequence of the normalized allocations,

$$x_r^{(n)} = \frac{\nu_r^{(n)}}{n} \to x_r.$$

Since m is finite, the logarithm of the finite sum is governed by its largest summand on the exponential scale. Hence along this subsequence,

$$-\frac{1}{n}\log S_n(\nu^{(n)}) \to \min_{1\le r\le m} \kappa_r x_r.$$

Thus any subsequential limit of optimal allocations must solve the limiting max-min problem

$$\max_{x\in\Delta_m} \min_{1\le r\le m} \kappa_r x_r, \qquad \Delta_m = \left\{ x \in [0,1]^m : \sum_{r=1}^m x_r = 1 \right\}.$$

The unique maximizer equalizes all exponents:

$$\kappa_r x_r = \tau \quad \text{for all } r.$$

Using $\sum_r x_r = 1$ gives

$$\tau = \left(\sum_{s=1}^m \frac{1}{\kappa_s} \right)^{-1}, \qquad x_r = \frac{1/\kappa_r}{\sum_{s=1}^m 1/\kappa_s}.$$

Since every convergent subsequence of the normalized optimal allocations has the same limit and by the compactness of Δ_m we get that the full sequence converges to this allocation. The common value τ is the exponential decay rate. □

Thus, to leading exponential order, the late-stage allocation is governed by decision difficulty through κ_r. The density weights w_r and prefactors C_r affect only lower-order corrections.

C.2 Low-Budget Approximation

The preceding result describes the large-budget exponential regime. For small budgets, the exponential factors may vary only weakly across bins. A complementary approximation is therefore obtained by dropping the exponential factors and solving the continuous relaxation

$$\min_{\nu_1,\dots,\nu_m>0} \sum_{r=1}^m A_r \nu_r^{-1/2} \quad \text{s.t.} \quad \sum_{r=1}^m \nu_r = n.$$

Considering the continuous relaxation the Lagrange equations are

$$-\frac{1}{2} A_r \nu_r^{-3/2} + \lambda = 0,$$

and therefore

$$\nu_r = n \frac{A_r^{2/3}}{\sum_{s=1}^m A_s^{2/3}}.$$

For the hard decision risk this gives

$$\nu_r \propto (w_r C_r)^{2/3},$$

whereas for the soft risk it gives

$$\nu_r \propto \big(w_r(\rho_{r,1} - \rho_{r,2}) C_r \big)^{2/3}.$$

Thus, in this approximation, density enters sublinearly through the exponent 2/3: high-weight bins receive more labels, but the allocation remains more diverse than density-proportional allocation. Unlike the late-stage allocation in Prop. 5, this low-budget rule is heuristic. It applies the large-count asymptotic form in a regime where finite-count and integer effects are most pronounced, and then solves a continuous relaxation after neglecting the variation of the exponential factors. Its purpose is therefore interpretive: it describes the exploratory tendency of the objective before the exponential rates dominate.

C.3 Cross-Entropy Comparison

For comparison, consider the same histogram model under cross-entropy loss. Under the usual interior asymptotics, the binwise excess cross-entropy risk satisfies

$$\mathbb{E}[H(\rho_r, \hat{\rho}_r)] - H(\rho_r) \sim \frac{c-1}{2\nu_r}.$$

The leading allocation problem is therefore

$$\min_{\nu_1,\dots,\nu_m>0} \sum_{r=1}^{m} w_r \frac{c-1}{2\nu_r} \quad \text{s.t.} \quad \sum_{r=1}^{m} \nu_r = n.$$

Solving the Lagrange equations gives

$$\nu_r = n \frac{\sqrt{w_r}}{\sum_{s=1}^{m} \sqrt{w_s}}.$$

Hence, in the histogram model, the leading cross-entropy allocation depends only on the bin weights and not on the bin-wise class probabilities. Under the continuous relaxation of the asymptotic approximation, this relative allocation is independent of the total budget, so the leading cross-entropy objective does not exhibit the low-budget to late-stage shift seen for zero-one loss. As in Prop. 5, the limiting relative allocation can be transferred from the asymptotic approximation to the exact cross-entropy objective in the large-budget limit.

D Posterior-Averaged Histogram Asymptotics

The fixed-distribution analysis in App. B treats the bin distribution ρ as known. This corresponds to the oracle design problem, or to a plug-in version of reflection in which the posterior is replaced by a point estimate. In the fully Bayesian reflective objective, however, the bin distribution is itself random under the current posterior.

There are two sample sizes involved. The posterior is based on the labels already observed in a bin, while the classifier in the reflective replay is trained on hypothetical labels generated for the selected design. Thus, conditional on a posterior draw ρ, the empirical distribution $\hat{\rho}_\nu$ is formed from ν independent labels drawn from ρ. The following theorem shows that posterior averaging preserves the exponential-in-count structure of the plug-in analysis, but changes the rate.

Theorem 6 (Posterior-averaged Sanov theorem)**.** *Let $\bar{\rho} \in \Delta_c$ have strictly positive entries and let $\sigma > 0$. For each ν, let*

$$\rho \sim \mathrm{Dirichlet}(\sigma\nu\,\bar{\rho}),$$

and, conditional on ρ, let $\hat{\rho}_\nu$ be the empirical distribution of ν i.i.d. labels drawn from ρ. If $A \subseteq \Delta_c$ satisfies $A \subseteq \overline{\mathrm{int}(A)}$, then

$$\lim_{\nu\to\infty} \frac{1}{\nu} \log P(\hat{\rho}_\nu \in A) = -\inf_{q\in A} J_\sigma(q;\bar{\rho}),$$

where

$$J_\sigma(q;\bar{\rho}) = D_{\mathrm{KL}}(q \parallel m_\sigma(q)) + \sigma D_{\mathrm{KL}}(\bar{\rho} \parallel m_\sigma(q)), \qquad m_\sigma(q) = \frac{q+\sigma\bar{\rho}}{1+\sigma}.$$

Proof. Let $\mathcal{T}_\nu$ denote the set of empirical distributions, or types, with denominator ν. For a type $q \in \mathcal{T}_\nu$, conditional on ρ, the multinomial probability of observing type q is

$$P(\hat{\rho}_\nu = q \mid \rho) = \binom{\nu}{\nu q_1, \dots, \nu q_c} \prod_{j=1}^{c} \rho_j^{\nu q_j}.$$

By Stirling's formula,

$$\frac{1}{\nu} \log P(\hat{\rho}_\nu = q \mid \rho) = -D_{\mathrm{KL}}(q \parallel \rho) + o(1),$$

uniformly up to subexponential factors.

The Dirichlet density of ρ is

$$f_\nu(\rho) = \frac{\Gamma(\sigma\nu)}{\prod_{j=1}^{c}\Gamma(\sigma\nu\bar{\rho}_j)} \prod_{j=1}^{c} \rho_j^{\sigma\nu\bar{\rho}_j - 1}.$$

Again by Stirling's formula, its logarithm satisfies, on the exponential scale,

$$\frac{1}{\nu}\log f_\nu(\rho) = -\sigma D_{\mathrm{KL}}(\bar{\rho} \parallel \rho) + o(1),$$

up to terms that are subexponential in ν.

Hence the posterior-averaged probability of observing a fixed type q has exponential rate

$$P(\hat{\rho}_\nu = q) = \int_{\Delta_c} P(\hat{\rho}_\nu = q \mid \rho) f_\nu(\rho)\, d\rho,$$

$$\frac{1}{\nu}\log P(\hat{\rho}_\nu = q) \to -\inf_{\rho\in\Delta_c}\left[D_{\mathrm{KL}}(q \parallel \rho) + \sigma D_{\mathrm{KL}}(\bar{\rho} \parallel \rho)\right].$$

The infimum can be computed explicitly. Minimizing the strictly convex objective over the simplex gives, by the KKT conditions,

$$\rho = m_\sigma(q) = \frac{q + \sigma\bar{\rho}}{1+\sigma}.$$

Substituting this minimizer yields

$$\inf_{\rho\in\Delta_c}\left[D_{\mathrm{KL}}(q \parallel \rho) + \sigma D_{\mathrm{KL}}(\bar{\rho} \parallel \rho)\right] = J_\sigma(q;\bar{\rho}).$$

It remains to pass from a fixed type to a set A. The number of possible types is polynomial in ν, so summing over all types in $A \cap \mathcal{T}_\nu$ does not change the exponential rate. Therefore the event $\hat{\rho}_\nu \in A$ has rate given by the smallest type-wise cost over A. The regularity assumption $A \subseteq \overline{\mathrm{relint}(A)}$ ensures that boundary effects do not change the infimum. Thus

$$\lim_{\nu\to\infty}\frac{1}{\nu}\log P(\hat{\rho}_\nu \in A) = -\inf_{q\in A} J_\sigma(q;\bar{\rho}).$$

□

The rate has a useful information-theoretic interpretation. Since $m_\sigma(q) = (q + \sigma\bar{\rho})/(1+\sigma)$, we can write

$$J_\sigma(q;\bar{\rho}) = (1+\sigma)\,\mathrm{JS}_{1/(1+\sigma)}(q,\bar{\rho}),$$

where $\mathrm{JS}_{1/(1+\sigma)}$ denotes the weighted Jensen–Shannon divergence with mixture weight $1/(1+\sigma)$ on q. Thus, posterior averaging replaces the plug-in KL rate by a weighted Jensen–Shannon divergence between the empirical distribution and the posterior center.

The theorem has two consequences for reflection. First, the Bayesian objective has the same structural form as the plug-in objective: for a decision region A,

$$P(\hat{\rho}_\nu \in A) = \exp\left(-\nu \inf_{q\in A} J_\sigma(q;\bar{\rho}) + o(\nu)\right).$$

This suggests a qualitatively similar allocation behavior as discussed in App. C.

Second, the plug-in rate is recovered when posterior uncertainty vanishes. As $\sigma \to \infty$, the mixture point satisfies $m_\sigma(q) \to \bar{\rho}$, and therefore

$$J_\sigma(q;\bar{\rho}) \to D_{\mathrm{KL}}(q \parallel \bar{\rho}).$$

In the reflective setting, σ is the ratio between the effective posterior sample size in a bin and the number of hypothetical labels used to train the classifier in the replay. Since the replayed classifier is trained on

the already selected instances plus any newly allocated ones, $\sigma \leq 1$ and is strictly smaller than one for bins receiving additional labels. Thus, the posterior-averaged rate generally differs from the plug-in rate. Intuitively, posterior averaging tempers the confidence with which bins are classified as easy or hard. A plug-in criterion may under-allocate a bin because the current point estimate suggests a large top-two separation. The Bayesian criterion also accounts for posterior uncertainty about this separation. As a result, bins that appear easy under the current model can still receive mass when the posterior assigns non-negligible probability to them being difficult.

For the top-two decision error, one applies the theorem to the crossing region

$$A = \{q \in \Delta_c : q_2 \geq q_1\}.$$

This gives the posterior-averaged rate

$$\kappa^{\text{Bayes}} = \inf_{q_2 \geq q_1} J_\sigma(q; \bar{\rho}),$$

which plays the same role for Bayesian reflection as κ does in the fixed-distribution analysis.

E Optimization Details for the Bin-Allocation Relaxation

The finite-bin objective in Section 5.1 is an integer allocation problem. Rather than solving this problem directly, we use a continuous relaxation of the final bin counts. The relaxed problem retains a useful structure: it is separable across bins, has simple lower and upper bounds, and contains only one coupling constraint, namely the total budget. This appendix derives the KKT characterization used to solve the relaxation and describes the active-set procedure used in the implementation.

E.1 Continuous relaxation and KKT characterization

We consider optimization problems of the form

$$\min_{\nu_1,\ldots,\nu_m} \sum_{r=1}^{m} e^{-\phi_r(\nu_r)} \qquad \text{s.t.} \qquad l_r \leq \nu_r \leq u_r, \quad \sum_{r=1}^{m} \nu_r = n. \tag{15}$$

Here ν_r denotes the relaxed final count assigned to bin r. We assume the problem is feasible, i.e.,

$$\sum_{r=1}^{m} l_r \leq n \leq \sum_{r=1}^{m} u_r,$$

and that the functions ϕ_r are defined on the relevant interval. For the plug-in hard-risk asymptotic,

$$e^{-\phi_r(\nu)} = \widehat{A}_r \nu^{-1/2} e^{-\widehat{\kappa}_r \nu}, \qquad \phi_r(\nu) = \widehat{\kappa}_r \nu + \frac{1}{2}\log \nu - \log \widehat{A}_r.$$

Thus the binwise objective is a decreasing function of the assigned count.

The KKT conditions take a particularly simple form after passing to log space. Define

$$\psi_r(\nu) := \phi_r(\nu) - \log \phi_r'(\nu).$$

Equivalently,

$$e^{-\psi_r(\nu)} = \phi_r'(\nu) e^{-\phi_r(\nu)} = -\frac{d}{d\nu} e^{-\phi_r(\nu)},$$

so $e^{-\psi_r(\nu)}$ is the marginal decrease of the binwise objective.

Theorem 7. *Assume that each ϕ_r is twice differentiable, satisfies $\phi_r'(\nu) > 0$, and is concave on $[l_r, u_r]$. Then the relaxed problem in equation 15 is convex. If the feasibility conditions hold, its minimizers are characterized by the existence of a scalar $\lambda \in \mathbb{R}$ such that*

$$\psi_r(\nu_r) \begin{cases} \leq \lambda, & \nu_r = u_r, \\ = \lambda, & l_r < \nu_r < u_r, \\ \geq \lambda, & \nu_r = l_r. \end{cases} \tag{16}$$

Moreover, on the active set

$$\mathcal{I}_a = \{r : l_r < \nu_r < u_r\},$$

the solution satisfies

$$\nu_r = \psi_r^{-1}(\lambda), \qquad r \in \mathcal{I}_a.$$

If

$$\mathcal{I}_l = \{r : \nu_r = l_r\}, \qquad \mathcal{I}_u = \{r : \nu_r = u_r\},$$

and

$$n_a := n - \sum_{r \in \mathcal{I}_l} l_r - \sum_{r \in \mathcal{I}_u} u_r,$$

then λ is determined by the scalar equation

$$\sum_{r \in \mathcal{I}_a} \psi_r^{-1}(\lambda) = n_a. \tag{17}$$

Proof. The feasible set is convex because it is defined by box constraints and one linear equality constraint. The objective is convex because its Hessian is diagonal with entries

$$\frac{d^2}{d\nu^2} e^{-\phi_r(\nu)} = \left((\phi_r'(\nu))^2 - \phi_r''(\nu)\right) e^{-\phi_r(\nu)} > 0,$$

since $\phi_r'(\nu) > 0$ and $\phi_r''(\nu) \leq 0$.

Let

$$f_r(\nu) = e^{-\phi_r(\nu)}, \qquad M_r(\nu) = -f_r'(\nu) = \phi_r'(\nu) e^{-\phi_r(\nu)} = e^{-\psi_r(\nu)}.$$

The quantity $M_r(\nu)$ is the marginal improvement from increasing the count in bin r. Since f_r is convex and decreasing, M_r is positive and nonincreasing, and therefore ψ_r is nondecreasing.

The KKT conditions for equation 15 are necessary and sufficient. Let η be the multiplier of the equality constraint, and let $\mu_r^l, \mu_r^u \geq 0$ be the multipliers of the lower and upper bounds. The stationarity condition is

$$f_r'(\nu_r) + \eta - \mu_r^l + \mu_r^u = 0.$$

For an interior coordinate, $\mu_r^l = \mu_r^u = 0$, so

$$M_r(\nu_r) = \eta.$$

Writing $\lambda = -\log \eta$ gives $\psi_r(\nu_r) = \lambda$.

At a lower bound, $\nu_r = l_r$, complementary slackness gives $\mu_r^u = 0$ and hence

$$\eta - M_r(l_r) = \mu_r^l \geq 0.$$

Thus $M_r(l_r) \leq \eta$, or equivalently $\psi_r(l_r) \geq \lambda$. At an upper bound, $\nu_r = u_r$, complementary slackness gives $\mu_r^l = 0$ and hence

$$M_r(u_r) - \eta = \mu_r^u \geq 0.$$

Thus $M_r(u_r) \geq \eta$, or equivalently $\psi_r(u_r) \leq \lambda$. This proves equation 16.

For $r \in \mathcal{I}_a$, the equality $\psi_r(\nu_r) = \lambda$ gives $\nu_r = \psi_r^{-1}(\lambda)$. Substituting this into the budget constraint yields

$$n = \sum_{r \in \mathcal{I}_l} l_r + \sum_{r \in \mathcal{I}_u} u_r + \sum_{r \in \mathcal{I}_a} \psi_r^{-1}(\lambda),$$

which is equivalent to equation 17. Since the functions ψ_r are monotone, the left-hand side of equation 17 is monotone in λ, so the scalar equation determines the active-set solution. □

E.2 Active-set solution

The KKT characterization reduces the relaxed allocation problem to finding the correct active set and the corresponding scalar threshold λ. For a fixed partition into active, lower-bound, and upper-bound indices, the inactive variables are clamped to their bounds and the active variables are determined by equation 17. If the resulting solution violates a bound, the corresponding coordinate is moved to the appropriate inactive set. If an inactive coordinate violates the KKT inequality in equation 16, it is released back into the active set. This procedure is described in Alg. 1.

Algorithm 1 Active-set method for the relaxed bin-allocation problem

initialize $\mathcal{I}_a \leftarrow \{1, \ldots, m\}$, $\mathcal{I}_l \leftarrow \emptyset$, $\mathcal{I}_u \leftarrow \emptyset$
repeat
  $n_a \leftarrow n - \sum_{r \in \mathcal{I}_l} l_r - \sum_{r \in \mathcal{I}_u} u_r$
  solve $\sum_{r \in \mathcal{I}_a} \psi_r^{-1}(\lambda) = n_a$ for λ
  set $\nu_r \leftarrow \psi_r^{-1}(\lambda)$ for $r \in \mathcal{I}_a$, $\nu_r \leftarrow l_r$ for $r \in \mathcal{I}_l$, and $\nu_r \leftarrow u_r$ for $r \in \mathcal{I}_u$
  if there exists $r \in \mathcal{I}_a$ with $\nu_r < l_r$ **then**
    move one such r from $\mathcal{I}_a$ to $\mathcal{I}_l$
  else if there exists $r \in \mathcal{I}_a$ with $\nu_r > u_r$ **then**
    move one such r from $\mathcal{I}_a$ to $\mathcal{I}_u$
  else if there exists $r \in \mathcal{I}_l$ with $\psi_r(l_r) < \lambda - \epsilon$ **then**
    move one such r from $\mathcal{I}_l$ to $\mathcal{I}_a$
  else if there exists $r \in \mathcal{I}_u$ with $\psi_r(u_r) > \lambda + \epsilon$ **then**
    move one such r from $\mathcal{I}_u$ to $\mathcal{I}_a$
  else
    terminate
  end if
until termination

The release tests are the KKT inequalities with a numerical tolerance ϵ. A lower-bound coordinate is consistent only if $\psi_r(l_r) \geq \lambda$, while an upper-bound coordinate is consistent only if $\psi_r(u_r) \leq \lambda$. The active-set loop terminates when all active coordinates are feasible and all inactive coordinates satisfy these inequalities.

For a fixed active set, the scalar equation for λ is monotone and can be solved by bisection or Newton's method. In our implementation we use Newton iterations with safeguarded updates as described in Alg. 2. Let

$$F(\lambda) = \sum_{r \in \mathcal{I}_a} \psi_r^{-1}(\lambda) - n_a.$$

Then

$$F'(\lambda) = \sum_{r \in \mathcal{I}_a} \frac{1}{\psi_r'(\psi_r^{-1}(\lambda))}.$$

Thus one Newton update has the form

$$\lambda \leftarrow \lambda - \frac{F(\lambda)}{F'(\lambda)}.$$

It remains to solve the one-dimensional inverse problem $\psi_r(\nu) = \lambda$. For the plug-in hard-risk surrogate,

$$\phi_r(\nu) = \widehat{\kappa}_r \nu + \frac{1}{2} \log \nu - \log \widehat{A}_r,$$

and therefore

$$\psi_r(\nu) = \widehat{\kappa}_r \nu + \frac{1}{2} \log \nu - \log \widehat{A}_r - \log\left(\widehat{\kappa}_r + \frac{1}{2\nu}\right).$$

Algorithm 2 Solving the scalar threshold equation

initialize λ
for a fixed number of safeguarded Newton iterations **do**
  for each $r \in \mathcal{I}_a$ **do**
    solve $\psi_r(\nu_r) = \lambda$ for ν_r
  end for
  $F \leftarrow \sum_{r \in \mathcal{I}_a} \nu_r - n_a$
  $F' \leftarrow \sum_{r \in \mathcal{I}_a} 1/\psi_r'(\nu_r)$
  update $\lambda \leftarrow \lambda - F/F'$, with bisection fallback if the update leaves the bracket
end for

This function is monotone on the feasible interval. We solve $\psi_r(\nu) = \lambda$ by Newton's method, described in Alg. 3, again with a bisection fallback to keep the iterate inside the interval. The derivative used in Newton's method is

$$\psi_r'(\nu) = \widehat{\kappa}_r + \frac{1}{2\nu} + \frac{1}{2\nu^2\left(\widehat{\kappa}_r + \frac{1}{2\nu}\right)}.$$

Algorithm 3 Solving $\psi_r(\nu) = \lambda$

initialize ν inside the feasible interval
for a fixed number of safeguarded Newton iterations **do**
  $G \leftarrow \psi_r(\nu) - \lambda$
  $G' \leftarrow \psi_r'(\nu)$
  propose $\nu_{\text{new}} \leftarrow \nu - G/G'$
  if ν_{new} is outside the feasible interval or nonfinite **then**
    replace ν_{new} by a bisection step
  end if
  $\nu \leftarrow \nu_{\text{new}}$
end for

E.3 Numerical implementation

In practice, the active-set loop is initialized by first identifying lower-bound constraints. Starting with all indices active, we repeatedly solve the active-set equation without upper-bound activations and move all indices with proposed value below its lower bound to $\mathcal{I}_l$. This greedily removes regions to which the relaxed solution assigns little additional mass. After this lower-bound initialization, we run the full active-set procedure, allowing both lower- and upper-bound activations and releases according to the KKT inequalities and moving only one index at a time. This initialization substantially reduces the number of active variables in the subsequent scalar solves and is the main numerical adaptation used to obtain the running times reported in App. H. For a fixed active set, the main numerical cost is solving the scalar equations

$$\sum_{r \in \mathcal{I}_a} \psi_r^{-1}(\lambda) = n_a, \qquad \psi_r(\nu_r) = \lambda.$$

Both equations are one-dimensional and monotone. We solve them by safeguarded Newton iterations, with bisection fallback whenever an iterate leaves the feasible interval.

For efficiency, the inner inverse solve $\psi_r(\nu) = \lambda$ is initialized using asymptotic approximations to ψ_r^{-1}. For the histogram surrogate, $\psi_r(\nu)$ has the form

$$\psi_r(\nu) = \widehat{\kappa}_r \nu + O(\log \nu),$$

so the leading initializer is obtained by inverting the linear term and applying one correction from the logarithmic remainder. In practice, we also use two additional closed-form fallback initializers and select the candidate with the smallest residual before applying Newton iterations. These initializations substantially reduce the number of Newton steps in the inner loop and are used in all runtime comparisons.

F Equivalence of Bin- and Instance-Level Formulations

The kernelized construction in Section 5.3 replaces hard bin counts by local kernel counts around candidate instances. As a sanity check, this appendix shows that the construction reduces exactly to the finite-bin allocation problem when the kernel is chosen as the bin-indicator kernel. The argument is a change of variables: instance-level masses are aggregated within each bin, and the optimal instance-level allocation distributes the bin mass uniformly among instances in the same bin.

Suppose the candidate sequence is partitioned into bins $\mathcal{V}_1, \ldots, \mathcal{V}_m$, with $\mu_r = |\mathcal{V}_r|$ and $s = \sum_{r=1}^m \mu_r$. Let $w_r = \mu_r/s$. Consider the bin-level relaxation

$$\min_{N_1,\ldots,N_m} \sum_{r=1}^m w_r e^{-\phi_r(N_r)} \qquad \text{s.t.} \qquad l_r \le N_r \le u_r, \quad \sum_{r=1}^m N_r = n. \tag{18}$$

Here N_r is the final local count assigned to bin r, and l_r, u_r are the corresponding lower and upper feasible counts.

The corresponding instance-level formulation assigns a normalized mass ν_v to each candidate instance v. For an instance $v \in \mathcal{V}_r$, the bin-indicator kernel gives $T_v = \mu_r$, so the local count represented by ν_v is $\mu_r \nu_v$. Thus the instance-level problem becomes

$$\min_{\nu} \frac{1}{s} \sum_{r=1}^m \sum_{v \in \mathcal{V}_r} e^{-\phi_r(\mu_r \nu_v)} \qquad \text{s.t.} \qquad \frac{l_r}{\mu_r} \le \nu_v \le \frac{u_r}{\mu_r}, \quad \sum_{r=1}^m \sum_{v \in \mathcal{V}_r} \nu_v = n. \tag{19}$$

Proposition 8. *Assume that $t \mapsto e^{-\phi_r(t)}$ is convex for each bin r. Then the instance-level problem equation 19 and the bin-level problem equation 18 have the same optimal value in the following sense. If $\nu^\star$ solves equation 19, then the aggregated masses*

$$N_r^\star = \sum_{v \in \mathcal{V}_r} \nu_v^\star$$

solve equation 18. Conversely, if $N^\star$ solves equation 18, then the uniform within-bin assignment

$$\nu_v^\star = \frac{N_r^\star}{\mu_r}, \qquad v \in \mathcal{V}_r,$$

solves equation 19.

Proof. Let ν be any feasible instance-level allocation and define its aggregated bin masses by

$$N_r = \sum_{v \in \mathcal{V}_r} \nu_v.$$

The bounds in equation 19 imply $l_r \le N_r \le u_r$, and the total-mass constraint gives $\sum_r N_r = n$. Thus $N = (N_1, \ldots, N_m)$ is feasible for equation 18.

By Jensen's inequality, for each bin r,

$$\frac{1}{\mu_r} \sum_{v \in \mathcal{V}_r} e^{-\phi_r(\mu_r \nu_v)} \ge e^{-\phi_r\left(\frac{1}{\mu_r} \sum_{v \in \mathcal{V}_r} \mu_r \nu_v\right)} = e^{-\phi_r(N_r)}.$$

Multiplying by $\mu_r/s = w_r$ and summing over bins gives

$$\frac{1}{s} \sum_{r=1}^m \sum_{v \in \mathcal{V}_r} e^{-\phi_r(\mu_r \nu_v)} \ge \sum_{r=1}^m w_r e^{-\phi_r(N_r)}.$$

Thus every feasible instance-level allocation has objective value at least as large as the bin-level objective value of its aggregation.

Conversely, let N be feasible for equation 18 and assign $\nu_v = N_r/\mu_r$ for all $v \in \mathcal{V}_r$. This assignment is feasible for equation 19, and the Jensen inequality above holds with equality. Therefore the two problems have the same optimal value, and the stated correspondence between optimizers follows. □

For the bin-indicator kernel, the quantities in Section 5.3 reduce to

$$T_v = \mu_r, \qquad l_v = l_r \quad \text{for } v \in \mathcal{V}_r.$$

Consequently,

$$n_{i-1}^{\text{ker}} = \sum_{v=1}^{s} \frac{l_v}{T_v} = \sum_{r=1}^{m} l_r,$$

so the kernelized budget constraint reduces to the original finite-bin budget constraint. The proposition therefore shows that optimizing the instance-level kernelized formulation is equivalent, in this special case, to optimizing the bin-level allocation and distributing each bin's mass uniformly among the candidate instances in that bin.

G Ablation

We ablate the main design choices of the practical query strategy. We compare the hard decision error and the stochastic zero-one risk in Figure 7, vary the effective number of bins in Figure 8, and study the effect of the kernel choice in Figure 9. We ran 5 repetition per experiment and display the mean across repetitions in the learning curve.

Hard versus soft zero-one risk. Figure 7 compares the hard decision error and stochastic zero-one risk variants. On MNIST, the distinction between the hard and soft variants has little effect. Since the classes are well separated, the method quickly moves from broad exploration to boundary-focused selection, and both objectives induce similar allocations. On Food-101, the hard-risk variant gives a substantial improvement, especially in later cycles. This aligns with the discussion in Section 4: the hard objective weights a bin by the probability of an incorrect majority decision, whereas the soft excess risk additionally contains the factor $\rho_{r,1} - \rho_{r,2}$. Near the decision boundary this factor can be small, so the soft objective may downweight precisely those regions where additional labels are useful for resolving the majority class.

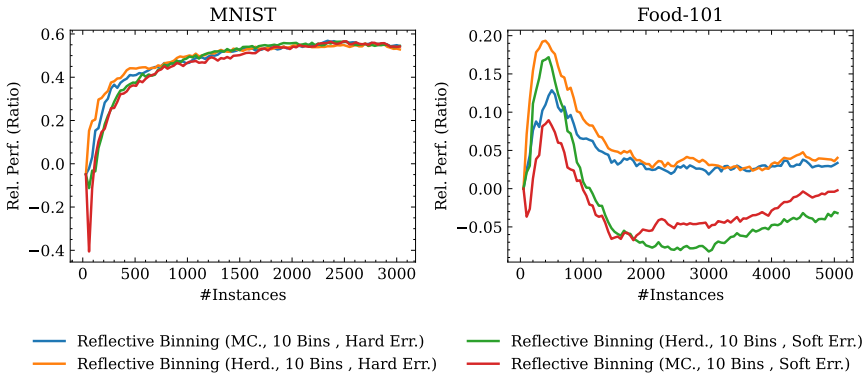


Figure 7: Hard versus soft zero-one variants using linear probing on DINOv3-ViT-S/16 embeddings.

Effective number of bins. Figure 8 shows the effect of varying the effective number of bins n_{bins}, which controls the kernel length scale through equation 12. Increasing n_{bins} degrades performance on MNIST, while providing a slight improvement on Food-101 during earlier cycles. This suggests that simple datasets benefit from larger neighborhoods and stronger aggregation, whereas more complex datasets can benefit from finer local resolution. Across the full learning curve, however, smaller values appear more stable.

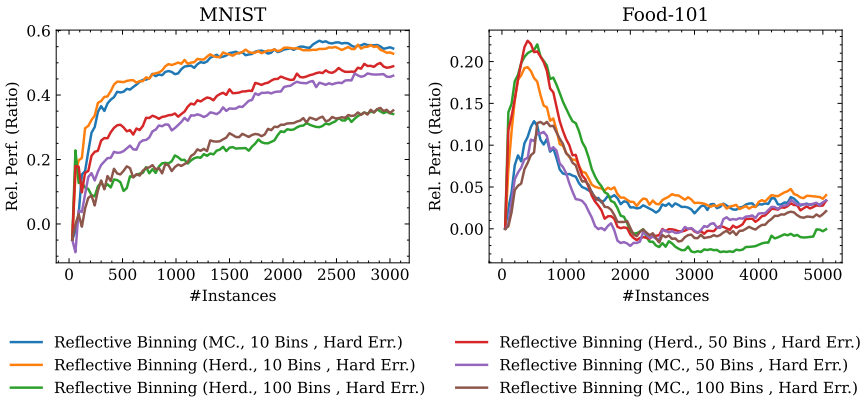


Figure 8: Effect of the effective number of bins n_{bins} using linear probing on DINOv3-ViT-S/16 embeddings.

Kernel choice. Figure 9 compares Laplace and radial basis function (RBF) kernels, each applied either to raw or normalized representations. For representations $z = h_\omega(x)$ and $z' = h_\omega(x')$, the Laplace kernel is

$$K_\ell^{\text{Lap}}(z, z') = \exp[-\ell\|z - z'\|]\,,$$

and the RBF kernel is

$$K_\ell^{\text{RBF}}(z, z') = \exp\left[-\frac{\ell}{2}\|z - z'\|^2\right].$$

The normalized variants apply the same formulas to

$$\bar{z} = \frac{z}{\|z\|}, \qquad \bar{z}' = \frac{z'}{\|z'\|}.$$

For the normalized RBF kernel, this is equivalent to an exponential cosine-distance kernel, since

$$\frac{1}{2}\|\bar{z} - \bar{z}'\|^2 = 1 - K_{\cos}(z, z').$$

All kernels use the same effective-bin heuristic from equation 12 to set the length scale.

Overall, the curves are similar, indicating that the method is not highly sensitive to this choice. The differences induced by Monte Carlo versus herding are more pronounced than the differences between kernels. On MNIST, Laplace-type kernels give a slight early advantage, whereas on Food-101 the RBF-type kernels perform slightly better. We use the exponential cosine kernel in the main experiments because it provides a stable angular notion of neighborhood for frozen representations.

H Running Time

We compare the query times of the evaluated strategies in the few-cycle / large-batch setting from Section 6. All timings were measured on the same CPU and include only the query-selection step, not model training or feature extraction. Figure 10 shows cumulative query time over active learning cycles.

The runtime of our method contains both a fixed initialization cost and a cycle-dependent optimization cost. The initialization cost comes mainly from setting the kernel scale through the effective-bin heuristic. The cycle-dependent cost is determined by the active-set solver for the relaxed allocation problem. Its runtime depends on how quickly the active lower and upper bounds are identified, and can therefore vary across cycles.

Our method is slower than lightweight strategies such as TypiClust and margin, which focus primarily on exploration or exploitation. However, it is substantially faster than BADGE on CIFAR-100 and comparable to BADGE on CIFAR-10. The difference is consistent with the structure of the optimization routine: most

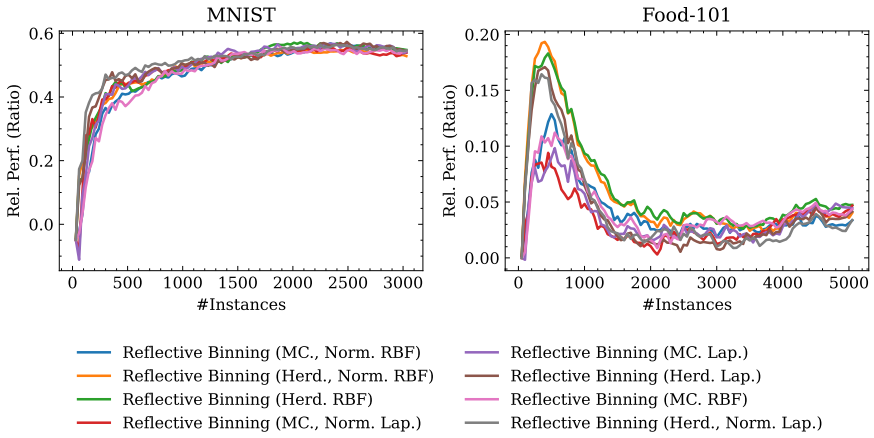


Figure 9: Effect of the kernel choice using linear probing on DINOv3-ViT-S/16 embeddings.

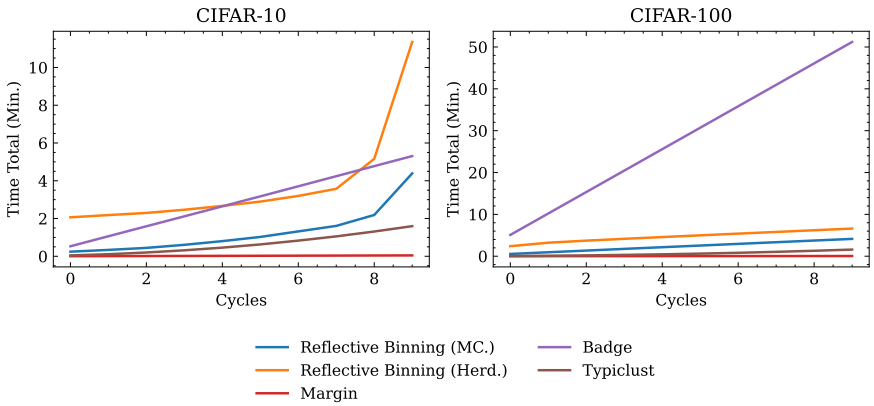


Figure 10: Query times using linear probing on DINOv3-ViT-S/16 embeddings in the large-batch / few-cycle regime.

of the work is spent solving the relaxed allocation problem and identifying inactive candidates. To accelerate this step, we initialize the active set by greedily fixing candidates that violate the lower bound, corresponding to regions where the relaxed solution assigns little additional mass. The numerical details of the active-set solver are described in App. E.

I Learning Curves

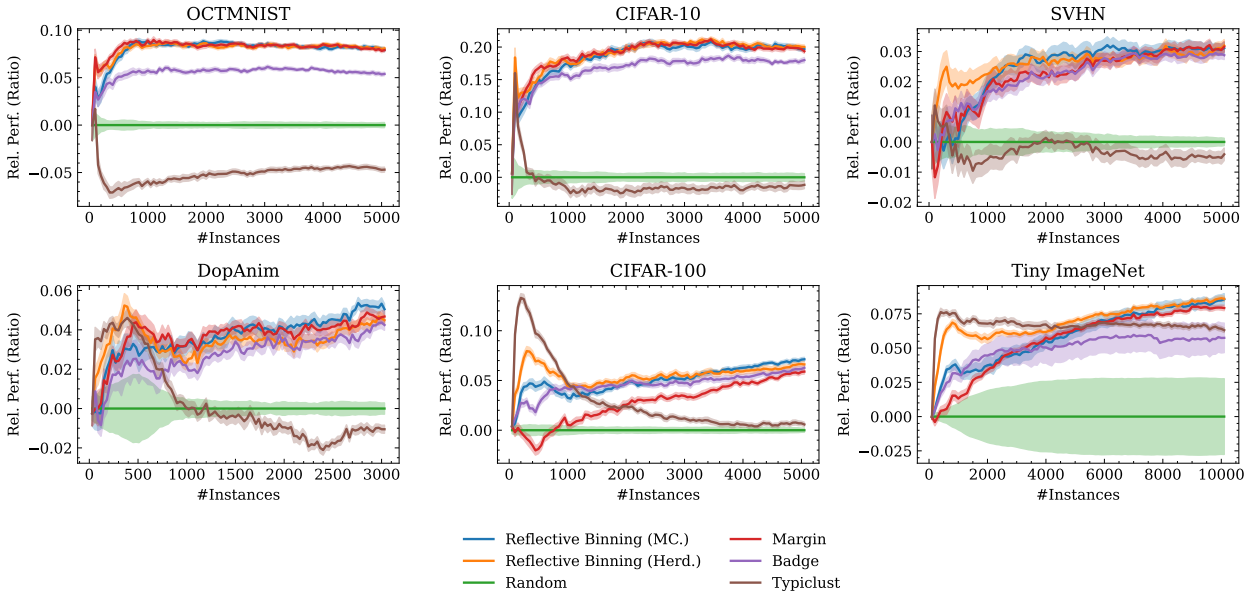


Figure 11: Learning curves using linear probing on CLIP-ViT-B/16 embeddings.

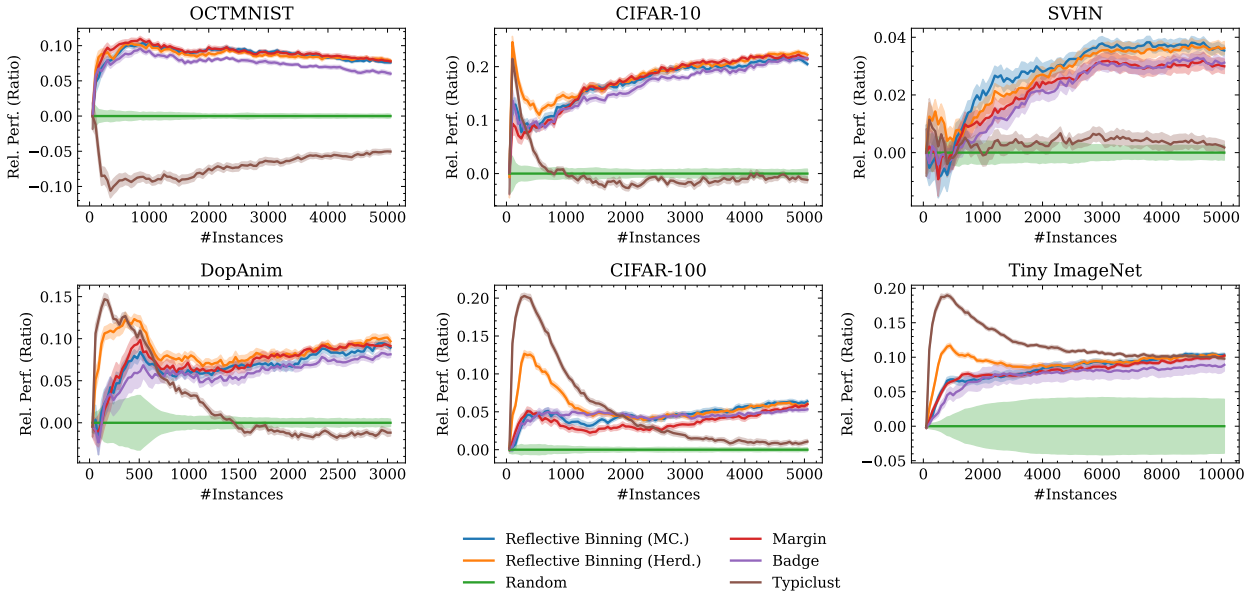


Figure 12: Learning curves using linear probing on DINOv3-ViT-S/16 embeddings.

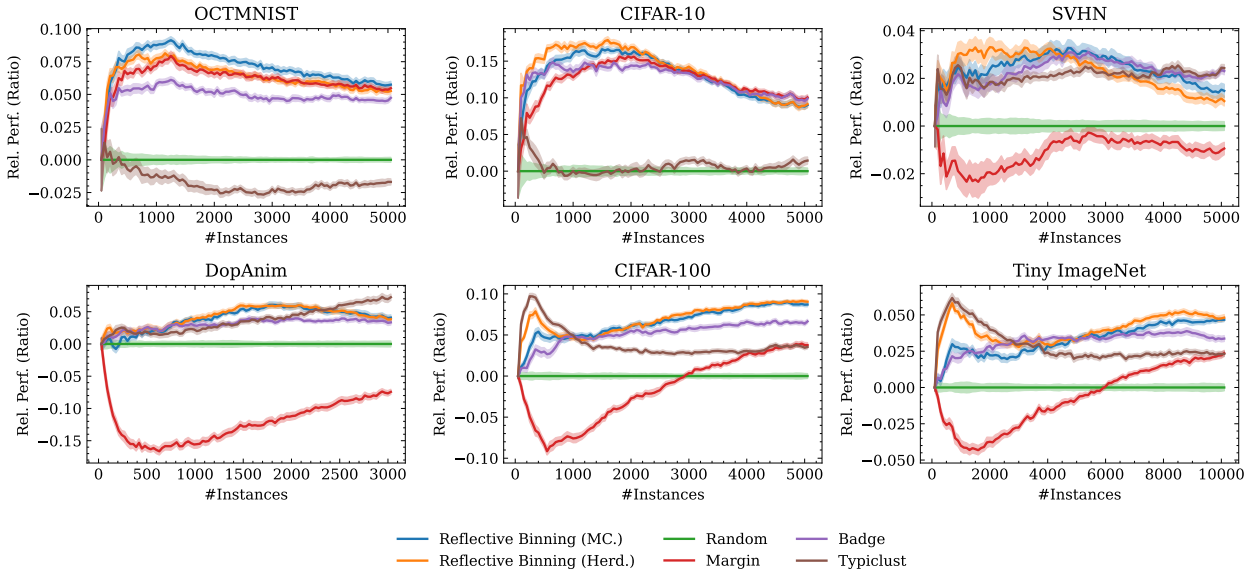


Figure 13: Learning curves using linear probing on CLIP-ViT-B/16 embeddings under duplicates.

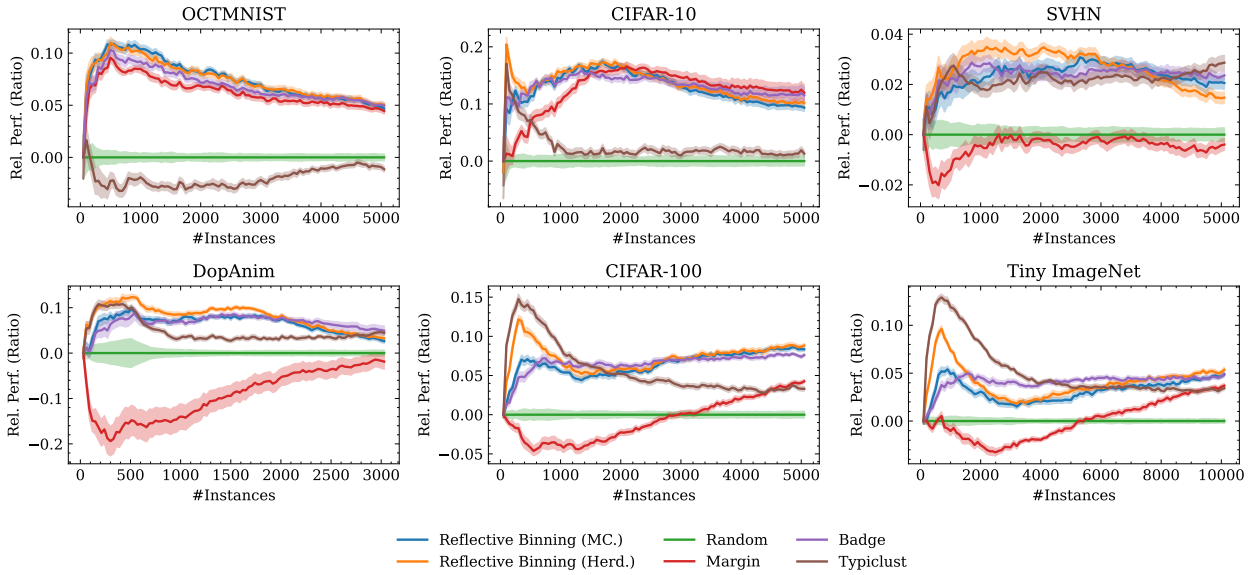


Figure 14: Learning curves using linear probing on DINOv3-ViT-S/16 embeddings under duplicates.

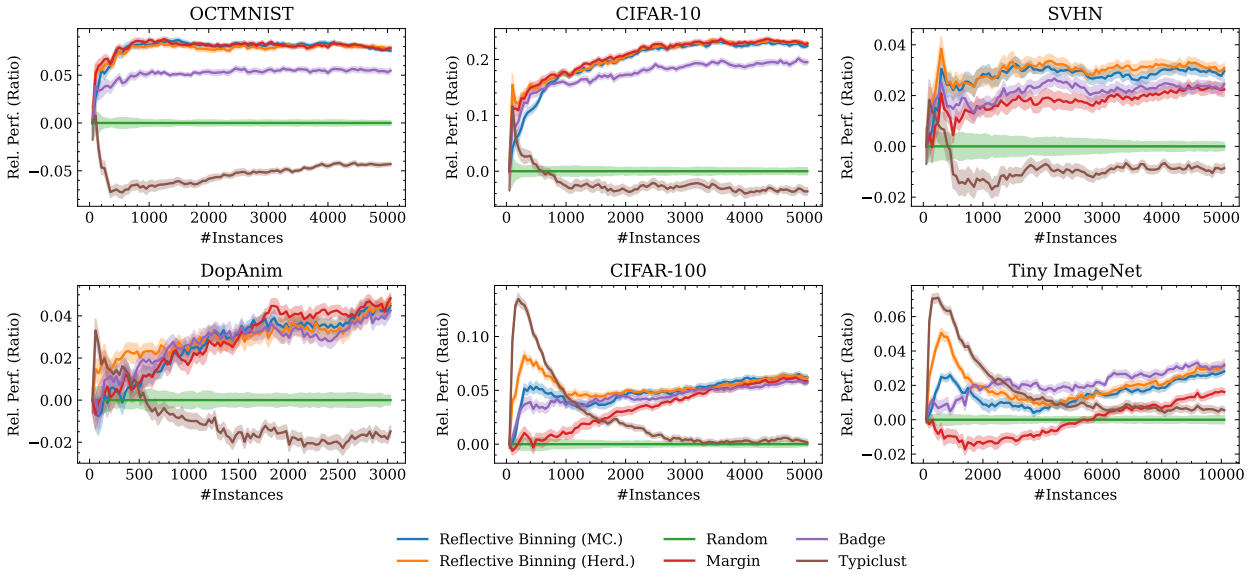


Figure 15: Learning curves using linear probing on CLIP-ViT-B/16 embeddings under label noise.

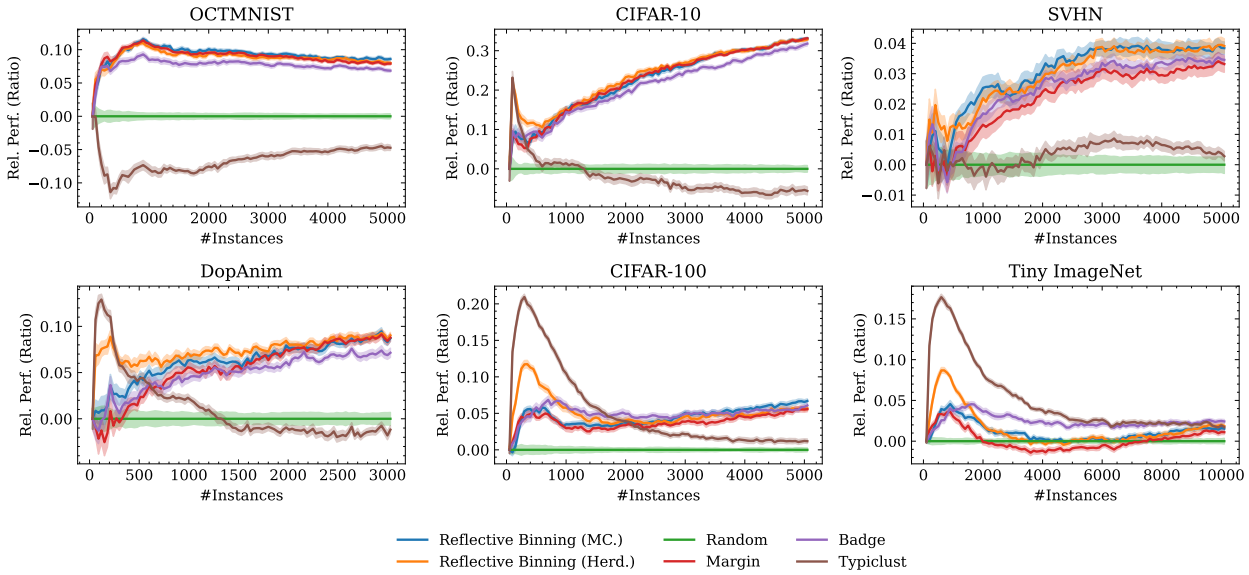


Figure 16: Learning curves using linear probing on DINOv3-ViT-S/16 embeddings under label noise.

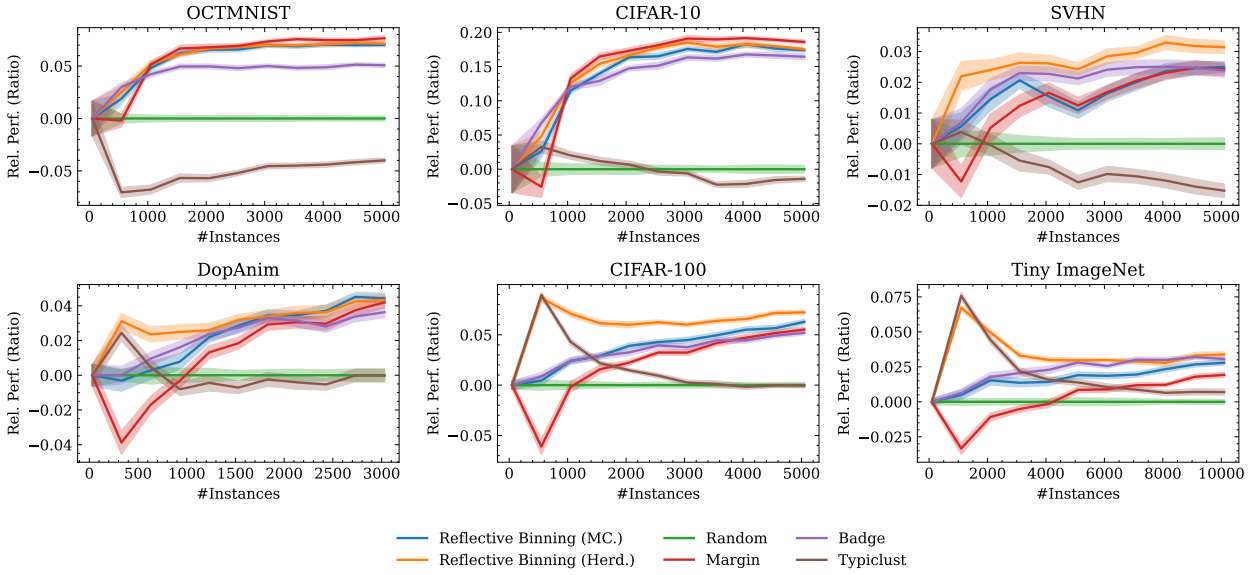


Figure 17: Learning curves using linear probing on CLIP-ViT-B/16 embeddings under few-cycles/large-batches.

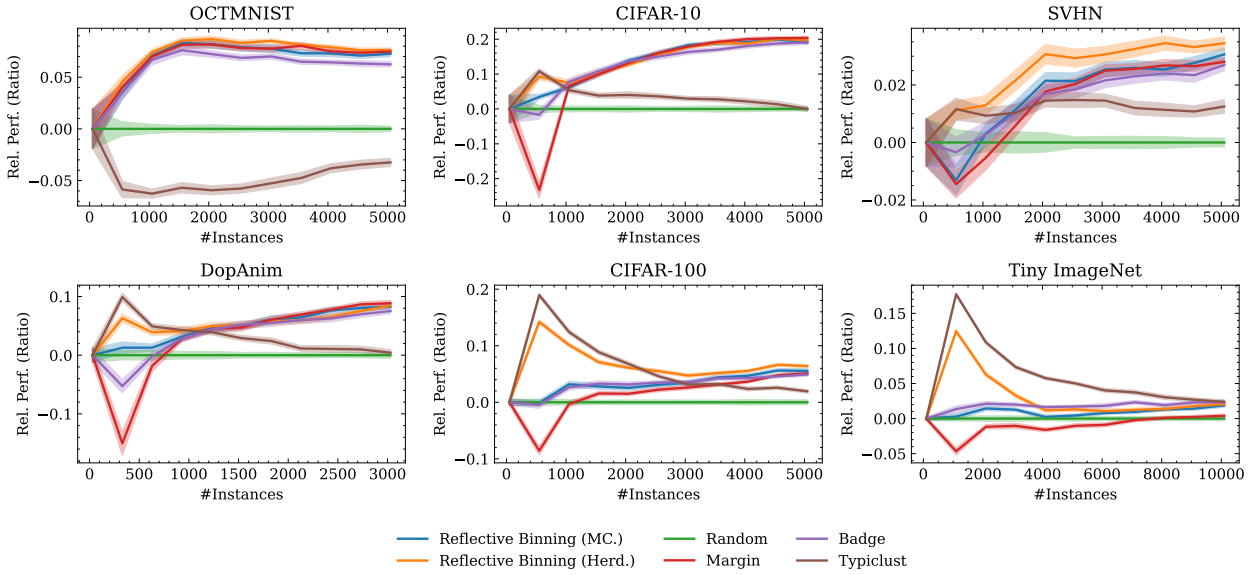


Figure 18: Learning curves using linear probing on DINOv3-ViT-S/16 embeddings under few-cycles/large-batches.