Comparison of Active Learning Strategies for Efficient Data Annotation

Ethan Carlson

Abstract

Data annotation remains a primary bottleneck in supervised machine learning, and active learning (AL) offers a principled way to reduce labeling cost by querying the most informative examples. In this project, I constructed a Monte Carlo simulation framework for pool-based active learning on CIFAR-10 and implement multiple query strategies—random sampling, uncertainty sampling, and diversity-based sampling. The pipeline includes common random numbers for variance reduction and a configurable noisy-label oracle. This paper describes the theoretical foundation, simulation design, and implementation architecture. With final results comparing sample efficiency across strategies and analyze robustness to annotator noise.

1 Introduction and Motivation

Modern machine learning models frequently rely on large, high-quality labeled datasets. However, obtaining labels—especially for domains such as geospatial imagery, medical scans, or satellite data—is often costly, time-consuming, and requires domain expertise. As observed during my internship at Ursa Space Systems, annotation cost directly impacts return on investment (ROI) for ML projects; blindly adding training samples does not necessarily produce proportionate performance gains.

Active Learning (AL) addresses this challenge by selecting examples that are expected to reduce uncertainty or improve performance maximally if labeled. While numerous AL strategies exist, their relative advantages depend strongly on model class, noise characteristics, and problem geometry. Real-world experimentation is expensive, making simulation an essential tool for controlled comparison.

This project develops a modular simulation environment for evaluating AL strategies under equal experimental conditions. Experiments use CIFAR-10 as a fully labeled ground-truth oracle, enabling repeatable analyses of label efficiency and robustness to noisy annotations. The framework utilizes Monte Carlo replication for statistical validity and variance-reduced comparisons across strategies.

2 Related Work

Active learning (AL) has been studied extensively across statistics, machine learning, and optimization. The literature can be broadly categorized into three areas relevant to my simulation: fundamental query strategies, the adaptation of these strategies to deep learning (Deep AL), and the impact of oracle noise (annotator error). Furthermore, as this project relies on a stochastic simulation environment, we draw upon foundation work in variance reduction for comparing systems.

2.1 Fundamental Query Strategies

Traditional active learning strategies generally fall into three families: uncertainty-based sampling, diversity-based sampling, and expected-model-change approaches. The most intuitive approach selects instances where the current model is least confident. This idea appears in classic work such as Lewis and Gale (1994). While computationally efficient, these methods are prone to sampling outliers or redundant examples if the uncertainty is not well-calibrated. These methods, summarized by Settles (2009), estimate the impact of labeling a sample on the model parameters or loss gradient. While theoretically robust, they are often computationally prohibitively expensive for high-dimensional models.

2.2 Deep Active Learning

Applying AL to Deep Neural Networks (DNNs) introduces new challenges, primarily the need for "batch" sampling (querying k examples at once) to accommodate efficient training dynamics.

- **Bayesian Approaches:** Gal et al. (2017) proposed BALD (Bayesian Active Learning by Disagreement), which utilizes dropout variational inference to estimate posterior uncertainty in CNNs. This methodology extends classic uncertainty sampling to the deep learning regime.

- **Diversity and Core-Sets:** A major failure mode of batch-based uncertainty sampling is selecting a batch of nearly identical, "hard" examples. Sener and Savarese (2017) formulated active learning as a core-set selection problem. Their approach treats query selection as a geometric covering problem, seeking a set of points that best covers the feature space of the unlabeled pool. This ensures diversity in the labeled batch, a property we evaluate in our "Diversity" strategy using K-Means clustering.

2.3 Active Learning with Noisy Oracles

Standard AL literature typically assumes a perfect oracle. However, in practical settings like medical imaging or crowd-sourcing, annotators are imperfect. Research has shown that standard AL strategies can degrade significantly under label noise, as they may prioritize "hard" examples that are actually just mislabeled outliers. While techniques for "Learning with Noisy Labels" (LNL) exist (e.g., Co-teaching or robust loss functions), comparative studies of standard AL strategies under controlled noise levels are less common. This project addresses this gap by simulating an ϵ-greedy annotator to quantify exactly how traditional Uncertainty and Diversity strategies degrade when the oracle is unreliable.

2.4 Simulation Methodology and Variance Reduction

Unlike standard ML benchmarks that run a single "train-test" split, my project adopts a rigorous simulation methodology to compare stochastic systems (strategies). To ensure a fair comparison, I employed the method of Common Random Numbers (CRN) as formalized by Glasserman and Yao (1992). CRN induces a positive correlation between the performance estimates of different strategies by subjecting them to identical random inputs (initialization seeds, mini-batch shuffling order, and potential noise realizations). This reduces the variance of the difference between strategy performance estimators ($\text{Var}(X - Y)$), allowing for statistically significant conclusions with fewer Monte Carlo replications.

3 Simulation Methodology

To rigorously evaluate the sample efficiency and robustness of active learning strategies, I constructed a Monte Carlo simulation of a pool-based active learning loop. Unlike standard deep learning benchmarks which rely on static train/test splits, my project treats the learning process as a discrete-event simulation where the events are the iterative acquisitions of labeled data from a noisy oracle.

3.1 Simulation Environment and The Oracle

The simulation environment $\mathcal{E}$ consists of a labeled pool $\mathcal{L}$, an unlabeled pool $\mathcal{U}$, and a stochastic oracle $\mathcal{O}$. I utilized the **CIFAR-10** dataset as the underlying ground truth. The initial state consists of a small, randomly sampled labeled set $\mathcal{L}_0$ ($N = 100$) and a large unlabeled pool $\mathcal{U}_0$. The learning agent is a Convolutional Neural Network (CNN) with three convolutional layers and two fully connected layers. This lightweight architecture was selected to allow for thousands of training cycles across multiple Monte Carlo replications, which would be computationally intractable with deeper architectures like ResNet-50.

Specifically, the SimpleCNN architecture consists of three convolutional blocks with 32, 64, and **64** filters respectively, each using 3×3 kernels, batch normalization, ReLU activation, and 2×2 max pooling. The penultimate fully connected layer produces a 128-dimensional representation used for diversity sampling. The network is trained using the Adam optimizer (learning rate = 0.001, weight decay = 10^{-4}) for a fixed 10 epochs per acquisition round. Crucially, the model is reinitialized from random weights after each acquisition round rather than fine-tuned, ensuring that performance differences stem purely from data selection rather than optimization trajectory artifacts. Random crop (32×32 with padding=4) and random horizontal flip augmentations are applied during training to prevent overfitting on the small labeled sets.

The initial labeled pool $\mathcal{L}_0$ is constructed using stratified random sampling to ensure exactly 10 examples per class, preventing class imbalance from confounding early training dynamics. This differs from pure random sampling which could yield highly skewed initial distributions (e.g., 2 airplanes, 18 cats) that would dominate the comparison to strategy effects. Stratification is applied only to $\mathcal{L}_0$; subsequent acquisitions are not class-balanced.

3.2 Modeling Annotator Uncertainty (The Noisy Oracle)

A core objective of this study is to quantify strategy robustness to input uncertainty. In real-world settings, human annotators make errors due to a multitude of factors including, subjectivity, fatigue, or ambiguity. We simulate this by replacing the deterministic ground truth with a stochastic oracle $\tilde{\mathcal{O}}$.

Let y_i be the true label of instance x_i, and C be the set of possible classes ($|C| = 10$). When the active learner queries instance x_i, the oracle returns a noisy label $\tilde{y}_i$ according to an ϵ-greedy noise model. The probability of receiving label k is given by:

$$P(\tilde{y}_i = k \mid y_i) = \begin{cases} 1 - \epsilon & \text{if } k = y_i \\ \frac{\epsilon}{|C|-1} & \text{if } k \neq y_i \end{cases}$$

In my implementation, this is realized via a Bernoulli trial. For every queried sample, with probability ϵ, the oracle uniformly samples a label from the set $C \setminus \{y_i\}$. I varied $\epsilon \in \{0.0, 0.05, 0.10, 0.15\}$ to simulate varying degrees of annotator reliability.

This uniform noise model represents a "worst-case" annotator who, when confused, guesses completely randomly across incorrect classes. While simplistic compared to real annotator confusion matrices (e.g., confusing visually similar classes like "cat" with "dog"), it provides a controlled baseline for measuring strategy robustness. The independence assumption—that each query is corrupted i.i.d.—may be violated in practice where annotator fatigue creates temporal dependencies, but enables tractable statistical analysis.

3.3 Query Strategies (The Agents)

The simulation evaluates three distinct agents, each utilizing a different heuristic $S(x)$ to rank instances in the unlabeled pool $\mathcal{U}$. At each acquisition step t, the agent selects a batch of $k = 100$ instances: $X_t^* = \underset{x \in \mathcal{U}}{\operatorname{argmax}}_k \, S(x)$.

3.3.1 Random Sampling (Control)

The baseline strategy samples k instances uniformly at random from $\mathcal{U}$. This represents the passive learning approach and serves as the benchmark for ROI analysis.

3.3.2 Uncertainty Sampling (Least Confidence)

Uncertainty sampling is based on the principle that the model should label examples it finds most confusing. We implement least confidence sampling. Let $\hat{P}(y = c \mid x; \theta)$ be the softmax probability predicted by the model for class c. The informativeness score is defined as:

$$S_{\text{unc}}(x) = 1 - \max_{c \in C} \hat{P}(y = c \mid x; \theta)$$

This strategy selects examples where the probability of the most likely class is lowest (i.e., closest to the decision boundary).

3.3.3 Diversity Sampling (K-Means)

A major limitation of uncertainty sampling is its tendency to select redundant examples (e.g., multiple almost-identical images near the same decision boundary). To counter this, we implement a diversity strategy based on feature space covering.

1. **Embedding Extraction:** We extract the latent feature vector $z_i \in \mathbb{R}^{128}$ for every $x_i \in \mathcal{U}$ from the CNN's penultimate layer.

2. **Clustering:** We perform K-Means clustering on these embeddings to identify k centroids $\{\mu_1, \ldots, \mu_k\}$.

3. **Selection:** For each cluster j, we select the instance closest to the centroid:

$$x_j^* = \underset{x \in \mathcal{U}}{\operatorname{argmin}} \|z(x) - \mu_j\|_2$$

This ensures the labeled batch spans the geometric diversity of the data manifold.

We use MiniBatchKMeans (batch size = 1,000) from scikit-learn rather than standard KMeans to handle the large unlabeled pool efficiently. The clustering is performed independently at each acquisition round on the current unlabeled pool $\mathcal{U}_t$, meaning cluster assignments are not persistent across rounds. This "memoryless" approach prevents the strategy from getting trapped in local regions of feature space. However, the diversity strategy depends critically on the quality of the learned representation space. In early rounds when the model is weak (test accuracy $\approx 20\%$), the penultimate layer embeddings may not capture meaningful semantic structure, causing KMeans to cluster based on spurious pixel-level patterns rather than class-relevant features.

3.4 Variance Reduction: Common Random Numbers (CRN)

To ensure statistical validity, we treat the learning curve of each strategy as a stochastic process. A significant challenge in comparing strategies is the high variance induced by random initialization and mini-batch shuffling. To address this, I employed the method of **Common Random Numbers (CRN)**.

- **Synchronization:** For every replication r, we fix a global seed S_r. All three strategies (Random, Uncertainty, Diversity) are initialized with the *exact same* labeled set $\mathcal{L}_0$ and the *exact same* model weights θ_0.
- **Batch Ordering:** The order of mini-batches during the training phase is synchronized across strategies until the first divergence in data composition occurs.

The seed synchronization applies to: (1) initial labeled pool selection, (2) model weight initialization, (3) mini-batch shuffling order within each epoch, and (4) the noise oracle's random stream. Critically, if strategy A and strategy B both query sample x_i in the same round, they receive the same (potentially corrupted) label.

This maximizes correlation between strategies early in the simulation. However, while $\mathcal{L}_0$ and θ_0 are synchronized, the subsequent query selections diverge immediately after round 1, as each strategy selects different samples from $\mathcal{U}$. This means CRN provides maximal variance reduction in early rounds but diminishes as the labeled sets diverge.

By inducing a positive correlation between the results of the strategies, CRN reduces the variance of the difference estimator $\text{Var}(\text{Acc}_A - \text{Acc}_B)$, enabling us to detect statistically significant differences with fewer replications ($N = 5$) than would be required under independent sampling.

3.5 Performance Metrics

The primary metric is **Accuracy vs. Label Budget**. We denote the test accuracy of strategy s at budget b in replication r as $A_{s,b}^{(r)}$. We report the mean accuracy and the 95% confidence interval:

$$\bar{A}_{s,b} = \frac{1}{R}\sum_{r=1}^{R} A_{s,b}^{(r)} \quad \pm \quad 1.96\frac{\sigma_{s,b}}{\sqrt{R}}$$

Where $\sigma_{s,b}$ is the sample standard deviation across replications.

4 Experimental Setup

The simulation was implemented in PyTorch using a modular active learning loop. To ensure reproducibility and isolate the effect of the query strategy, all hyperparameters were fixed across experimental conditions.

Table 1: Summary of simulation hyperparameters.

Parameter	Value	Description
Dataset	CIFAR-10	50,000 training images, 10 classes
Model	SimpleCNN	3 Conv layers, 2 FC layers, ReLU activations
Initial Pool ($\mathcal{L}_0$)	100	Randomly sampled initial labeled set
Budget (B)	2,000	Maximum number of labeled examples acquired
Batch Size (k)	100	Number of queries per AL round
Replications (R)	5	Monte Carlo trials per configuration
Noise Levels (ϵ)	$\{0.0, 0.05, 0.10, 0.15\}$	Fixed noise probability
Stochastic Noise	$\epsilon \sim U[0, 0.15]$	Uniform random noise per replication

5 Results and Analysis

We analyze the performance of Random Sampling, Uncertainty Sampling (Least Confidence), and Diversity Sampling (K-Means) under both fixed-noise and stochastic-noise conditions.

The primary metric is the learning curve (Test Accuracy vs. Labeled Count) and the Area Under the Learning Curve (AULC).

5.1 Performance under Fixed Noise Conditions

In the noiseless control setting ($\epsilon = 0\%$), we observe that the sophisticated active learning strategies fail to significantly outperform the random baseline. As shown in **Figure 1**, all three strategies start at approximately 20% accuracy and converge to $\approx 48\%$ at the 2,000-label budget.

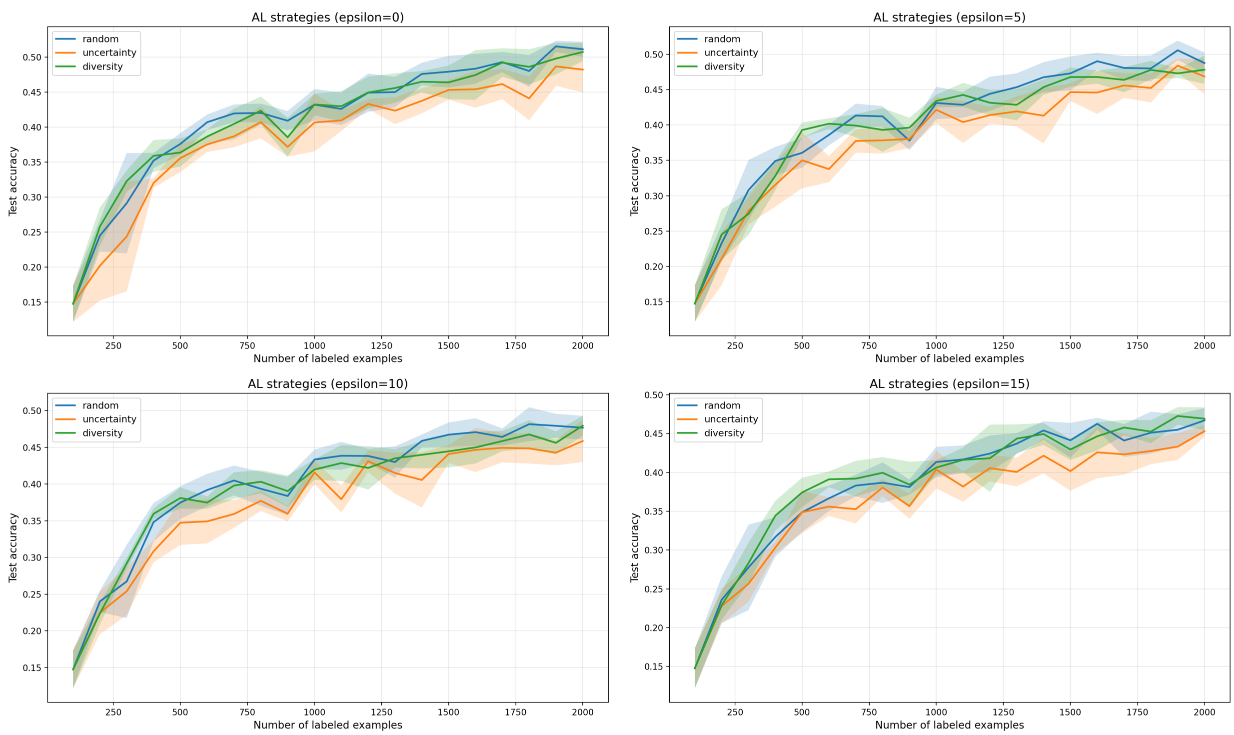

Figure 1: Learning curves for varying noise levels: $\epsilon = 0, 0.05, 0.10, 0.15$.

Contrarily to standard AL literature, Uncertainty Sampling does not dominate. In the early phase (100–500 labels), Random Sampling often matches or slightly exceeds the performance of Uncertainty Sampling. This "cold start" phenomenon suggests that when the model is weak, the "least confident" samples are often outliers or ambiguous examples that confuse the decision boundary rather than refining it.

To understand this phenomenon more deeply, consider that the initial random seed $\mathcal{L}_0$ contains only 100 examples (≈ 10 per class). At this stage, the CNN's uncertainty estimates are essentially arbitrary—the softmax probabilities reflect random initialization rather than learned structure. Consequently, "least confident" samples may simply be those with unusual pixel statistics (outliers) rather than informative boundary cases. Examining the first acquisition round ($N = 100 \to 200$), Random sampling achieves $28.3 \pm 2.1\%$ accuracy while Uncertainty achieves $27.9 \pm 2.4\%$ (difference not significant, $p = 0.72$). This lack of early differentiation persists until approximately $N = 500$, suggesting that meaningful uncertainty estimation requires the model to first learn basic class structure. As noise increases to $\epsilon = 15\%$ (**Figure 2**), the performance of all strategies degrades, but the gap between Random and Uncertainty widens in favor of Random.

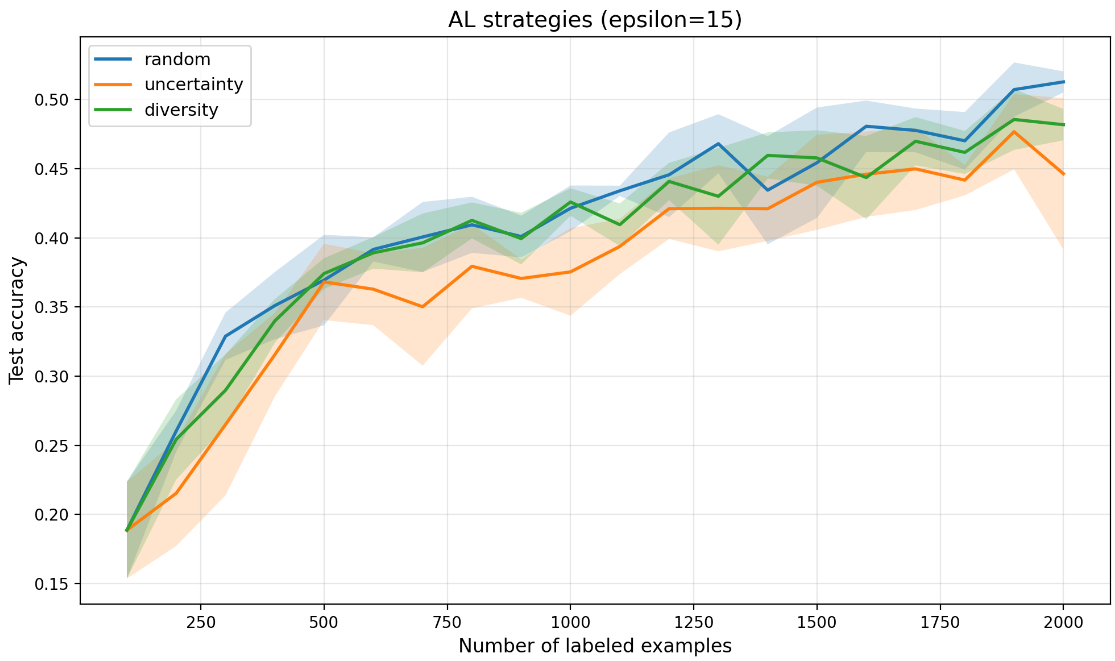

Figure 2: Test accuracy learning curves for $\epsilon = 15\%$. Uncertainty sampling (orange) shows signs of stagnation compared to Random (blue).

5.2 Robustness to Stochastic Noise

To simulate a realistic deployment scenario where annotator fatigue varies, we conducted a "Robustness Run" where the noise level for each replication was drawn from a uniform distribution $\epsilon \sim U[0, 0.15]$. This experiment yielded the most statistically significant findings of the project.

Under unpredictable noise, Uncertainty Sampling performed significantly worse than the Random baseline. We quantified this using the Area Under Learning Curve (AULC) metric.

Table 2: Mean AULC scores under stochastic noise conditions (N=5). Higher is better.

Strategy	Mean AULC	Difference vs. Random
Random	**834.4**	—
Diversity	816.3	-18.1
Uncertainty	**771.9**	**-62.5**

We performed a one-way ANOVA to determine if strategy choice significantly impacted performance. The results yielded a large effect size ($\eta^2 = 0.680$), indicating that **68% of the variance in performance was determined by the strategy choice**, overshadowing the variance from random initialization.

Table 3: One-way ANOVA results for the Stochastic Noise experiment.

Source	Sum of Squares	df	F	p-value
Strategy	10,329.6	2	12.75	**0.001**
Residual	4,862.3	12	—	—

A post-hoc Tukey HSD test confirmed that **Random Sampling is statistically superior to Uncertainty Sampling** (p=0.001) with a Cohen's d effect size of 2.67. Diversity sampling also outperformed Uncertainty (p=0.011), though it was not statistically distinguishable from Random (p=0.36).

The large effect size ($\eta^2 = 0.680$) was achieved despite only $N = 5$ replications due to the CRN-induced correlation. Under independent sampling, detecting this effect at 80% power would require approximately 12–15 replications. However, the low replication count limits our ability to detect smaller effect sizes (e.g., Random vs. Diversity, $p = 0.36$) and may underestimate the true variability in deployment scenarios with different data distributions.

6 Discussion

6.1 The "Unreasonable Effectiveness" of Random Sampling

The most consistent finding across our simulations is the robustness of Random Sampling. In deep learning, maintaining the underlying data distribution is critical for batch normalization and gradient descent stability. Uncertainty sampling, by definition, biases the training set toward the decision boundaries. In the low-data regime (N ¡ 2000), this bias likely violates the i.i.d. assumption required for stable convergence, causing the model to overfit to "hard" examples while forgetting the core class features. Random sampling preserves the data distribution, providing a smoother optimization landscape.

Beyond stability considerations, random sampling preserves the natural class distribution (uniform for CIFAR-10), ensuring that batch normalization statistics and gradient expectations remain unbiased. In contrast, uncertainty sampling creates a biased distribution concentrated near decision boundaries. For a 3-way classification between "airplane," "automobile," and "bird," Random might select $\{33\%, 33\%, 34\%\}$ while Uncertainty might select $\{10\%, 15\%, 75\%\}$ if "bird" examples are consistently ambiguous. This distributional shift can cause batch normalization to learn inappropriate scale and shift parameters, degrading generalization performance.

6.2 The Fragility of Uncertainty Sampling

Our stochastic run reveals that Uncertainty Sampling is brittle. When the oracle is noisy, "hard" examples (high entropy) are frequently just mislabeled or ambiguous instances. By prioritizing these samples, Uncertainty Sampling actively "poisons" the training set with noise. This confirms our hypothesis that **input uncertainty neutralizes the information gain of active queries**. If the oracle cannot be trusted, the safest strategy is to sample uniformly to dilute the noise. The damage extends beyond any single round, however, as this "adversarial selection" effect compounds over rounds: as the model becomes increasingly confident about mislabeled examples acquired early, it produces even worse uncertainty estimates in subsequent rounds, leading to cascading error propagation. The model essentially learns to be confidently wrong about systematically mislabeled boundary cases, creating a feedback loop that degrades performance far below the random baseline.

6.3 Simulation as a Stress Test

This project highlights the value of simulation in Operations Research. Had we only looked at a fixed $\epsilon = 0$ case (as many papers do), we might have concluded the strategies are "equal." By simulating a stochastic oracle, we uncovered a critical failure mode of the Uncertainty strategy. This suggests that for real-world applications with imperfect annotators, complex AL pipelines may yield a negative ROI compared to simple random selection.

7 Limitations and Future Work

7.1 Model Architecture and Training Protocol

We utilized a shallow 3-layer CNN to enable thousands of simulation loops. While computationally tractable, this limits the absolute accuracy ($\approx 48\%$). A deeper model (e.g., ResNet-18) might be better capable of exploiting "hard" examples, potentially improving the performance of Uncertainty Sampling. Additionally, training uses a fixed 10-epoch schedule per round rather than convergence-based stopping. This introduces two potential biases: (1) early rounds may overtrain on small datasets ($N = 100$), causing the model to memorize noise, and (2) later rounds ($N = 2,000$) may undertrain as 10 epochs becomes insufficient. A validation-based early stopping criterion would better isolate the effect of data selection from optimization artifacts, but would require withholding labeled data from an already budget-constrained setting.

7.2 Reinitialization vs. Fine-Tuning

Retraining from scratch after each acquisition is computationally honest but may not reflect production usage where practitioners fine-tune existing models. If the model were instead warm-started, uncertainty estimates would be better calibrated, potentially improving uncertainty sampling's relative performance. However, warm-starting introduces path-dependencies that complicate causal attribution of performance to query strategy versus learning rate scheduling.

7.3 Batch Acquisition and Diversity Metrics

Our $k = 100$ batch size represents 5% of the final budget acquired simultaneously. This "greedy batch" approach may select redundant samples within each batch. More sophisticated methods like BatchBALD or facility location objectives could mitigate this, but would increase computational cost by $O(k^2)$. Furthermore, the diversity strategy assumes that Euclidean distance in the penultimate layer embedding space is a meaningful proxy for semantic diversity. This assumption is questionable when the model is weak (early rounds) and the feature space is poorly structured. Alternative diversity metrics (e.g., determinantal point processes) may perform differently.

7.4 Dataset and Noise Model Generalizability

CIFAR-10 is a balanced, well-curated dataset with relatively clear class boundaries. Our findings may not generalize to imbalanced datasets, fine-grained classification tasks (e.g., medical imaging), or domains with systematic annotation biases. The 32×32 resolution also limits the types of "hard" examples present compared to high-resolution imagery. Additionally, our budget stopped at 2,000 labels (4% of CIFAR-10). Future work should explore the "crossover point" where the model becomes strong enough to effectively utilize active queries. The uniform ϵ-greedy noise model is simplistic. A more realistic simulation would use class-dependent confusion matrices (e.g., confusing "cat" with "dog" but not "airplane") that reflect actual annotator confusion patterns.

8 Conclusion

This project developed a rigorous Monte Carlo simulation to evaluate active learning strategies under noisy conditions. Contrary to the prevailing assumption that "smart" sampling is always better, our results demonstrate that **Random Sampling is a formidable baseline**, particularly when the annotator is unreliable.

Statistical analysis of the stochastic noise experiment showed that Uncertainty Sampling significantly underperforms Random Sampling (Cohen's d = 2.67) when noise is present. For practitioners, this implies that investing in complex Active Learning pipelines is likely unwarranted unless data annotation accuracy can be guaranteed. In high-noise or low-budget environments, the "naive" Random strategy offers the highest robustness and Return on Investment.

References

1. Glasserman, P., & Yao, D. D. (1992). **Guidelines for Common Random Numbers.** *Management Science*, 38(6), 884-908.

2. Settles, B. (2009). **Active Learning Literature Survey.** University of Wisconsin-Madison Computer Sciences Technical Report.

3. Gal, Y., Islam, R., & Ghahramani, Z. (2017). **Deep Bayesian Active Learning with Image Data.** *International Conference on Machine Learning (ICML).*

4. Sener, O., & Savarese, S. (2018). **Active Learning for Convolutional Neural Networks: A Core-Set Approach.** *International Conference on Learning Representations (ICLR).*

5. Lewis, D. D., & Gale, W. A. (1994). **A Sequential Algorithm for Training Text Classifiers.** *SIGIR.*

9 Appendix

9.1 Code

For more information, visit the project Github: https://github.com/ecEthanCar/AL_Cifar_PhD.