



Research article

DAL-Chat-SSL: Deep active learning with LLM oracles and embedding-space entropy for multi-modal classification

Necla Kochan*

Department of Mathematics, Faculty of Arts and Sciences, Izmir University of Economics, Izmir, Turkiye

* **Correspondence:** Email: necla.kochan@ieu.edu.tr.

Abstract: Deep active learning reduces annotation costs by querying only the most informative samples, but it depends on human annotators who are often unavailable in data-scarce settings. We propose deep active learning with artificial intelligence chatbots for semi-supervised learning (DAL-Chat-SSL), a hybrid framework that replaces human annotators with large language model (LLM) chatbots as labeling oracles and extends this paradigm—previously confined to text classification—to tabular, time-series, and image modalities. The framework introduces an embedding-space entropy acquisition function based on inverse-distance weighting in the learned representation, designed to address the miscalibration of softmax outputs that limits conventional predictive entropy under small labeled pools. Two chatbots, ChatGPT and Claude, serve as annotation oracles under a documented prompting protocol; their annotation accuracy is measured on held-out probe sets rather than assumed. We integrate active acquisition with semi-supervised self-training and evaluate the framework on six benchmarks (Sonar, Ionosphere, Optdigits, ECG5000, MNIST binary, MNIST 10-class) over 10 independent runs each, using a held-out test split that never participates in training, acquisition, or model selection. Against a fully supervised reference trained on all available labels, the framework recovers 84.8–99.8% of attainable performance while annotating 4.4–37% of the pool with no human involvement; on Ionosphere, 46 chatbot annotations are statistically indistinguishable from full supervision on 208 ground truth labels. Annotation error rates of 14–33% cost between one and seven points of final performance, and per-class analysis confirms that no class is lost to annotation noise. Acquisition strategies are separate from random sampling on exactly one of the six benchmarks, the one on which the initial model is already close to its ceiling; we relate this to the position of the model on its learning curve rather than to the annotation budget alone. The results indicate that LLM oracles, combined with semi-supervised self-training, offer a practical route to label-efficient learning beyond the text domain.

Keywords: active learning; artificial intelligence; entropy; large language models; deep learning; semi-supervised classification

1. Introduction

Due in part to the availability of large-scale annotated datasets, deep learning has drawn increasing attention for its high performance in a variety of applications, including image classification, natural language processing, and medical diagnosis [1]. However, in many real-world problems, obtaining high-quality labeled data remains a significant challenge. Manual annotation is costly and time-consuming, requiring expertise in domains such as medical imaging, autonomous driving, and bioinformatics [2, 3]. For instance, annotating a single urban scene image pixel-by-pixel can take more than 90 minutes, and delineating tumors in three-dimensional (3D) medical volumes demands considerable time and clinical expertise [2]. The same tension is acute in safety- and reliability-critical systems, including medical imaging equipment, where deep models are increasingly deployed for fault diagnosis, degradation modeling, and predicting remaining useful life, yet labeled failure events are intrinsically rare and expensive to obtain [4–6]. Label-efficient learning is therefore not only a computational convenience but a practical prerequisite in such settings.

Active learning (AL) has emerged as one of the main approaches to address these challenges. AL aims to maximize models' performance under a limited labeling budget by iteratively selecting the most informative unlabeled samples for annotation by a human oracle [7]. Classical AL strategies rely on heuristics such as uncertainty sampling, diversity-based selection, and query-by-committee [2, 8]. With the advent of deep learning, deep active learning (DAL) methods have gained considerable attention, combining neural network representations with active selection criteria. Recent studies provide comprehensive taxonomies covering uncertainty-based, distribution-based, hybrid, and automatic query strategies for DAL [9–11]. Notable DAL methods include Bayesian approaches using Monte Carlo dropout [12], batch-mode strategies such as CoreSet [13] and BatchBALD [14], loss-prediction modules [15], and generative adversarial active learning methods such as VAAL [16]. Despite these advances, DAL still faces critical challenges such as noisy oracles, unstable performance in the early stages of training—commonly referred to as the cold-start problem—and difficulty in cross-domain transfer [2]. The cold-start regime is directly relevant to the present study, since chatbot annotation is most attractive precisely when the labeled pool is small.

Semi-supervised learning (SSL) has attracted growing interest as a complementary paradigm for utilizing a small collection of labeled samples in conjunction with an abundance of unlabeled data. Modern deep SSL techniques combine pseudo-labeling and consistency regularization to efficiently exploit unlabeled data. Methods such as FixMatch [17], FlexMatch [18], and MixMatch [19] have demonstrated that SSL can approach fully supervised performance with dramatically fewer labels. A comprehensive survey by Yang et al. [20] categorizes deep SSL methods into generative, consistency regularization, graph-based, pseudo-labeling, and hybrid approaches. An empirical evaluation of 16 deep SSL algorithms across image, text, and sound datasets provided further practical recommendations for selecting the most suitable technique, given the data characteristics and labels' availability [21]. More recently, Wen et al. [22] showed that when labels are extremely scarce, weight initialization alone is insufficient to transfer self-supervised knowledge effectively, as semi-supervised

training can quickly disrupt pre-trained feature representations.

A complementary line of work dispenses with the human annotator altogether by generating labels from a probabilistic model. Zhao et al. [6] proposed a two-stage framework in which an offline multi-condition Bayesian Gaussian-process change-point detector infers the posterior distribution of the degradation-onset time, and the resulting maximum a posteriori estimates are then used as ground-truth labels to train an online bidirectional long short-term memory (LSTM) with conditional attention. The architecture of that framework—an expensive, uncertainty-aware label generator feeding a cheap discriminative learner—is structurally the same problem we address, with one substitution. We replace the probabilistic label generator with a large language model (LLM) queried in an active learning loop. Their work also demonstrates that automatically generated labels can support a downstream model that generalizes to previously unseen units, which is the property our framework must likewise establish.

The integration of AL with SSL constitutes a promising research direction, as both paradigms seek to achieve strong performance with minimal supervision [11, 23]. Semi-supervised methods can improve DAL by providing structural information about the feature distribution, while DAL can boost SSL by propagating more informative labels from carefully selected samples to unlabeled data [23]. A number of studies have investigated this combination: CEAL [24] assigns pseudo-labels to high-confidence samples while querying annotations for the most uncertain ones; Song et al. [25] combined MixMatch with active learning and achieved notable improvements on CIFAR-10/100 and SVHN (Street View House Numbers); and Guo et al. [26] proposed adversarial examples with graph-based virtual labels for semi-supervised active learning. A recent study by Mittal et al. [27] revealed that AL methods improve by a large margin when integrated with data augmentation and SSL, but barely outperform random baselines without these additions, underscoring the synergistic nature of combining AL with SSL [11]. Our own results, reported in Section 5, are consistent with this observation: Differences between acquisition strategies remain small relative to run-to-run variability at small annotation budgets, whereas the contribution of the semi-supervised component is positive on all six benchmarks and, on two of them, statistically significant.

Recent studies have begun to explore LLMs as active contributors in the learning process. Xia et al. [28] provided a comprehensive survey covering both LLM-based selection and generation paradigms. Building on this line of work, Bayer et al. [29] introduced ActiveLLM, which leverages instruction-tuned LLMs such as GPT-4 and Llama 3 for instance selection and addresses the cold-start problem in few-shot text classification. Qi et al. [30] proposed MoLLIA, a mixture of LLMs in the loop framework that uses multiple LLMs as annotators within an AL loop. Taken together, these studies demonstrate that LLMs can meaningfully participate in the AL process, functioning both as query selectors and as sources of annotation. What remains largely unexamined is whether this paradigm transfers beyond text, where the instance to be annotated is not natively a sequence of tokens and must first be rendered into a form that a language model can read.

In this study, we propose DAL-Chat-SSL (deep active learning with artificial intelligence chatbots for semi-supervised learning), an approach that extends the use of chatbot oracles across various domains. The proposed method integrates three components: (i) A DAL module with a feed-forward encoder applied uniformly to tabular, time-series, and image inputs, using entropy-based uncertainty sampling; (ii) an artificial intelligence (AI) chatbot oracle module in which two LLMs (ChatGPT and Claude) annotate the selected samples under a documented prompting protocol, with the annotation accuracy measured on held-out probe sets rather than assumed, enabling a controlled analysis of the

oracle noise; and (iii) a semi-supervised self-training module that propagates the chatbot-provided labels to the unlabeled dataset via pseudo-labeling. We compare two uncertainty strategies: Predictive entropy based on the model's output and a novel embedding-space inverse-distance entropy calculated in the learned representation. The framework is evaluated on six benchmarks over 10 independent runs each, using a held-out test split that never participates in training, acquisition, or model selection, and is benchmarked against a fully supervised reference so that its performance is reported as a fraction of the accuracy achievable with complete human annotation.

2. Background

This section establishes the technical foundations of the proposed approach. We describe the learning setting, the neural architectures used for feature extraction and classification, and the two uncertainty quantification strategies that form the basis of our ablation study.

2.1. Semi-supervised learning setting

We consider a dataset of n observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with the inputs $x_i \in \mathbb{R}^d$ and the class labels $y_i \in \{1, \dots, K\}$. Only a small fraction of the labels is initially available. We partition the index set into a labeled pool $\mathcal{L}$ with n_L observations and an unlabeled pool $\mathcal{U}$ with n_U observations, where $n_L \ll n_U$. A deep neural network f_θ , parameterized by weights θ , maps each input to a probability vector over K classes.

Training proceeds in a semi-supervised fashion. The model minimizes a composite objective

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{lab}}(\theta) + \lambda \mathcal{L}_{\text{unlab}}(\theta), \quad (2.1)$$

where $\mathcal{L}_{\text{lab}}$ is the cross-entropy loss computed on $\mathcal{L}$, and $\mathcal{L}_{\text{unlab}}$ is a pseudo-label loss computed on the subset of $\mathcal{U}$ for which the model's maximum predicted probability exceeds a confidence threshold τ . We set $\lambda = 0.5$, so that pseudo-labeled instances contribute at half the weight of annotated ones and an incorrect pseudo-label cannot dominate the objective. We adopt an iterative self-training strategy, described in Section 3.2, which runs for at most R iterations per active learning round.

2.2. Deep learning architectures

The architecture is treated as an experimental factor rather than as a fixed design choice, and the same two encoders are applied to every dataset and every modality. Both share a two-stage design: An encoder producing a d_e -dimensional embedding $\mathbf{e}(x)$, followed by a linear classification head. The embedding is the key quantity for the representation-based uncertainty mode of Section 2.3.

2.2.1. Multilayer perceptron

The multilayer perceptron treats the input as an unordered feature vector. It flattens the input, applies two fully connected layers with ReLU (Rectified Linear Unit) activations and dropout, and produces the embedding $\mathbf{e}(x) = \text{dropout}(\phi(x)) \in \mathbb{R}^{d_e}$, where ϕ denotes the two-layer body. The logits are $f_\theta(x) = W_o \mathbf{e}(x) + b_o$ with $W_o \in \mathbb{R}^{K \times d_e}$. This encoder makes no assumption about the ordering of the coordinates and serves as the reported model throughout Section 5.

2.2.2. Gated recurrent unit

The gated recurrent unit (GRU) treats the input as an ordered sequence and is included as a comparison encoder, since several of the benchmarks have features that are ordered by construction: The 60 Sonar attributes are energies integrated over contiguous frequency bands indexed by increasing frequency, the 33 retained Ionosphere attributes are the real and imaginary components of 17 radar pulses received in succession, and the 140 ECG5000 attributes are consecutive samples of a single heartbeat. Whether this ordering is worth exploiting is an empirical question, which Section 5.1 answers.

We use a single-layer GRU [31]. A d -dimensional observation is read as a univariate sequence of length $T = d$; an image of shape $C \times H \times W$ is read row-wise as a sequence of length H with $C \cdot W$ features per step. The hidden state is updated by

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \quad (2.2)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \quad (2.3)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h), \quad (2.4)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \quad (2.5)$$

where σ is the sigmoid function and $\odot$ denotes element-wise multiplication. The terminal hidden state $h_T \in \mathbb{R}^{d_h}$ summarizes the sequence and serves as the embedding, $\mathbf{e}(x) = \text{dropout}(h_T)$.

2.2.3. Convolutional encoder

For the image benchmarks, we additionally report a convolutional encoder, which is the architecture conventional for that modality. Two blocks of convolution, batch normalization, ReLU, and max-pooling are followed by a fully connected layer, and the embedding is again the penultimate activation, so the acquisition functions of Section 2.3 operate on it exactly as they do for the other encoders..

2.3. Uncertainty quantification

At each AL round the method must rank all unlabeled instances by informativeness. We use Shannon entropy [32] as the ranking criterion throughout, but investigate two distinct ways of constructing the class probability distribution from which entropy is computed.

2.3.1. Predictive entropy

The first approach uses the model's own softmax output as the probability estimate,

$$\hat{p}_k(x_u) = \frac{\exp(f_{\theta,k}(x_u))}{\sum_{c=1}^K \exp(f_{\theta,c}(x_u))}, \quad (2.6)$$

yielding the predictive entropy

$$H_{\text{pred}}(x_u) = - \sum_{k=1}^K \hat{p}_k(x_u) \log_2 \hat{p}_k(x_u). \quad (2.7)$$

This is the standard uncertainty measure in pool-based active learning [7]. Its reliability depends on the calibration of the softmax layer, which can be poor when the model has been trained on very few labeled examples. We note that for $K = 2$, predictive entropy and least confidence are strictly monotone functions of $|\hat{p} - \frac{1}{2}|$ and therefore induce the same ranking; on the binary benchmarks, they are consequently a single strategy.

2.3.2. Embedding-space entropy

As an alternative, we construct a non-parametric class distribution in the learned embedding space. For an unlabeled sample x_u , we compute the inverse Euclidean distance between its embedding $\mathbf{e}(x_u)$ and the embedding of each labeled sample $\mathbf{e}(x_i)$, $i \in \mathcal{L}$ as follows:

$$w_i(x_u) = (\|\mathbf{e}(x_u) - \mathbf{e}(x_i)\|_2 + \epsilon)^{-1}, \quad (2.8)$$

with $\epsilon = 10^{-10}$ for numerical stability. Aggregating these weights within each class yields a proximity-based class distribution

$$q_k(x_u) = \frac{\sum_{i \in \mathcal{L}: y_i=k} w_i(x_u)}{\sum_{c=1}^K \sum_{i \in \mathcal{L}: y_i=c} w_i(x_u)}, \quad (2.9)$$

from which the embedding-space entropy is

$$H_{\text{emb}}(x_u) = - \sum_{k=1}^K q_k(x_u) \log_2 q_k(x_u). \quad (2.10)$$

This measure is computed in the learned representation rather than from the model's output probabilities. Because the embedding is shaped by the classification objective during training, it is expected to capture task-relevant structure that is not evident in the raw feature space, so local distances to labeled samples may provide a more informative estimate of uncertainty than the softmax layer when the labeled pool is small. Note that q_k is defined only relative to the annotated anchors; when $\mathcal{L}$ is very small, the estimate can degenerate, a failure mode we observe and report in Section 6.5.

3. Proposed method

Building on the components introduced in Section 2, we now describe how they are assembled into a complete DAL pipeline with chatbot oracles and semi-supervised self-training.

3.1. Pipeline description

The method operates for R_{AL} acquisition rounds. Each round consists of four steps.

1. **Model update.** The classifier f_θ is retrained from scratch on the current labeled pool $\mathcal{L}$ using the pseudo-label self-training procedure of Section 3.2, with early stopping on a validation split that is disjoint from both the pool and the test set.
2. **Uncertainty scoring.** For every remaining unlabeled instance $x_u \in \mathcal{U}$, an entropy score is computed under one of the modes of Section 2.3,

$$H(x_u) \in \{H_{\text{pred}}(x_u), H_{\text{emb}}(x_u)\}. \quad (3.1)$$

3. **Query selection.** The b highest-scoring instances are selected as a batch,

$$Q = \arg \text{top-}b_{x_u \in \mathcal{U}} H(x_u), \quad (3.2)$$

with ties broken by ascending index so that the runs are exactly reproducible. Batch acquisition is used rather than single-instance acquisition because retraining the network after every individual label would be computationally prohibitive at the number of repeated runs reported here.

4. **Chatbot annotation.** Each selected instance is serialized into a text prompt and submitted to an LLM-based oracle, which returns a class label $\hat{y}$. The returned label may be incorrect; no correction is applied. The pools are updated as $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, \hat{y}) : x \in Q\}$ and $\mathcal{U} \leftarrow \mathcal{U} \setminus Q$.

The total annotation budget is $B = b \cdot R_{\text{AL}}$. The model is retrained and evaluated once more after the final acquisition round, so a run yields $R_{\text{AL}} + 1$ measurements along the learning curve.

3.2. Self-training loop

Within each acquisition round, the model is retrained from scratch by iterative pseudo-labeling. Let $\mathcal{L}_0$ denote the annotated set, that is, the initial seed set together with every instance labeled by the oracle so far. At iteration r , the model is trained on the current training set, then predicts class probabilities for all pool instances that are not annotated. The pseudo-labeled set is

$$\mathcal{P}^{(r)} = \{x_j \in \mathcal{U} : \max_k \hat{p}_k(x_j) \geq \tau\}, \quad (3.3)$$

with $\hat{y}_j = \arg \max_k \hat{p}_k(x_j)$, and the training set for the next iteration is

$$\mathcal{L}^{(r+1)} = \mathcal{L}_0 \cup \{(x_j, \hat{y}_j) : x_j \in \mathcal{P}^{(r)}\}. \quad (3.4)$$

Two details matter. First, the pseudo-labeled set is recomputed from $\mathcal{L}_0$ at every iteration rather than accumulated, so that an incorrect pseudo-label assigned early cannot persist once the model's belief about that instance changes. Second, pseudo-labeled instances enter the loss with weight $\lambda = 0.5$. Only instances from the training pool are ever pseudo-labeled; the validation and test splits are excluded, so their features do not enter the self-training objective. The loop terminates early if no instance clears the threshold.

3.3. Chatbot oracle

We replace the human annotator with an LLM-based chatbot. Each queried instance is serialized into a structured text prompt containing the task description, the class definitions, a statement of the input encoding, few-shot exemplars drawn exclusively from the already annotated pool, and the query instance itself; the oracle answers with a single JSON (JavaScript Object Notation) object that is parsed automatically. The serialization is modality-specific and is specified in full, together with a rendered example, in Section 4.3. Two oracles are evaluated independently, ChatGPT and Claude, and their annotation accuracy is measured on a stratified probe set drawn from the training pool rather than assumed.

3.4. Model configurations

All models are trained from scratch at every acquisition round with no pre-trained weights. Experiments were implemented in Python using PyTorch and executed on Google Colab with an NVIDIA T4 GPU (Graphics Processing Unit). All random number generator states (Python, NumPy, PyTorch, CUDA) are fixed per run and the data split is re-drawn from the seed, so that a run is reproducible end to end; deterministic kernels are requested via `torch.use_deterministic_algorithms`.

The multilayer perceptron uses a hidden width of $d_h = 64$ on the tabular and time-series benchmarks and $d_h = 128$ on the image benchmarks, with dropout of 0.3. The GRU uses the same hidden widths and dropout. Optimization uses Adam [33] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 10^{-3} , and weight decay of 10^{-4} , with gradient-norm clipping at 5.0 and cross-entropy loss. Batch sizes are 16 for the Sonar and Ionosphere datasets, 32 for ECG5000, and 64 for the image benchmarks. The maximum number of epochs per training call is 60 for the tabular benchmarks, 30 for ECG5000 and 10–12 for the image benchmarks, with early stopping on validation macro- F_1 and the best checkpoint restored before every evaluation. The pseudo-label confidence threshold is $\tau = 0.75$ and the number of self-training iterations is $R = 2$ on the tabular and time-series benchmarks and $R = 1$ on the image benchmarks. The annotation protocol, including the initial seed size, the acquisition batch size b and the budget B for each dataset, is given in Table 2.

3.5. Experimental design

The experiment is a full factorial design across three factors: The acquisition strategy, the encoder, and the random seed. The strategies are random sampling as a budget-matched control, predictive entropy, MC-dropout (Monte Carlo dropout) mutual information, and the proposed embedding-space entropy; the encoders are the multilayer perceptron and the GRU; and each combination is repeated over 10 seeds. Because the split is redrawn for every seed, the reported variability includes split variability and not only the initialization noise.

Within a seed, every condition shares the same stratified split, the same initial labeled set, the same annotation budget, and the same oracle. The oracle is a deterministic function of the instance rather than of the query order, so a given instance receives the same label regardless of which strategy selected it; all comparisons are therefore exactly paired. Conditions differ only in the factor under study.

Three reference conditions accompany the design. The first is the majority-class predictor, evaluated on the same test splits, which marks the floor below which a configuration has learned nothing. The second is the initial seed set alone, trained with self-training but without any acquisition, which isolates the contribution of active annotation. The third is a fully supervised model given ground truth labels for the entire training pool under an otherwise identical configuration, which provides the ceiling against which the label efficiency is expressed. Two further ablations are reported: The pipeline with pseudo-labeling disabled, and a noiseless oracle in place of the chatbot.

3.6. Algorithm

Algorithm 1 describes the inner pseudo-label self-training procedure of Section 3.2, and Algorithm 2 is for the outer acquisition loop of Section 3.1.

Algorithm 1: Pseudo-label self-training

Data: Dataset $\mathcal{X}$, label vector $\mathbf{y}_{\text{obs}}$ (-1 for unlabeled), threshold τ , max rounds R , epochs E , pseudo-label weight λ

Result: Trained deep model f_θ

```

1 Initialize  $f_\theta$  with random weights;
2  $\mathcal{L}_0 \leftarrow \{(x_i, y_i) : y_{\text{obs},i} \neq -1\}$ ;  $\mathcal{L}_{\text{train}} \leftarrow \mathcal{L}_0$ ;
3  $\mathbf{y}_{\text{work}} \leftarrow \mathbf{y}_{\text{obs}}$ ;
4 for  $r = 1, \dots, R$  do
5   Train  $f_\theta$  on  $\mathcal{L}_{\text{train}}$  for  $E$  epochs, pseudo-labels weighted by  $\lambda$ , early-stopped on the validation split;
6    $\mathcal{I}_u \leftarrow \{j : y_{\text{work},j} = -1\}$ ;
7   if  $\mathcal{I}_u = \emptyset$  then
8     break;
9   end
10  Compute  $\hat{\mathbf{p}}_j \leftarrow f_\theta(x_j)$  for all  $j \in \mathcal{I}_u$ ;
11   $\text{conf}_j \leftarrow \max_k \hat{p}_{j,k}$ ;  $\hat{y}_j \leftarrow \arg \max_k \hat{p}_{j,k}$ ;
12   $\mathcal{P} \leftarrow \{j \in \mathcal{I}_u : \text{conf}_j \geq \tau\}$ ;
13  if  $\mathcal{P} = \emptyset$  then
14    break;
15  end
16   $\mathcal{L}_{\text{train}} \leftarrow \mathcal{L}_0 \cup \{(x_j, \hat{y}_j) : j \in \mathcal{P}\}$ ; // rebuilt, not accumulated
17 end
18 Train  $f_\theta$  on final  $\mathcal{L}_{\text{train}}$  for  $E$  epochs;
19 return  $f_\theta$ ;
```

Algorithm 2: DAL-Chat-SSL: Deep active learning with chatbot oracles

Data: Dataset $\mathcal{X}$; initial labeled indices $\mathcal{L}$ with ground-truth labels; unlabeled indices $\mathcal{U}$; rounds R_{AL} ; batch size b ; mode $m \in \{\text{pred}, \text{emb}\}$; chatbot oracle O ; self-training parameters τ, R

Result: Annotated set $\mathcal{L}$, trained model f_θ

```

1 Set  $y_{\text{obs},i} \leftarrow y_i$  for  $i \in \mathcal{L}$ ; set  $y_{\text{obs},j} \leftarrow -1$  for  $j \in \mathcal{U}$ ;
2 for  $t = 1, \dots, R_{\text{AL}}$  do
3    $f_\theta \leftarrow \text{SELFTRAIN}(\mathcal{X}, \mathbf{y}_{\text{obs}}, \tau, R)$ ; // Alg. 1
4   if  $m = \text{emb}$  then
5      $\{\mathbf{e}_i\}_{i \in \mathcal{L}} \leftarrow$  embeddings of annotated instances from  $f_\theta$ ;
6   end
7   for each  $u \in \mathcal{U}$  do
8     if  $m = \text{pred}$  then
9        $H_u \leftarrow H_{\text{pred}}(x_u)$ ; // Equation (2.7)
10    else
11       $\mathbf{e}_u \leftarrow$  embedding of  $x_u$  from  $f_\theta$ ;
12       $H_u \leftarrow H_{\text{emb}}(x_u)$ ; // Equations (2.8)--(2.10)
13    end
14  end
15   $Q \leftarrow$  indices of the  $b$  largest  $H_u, u \in \mathcal{U}$ ; // Equation (3.2); ties by index
16  for each  $q \in Q$  do
17     $y_{\text{obs},q} \leftarrow O(x_q)$ ; // QUERYCHATBOT, no correction
18  end
19   $\mathcal{L} \leftarrow \mathcal{L} \cup Q$ ;  $\mathcal{U} \leftarrow \mathcal{U} \setminus Q$ ;
20 end
21  $f_\theta \leftarrow \text{SELFTRAIN}(\mathcal{X}, \mathbf{y}_{\text{obs}}, \tau, R)$ ; // Final retraining
22 return  $f_\theta, \mathcal{L}$ ;
```

4. Experimental setup

4.1. Datasets

We evaluate the proposed framework on six benchmark datasets spanning three data modalities: Tabular (Sonar, Ionosphere, Optdigits), time series (ECG5000) and images (MNIST). Optdigits was added during the revision to separate two possible explanations for the weak acquisition results on MNIST: A large number of classes, or a high-dimensional raw-pixel input. It has 10 classes in 62 dimensions, so it has the first without the second. Table 1 reports the datasets exactly as loaded by our experimental code, including the stratified sub-sampling applied to the image and time-series collections to keep the full factorial design computationally tractable.

Table 1. Datasets used in the experiments, where n is the number of instances after stratified sub-sampling, d is the input dimensionality, and K is the number of classes. The last column gives the sizes of the AL pool, the validation split and the held-out test split.

Dataset	Modality	n	d	Class distribution	Pool/Val/Test
Sonar	Tabular	208	60	97/111	123/22/63
Ionosphere	Tabular	351	33	126/225	208/37/106
Optdigits	Tabular	5620	62	554/571/557/572/568/558/558/566/554/562	3343/591/1686
ECG5000	Time series	3000	140	1751/1060/58/117/14	1785/315/900
MNIST (3/8)	Image	2700	28×28	1381/1319	1606/284/810
MNIST (10)	Image	6000	28×28	592/675/599/612/585/541/589/625/585/597	3570/630/1800

Tabular and time-series features are standardized to a zero mean and unit variance; images are scaled to $[0, 1]$. One constant column of the Ionosphere data is removed prior to standardisation. For ECG5000, the official training and test partitions are pooled and re-split by our own protocol so that the held-out split is stratified and identical across all compared methods.

4.2. Evaluation protocol

Each dataset is split, independently for every random seed, into three disjoint parts: a held-out test split (30% of all instances), a validation split (15% of the remainder), and the training pool. The training pool supplies the initial labeled set, the acquisition candidates, and the unlabeled data used for pseudo-labeling. The validation split is used exclusively for early stopping and checkpoint selection. The test split is read only to report the numbers below; it never participates in training, pseudo-labeling, acquisition, or model selection, and disjointness is asserted programmatically at the start of every run.

Every configuration is repeated over 10 independent runs. We report the mean $\pm$ standard deviation of each metric. Comparisons against the random sampling control are paired by seed and assessed with a two-sided paired t -test, corroborated by the Wilcoxon signed-rank test, with Holm–Bonferroni correction across the family of acquisition strategies; effect sizes are reported as Cohen’s d for paired samples. With $n = 10$, the Wilcoxon floor is $p = 0.002$, low enough for the distribution-free test to corroborate the parametric one.

Table 2 summarizes the annotation protocol, which is identical for every acquisition strategy, encoder and oracle on a given dataset.

Table 2. Annotation protocol. The budget is identical across all compared methods on a given dataset.

Dataset	Init/class	Init total	Batch	Rounds	Budget	Final $ L $
Sonar	8	16	5	6	30	46
Ionosphere	8	16	5	6	30	46
Optdigits	5	50	20	6	120	170
ECG5000	4	20	10	6	60	80
MNIST (3/8)	5	10	10	6	60	70
MNIST (10)	5	50	20	6	120	170

Metrics are defined uniformly for binary and multi-class problems: Sensitivity and specificity are macro-averaged over classes, so that for $K = 2$, they reduce to the textbook binary definitions. On ECG5000, which is strongly imbalanced, macro-F1 is used as the primary metric for significance testing; accuracy is reported alongside it but is uninformative there (see Section 5.4).

4.3. LLM annotation protocol

Queried instances are serialized into text and submitted to the chatbot oracle under a fixed prompt template consisting of (i) a task description, (ii) the class definitions, (iii) a statement of the input encoding, (iv) few-shot exemplars drawn exclusively from the already annotated pool, and (v) the query instance. Tabular instances are rendered as `name=value` pairs; and the time-series instances as a uniformly decimated sequence of amplitudes; images are average-pooled to a 16×16 grid, quantized to 10 intensity levels and rendered twice, once as an ASCII intensity map and once as the corresponding digit matrix, so that the same array is available to the model in both a pictorial and a numeric form. Fully rendered example prompts for every dataset are provided in Appendix A.3.

Oracle accuracy is *measured*, not assumed. For each dataset, a stratified probe set is drawn from the training pool only, annotated under exactly the prompt above, and compared against the ground truth; Table 3 reports the resulting accuracies with Wilson 95% confidence intervals. These measured rates calibrate the annotation simulator used in the main experiments, which places errors on the most ambiguous instances and returns the most confusable alternative class rather than a uniformly random one.

Table 3. Measured oracle accuracy on a stratified probe set drawn from the training pool, with Wilson 95% confidence intervals.

Dataset	Probe n	ChatGPT	Claude
Sonar	123	0.911 [0.847, 0.949]	0.764 [0.682, 0.831]
Ionosphere	208	0.909 [0.862, 0.941]	0.803 [0.744, 0.851]
Optdigits	400	0.848 [0.809, 0.879]	0.755 [0.711, 0.795]
ECG5000	272	0.706 [0.649, 0.757]	0.610 [0.551, 0.666]
MNIST (3/8)	400	0.890 [0.856, 0.917]	0.833 [0.793, 0.866]
MNIST (10)	400	0.762 [0.718, 0.802]	0.738 [0.692, 0.778]

A methodological observation follows from these measurements. The annotation accuracy *realised inside the active-learning loop* is consistently lower than the probe accuracy: On MNIST (10 classes), the oracle scores 0.762 on a random probe but 0.746 on the instances actually queried, and on Ionosphere, 0.909 versus 0.863. Informative-sample selection and chatbot annotation interact adversely, because the instances an acquisition function considers to be most uncertain are also the instances the language model is most likely to mislabel. To our knowledge, this interaction has not been quantified in previous work on LLM-assisted annotation.

4.4. Evaluation metrics

All metrics are computed on the held-out test split described in Section 4 and are averaged over the 10 independent runs. Let $\mathbf{C} \in \mathbb{N}^{K \times K}$ denote the confusion matrix over K classes, with C_{ij} being the number of instances of true class i predicted as class j , and let $N = \sum_{i,j} C_{ij}$. For class k , we write $\text{TP}_k = C_{kk}$, $\text{FN}_k = \sum_{j \neq k} C_{kj}$, $\text{FP}_k = \sum_{i \neq k} C_{ik}$, and $\text{TN}_k = N - \text{TP}_k - \text{FN}_k - \text{FP}_k$.

Aggregate classification metrics. Accuracy is the proportion of correctly classified instances,

$$\text{Acc} = \frac{1}{N} \sum_{k=1}^K \text{TP}_k. \quad (4.1)$$

Because accuracy is dominated by the frequent classes on imbalanced data, we also report the macro-averaged F_1 score,

$$F1_{\text{macro}} = \frac{1}{K} \sum_{k=1}^K \frac{2 P_k R_k}{P_k + R_k}, \quad P_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}, \quad R_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}, \quad (4.2)$$

which weights every class equally regardless of its support. Cohen's κ is reported as a chance-corrected measure of agreement.

Sensitivity and specificity. For the binary datasets (Sonar, Ionosphere, MNIST 3 vs. 8), sensitivity and specificity are defined with respect to the positive class p in the conventional way,

$$\text{Sen} = \frac{\text{TP}_p}{\text{TP}_p + \text{FN}_p}, \quad \text{Spe} = \frac{\text{TN}_p}{\text{TN}_p + \text{FP}_p}. \quad (4.3)$$

For the multi-class datasets (ECG5000, MNIST with 10 classes), no single class is privileged, and the two quantities are macro-averaged over classes, $\text{Sen}_{\text{macro}} = \frac{1}{K} \sum_k R_k$ and $\text{Spe}_{\text{macro}} = \frac{1}{K} \sum_k \text{TN}_k / (\text{TN}_k + \text{FP}_k)$. We note that macro-averaging is not appropriate in the binary case, since for $K = 2$, the true negatives of one class are the true positives of the other and the two macro quantities collapse onto a single value, which also coincides with balanced accuracy. Balanced accuracy is therefore reported only for the multi-class datasets, where it is equal to $\text{Sen}_{\text{macro}}$ and is stated once.

Per-class recall. On the multi-class datasets, we additionally report recall for each class separately, together with the class support. This is necessary because an aggregate score can remain high while a rare class is never predicted: On ECG5000, the model reaches 0.90 accuracy while the macro- F_1 is

0.51, and only the per-class breakdown reveals which classes are responsible. The same breakdown is used to verify that robustness to annotation noise holds uniformly across the label space rather than only on the majority class.

Clustering-based metrics. Cluster purity and normalized mutual information (NMI) are retained from the original formulation,

$$\text{Purity} = \frac{1}{N} \sum_{j=1}^K \max_i C_{ij}, \quad \text{NMI} = \frac{2I(Y; \hat{Y})}{H(Y) + H(\hat{Y})}, \quad (4.4)$$

where $I(\cdot; \cdot)$ is mutual information and $H(\cdot)$ is entropy. They provide an information-theoretic view of how well the predicted partition preserves the true class structure, which is of interest here because a large fraction of the training labels originate from pseudo-labeling rather than from the oracle. We note, however, that both quantities are computed from hard predicted labels and are monotonically related to accuracy for a well-behaved classifier; the conclusions of this study rest on accuracy and macro- F_1 ; purity and NMI are reported for continuity rather than as primary evidence.

Primary metric and reference points. Significance testing is performed on a single primary metric per dataset, declared in advance: Macro- F_1 for ECG5000, whose class distribution is strongly imbalanced, and accuracy for the remaining four benchmarks. Two reference points accompany every table. The first is the majority-class predictor, which always outputs the most frequent class of the training pool and is evaluated on the same test splits; a configuration that does not clear this line has learned nothing, and its acquisition results are not interpretable. The second is a fully supervised model trained on the entire training pool with ground truth labels under an otherwise identical configuration, which provides the ceiling against which label efficiency is expressed.

Learning-curve summary. Because a single end-of-budget score discards the trajectory, we also report the area under the learning curve (ALC),

$$\text{ALC} = \frac{1}{R} \sum_{r=0}^{R-1} \frac{m_r + m_{r+1}}{2}, \quad (4.5)$$

where m_r is the primary metric after round r , and R is the number of acquisition rounds. A method that reaches a given score earlier in the budget is preferable to one that reaches it only at the end, and ALC captures that distinction.

Failure counting. Alongside the mean and standard deviation, we report the number of runs, out of 10, whose final score falls at or below the majority-class reference. A configuration can have an acceptable mean while failing outright in a minority of runs, and this count makes such behaviour visible where the standard deviation alone would understate it.

5. Results

5.1. Architecture

Tabular and time-series results use a multilayer perceptron encoder. The choice was made empirically: We compared the recurrent encoder used in the original formulation against an order-agnostic multilayer perceptron under an identical protocol (the same splits, seeds, annotation budget, oracle, and acquisition strategies) on the Sonar, Ionosphere, ECG5000 and both MNIST datasets. The multilayer perceptron was superior on each of them and under every acquisition strategy, including on ECG5000, the one dataset whose features are genuinely ordered in time. Under the recurrent encoder, between 4 and 10 of 10 runs terminate at or below the majority-class floor. The full comparison is reported in Appendix A.1 (Tables A1 and A2).

5.2. Label efficiency

Table 4 compares four reference points measured on identical splits: The majority-class predictor, the model trained on the initial seed set alone, the model after LLM-assisted active learning, and a fully supervised model given ground truth labels for the entire training pool.

With no human annotator at any stage, the framework recovers between 84.8% and 99.8% of fully supervised performance. The most favorable case is Ionosphere, where 46 chatbot annotations are statistically indistinguishable from full supervision on 208 ground truth labels (0.8991 ± 0.0298 versus 0.9009 ± 0.0432). On the three largest pools, the labeling cost is 4.5% (ECG5000), 4.8% (MNIST, 10 classes); and 5.1% (Optdigits) of the available instances. Sonar is the least favorable case: Its pool contains only 123 instances, so the same protocol consumes 37% of it while recovering 91% of the ceiling.

The fully supervised ceiling should be read as the score of the same architecture trained on the same pool with ground-truth labels, not as the state-of-the-art performance on these benchmarks.

Table 4. Label efficiency (10 runs, mean $\pm$ standard deviation (SD)). “% of ceiling” is the ratio to the fully supervised score in the primary metric (macro- F_1 for ECG5000; accuracy elsewhere). Image rows use the convolutional encoder; the remainder use the multilayer perceptron.

Dataset	Setting	#Labels	Primary metric	% of ceiling
Sonar	Majority class	0	0.5397	–
	Initial seed set	16	0.6762 ± 0.0782	83.9
	+ LLM active learning	46	0.7349 ± 0.0778	91.1
	Fully supervised	123	0.8063 ± 0.0395	100.0
Ionosphere	Majority class	0	0.6415	–
	Initial seed set	16	0.7651 ± 0.0801	84.9
	+ LLM active learning	46	0.8991 ± 0.0298	99.8
	Fully supervised	208	0.9009 ± 0.0432	100.0
Optdigits	Majority class	0	0.1014	–
	Initial seed set	50	0.8521 ± 0.0261	86.8
	+ LLM active learning	170	0.9167 ± 0.0169	93.4
	Fully supervised	3343	0.9820 ± 0.0022	100.0
ECG5000	Majority class	0	–	–
	Initial seed set	20	0.4590 ± 0.0669	78.2
	+ LLM active learning	80	0.5088 ± 0.0420	86.7
	Fully supervised	1785	0.5871 ± 0.0399	100.0
MNIST (3/8)	Majority class	0	0.5111	–
	Initial seed set	10	0.8506 ± 0.0396	85.7
	+ LLM active learning	70	0.9106 ± 0.0774	91.7
	Fully supervised	1606	0.9931 ± 0.0035	100.0
MNIST (10)	Majority class	0	0.1124	–
	Initial seed set	50	0.5842 ± 0.1143	60.1
	+ LLM active learning	170	0.8246 ± 0.0221	84.8
	Fully supervised	3570	0.9719 ± 0.0068	100.0

5.3. Acquisition strategy comparison

Table 5 compares the acquisition rules under an identical annotation budget.

Table 5. Final test performance under an identical annotation budget (10 runs, mean $\pm$ SD). Δ and the tests are computed on the primary metric against random sampling, paired by seed. Holm–Bonferroni correction is applied across the comparisons within each dataset; * marks differences that survive it. Image rows use the convolutional encoder. BALD represents Bayesian active learning by disagreement.

Dataset	Strategy	Primary metric	Δ	p	d
Sonar	Random sampling	0.7413 $\pm$ 0.0613	–	–	–
	Predictive entropy	0.7333 $\pm$ 0.0579	–0.0079	0.751	–0.10
	MC-dropout (BALD)	0.7778 $\pm$ 0.0725	+0.0365	0.057	+0.69
	Proposed (embedding)	0.7349 $\pm$ 0.0778	–0.0063	0.818	–0.07
Ionosphere	Random sampling	0.8519 $\pm$ 0.0561	–	–	–
	Predictive entropy	0.8406 $\pm$ 0.0653	–0.0113	0.370	–0.30
	MC-dropout (BALD)	0.8623 $\pm$ 0.0486	+0.0104	0.592	+0.18
	Proposed (embedding)	0.8991 $\pm$ 0.0298	+0.0472	0.039	+0.76
Optdigits	Random sampling	0.8999 $\pm$ 0.0179	–	–	–
	Predictive entropy	0.9353 $\pm$ 0.0095	+0.0354	0.001*	+1.60
	MC-dropout (BALD)	0.9410 $\pm$ 0.0084	+0.0410	0.000*	+2.09
	Proposed (embedding)	0.9167 $\pm$ 0.0169	+0.0168	0.031*	+0.81
ECG5000	Random sampling	0.4863 $\pm$ 0.0658	–	–	–
	Predictive entropy	0.4969 $\pm$ 0.0879	+0.0106	0.482	+0.23
	MC-dropout (BALD)	0.4495 $\pm$ 0.0770	–0.0367	0.119	–0.54
	Proposed (embedding)	0.5088 $\pm$ 0.0420	+0.0225	0.273	+0.37
MNIST (3/8)	Random sampling	0.9117 $\pm$ 0.0280	–	–	–
	Predictive entropy	0.9167 $\pm$ 0.0466	+0.0049	0.771	+0.09
	MC-dropout (BALD)	0.9196 $\pm$ 0.0259	+0.0079	0.565	+0.19
	Proposed (embedding)	0.9106 $\pm$ 0.0774	–0.0011	0.966	–0.01
MNIST (10)	Random sampling	0.8357 $\pm$ 0.0123	–	–	–
	Predictive entropy	0.8232 $\pm$ 0.0270	–0.0125	0.179	–0.46
	MC-dropout (BALD)	0.8451 $\pm$ 0.0283	+0.0094	0.268	+0.37
	Proposed (embedding)	0.8246 $\pm$ 0.0221	–0.0111	0.102	–0.58

The comparison separates sharply into two regimes. On Optdigits, all three informed strategies outperform random sampling, and every difference survives Holm–Bonferroni correction, with effect sizes from $d = 0.81$ to $d = 2.09$. On the remaining five datasets, no difference survives correction; the closest is the proposed criterion on the Ionosphere dataset (raw $p = 0.039$). We return to the reason for this separation in Section 6.

5.4. Per-class behaviour

Aggregate accuracy is misleading on imbalanced problems. On ECG5000 the model attains 0.90 accuracy while macro- F_1 is only 0.51, and the benefit of informed acquisition is concentrated entirely in the rare classes: Recall on the PVC class ($n = 58$) rises from 0.200 under random sampling to 0.322 under the proposed criterion, a 61% relative improvement, while recall on the dominant Normal class is unchanged at 0.97. Because the two frequent classes account for 94% of the instances, this shift is almost invisible in accuracy. The rarest class, with 14 instances in total, is not learned by any configuration at this budget. Full per-class tables are given in Tables A4 and A5 (Appendix A.5).

5.5. Robustness to annotation noise

Table 6 reports the effect of annotation quality on final performance, with a noiseless oracle included as an upper reference.

Table 6. Effect of annotation quality (10 runs). “Realized accuracy” is the annotation accuracy measured on the instances actually queried during AL. MNIST (10) uses the convolutional encoder.

Dataset	Oracle	Realized accuracy	Final performance
Ionosphere	Noiseless	1.000	0.8981 ± 0.0326
	ChatGPT	0.863 ± 0.066	0.8991 ± 0.0298
	Claude	0.740 ± 0.098	0.8425 ± 0.0361
Optdigits	Noiseless	1.000	0.9282 ± 0.0137
	ChatGPT	0.787 ± 0.040	0.9167 ± 0.0169
	Claude	0.718 ± 0.041	0.9113 ± 0.0181
MNIST (10)	Noiseless	1.000	0.8790 ± 0.0243
	ChatGPT	0.729 ± 0.052	0.8246 ± 0.0221
	Claude	0.667 ± 0.045	0.8092 ± 0.0293

On Optdigits, annotation error rates of 21% and 28% cost 1.2 and 1.7 points of final accuracy relative to noiseless annotation. On MNIST with 10 classes under a convolutional encoder, the same comparison costs 5.4 and 7.0 points. The ordering is monotone in both cases, and the effect is larger where the encoder is stronger: Under the multilayer perceptron, the three oracles on MNIST were within one point of each other, whereas under the convolutional encoder, they separate clearly. A model that extracts more from its labels is correspondingly more sensitive to the labels being wrong, which qualifies the robustness claim rather than contradicting it.

Because an aggregate score can conceal the loss of individual classes, we verify the claim at the class level. On MNIST with 10 classes, per-class recall under noiseless, ChatGPT and Claude annotation is within a few points for all 10 digits, so the degradation is uniform across the label space rather than concentrated on particular classes.

5.6. Contribution of pseudo-labelling

Table 7 isolates the semi-supervised component by running the identical pipeline with pseudo-labelling enabled and disabled, paired by seed.

Table 7. Effect of the pseudo-labelling stage (10 paired runs). Image rows use the convolutional encoder.

Dataset	SSL off	SSL on	Δ	p	d
Sonar	0.7063 ± 0.1090	0.7349 ± 0.0778	+0.0286	0.304	+0.34
Ionosphere	0.8528 ± 0.0345	0.8991 ± 0.0298	+0.0462	0.013	+0.97
Optdigits	0.8857 ± 0.0157	0.9167 ± 0.0169	+0.0310	0.002	+1.41
ECG5000	0.5016 ± 0.0555	0.5088 ± 0.0420	+0.0072	0.557	+0.19
MNIST (3/8)	0.8609 ± 0.1310	0.9106 ± 0.0774	+0.0498	0.392	+0.28
MNIST (10)	0.7904 ± 0.0453	0.8246 ± 0.0221	+0.0342	0.054	+0.70

Pseudo-labeling improves the performance on all six datasets, with a mean gain of +0.033. The improvement is statistically significant with a large effect size on the Ionosphere dataset ($p = 0.013$, $d = 0.97$) and Optdigits dataset ($p = 0.002$, $d = 1.41$), and is marginal on MNIST with 10 classes ($p = 0.054$, $d = 0.70$). This is the most consistent positive result in the study: Unlike the acquisition rule, the semi-supervised component helps in every case examined.

5.7. Sensitivity analysis

A one-at-a-time sweep over the hidden width, dropout rate, pseudo-label confidence threshold, annotation budget, and initial pool size was conducted on all six datasets with five seeds per configuration. Under the multilayer perceptron, the primary metric varies by less than 0.05 across every axis on the datasets where the model learns well; the exception is MNIST with 10 classes, where a hidden width of 32 is insufficient for a 10-way problem. Neither axis alters any conclusion. Full tables are given in Table A3 (Appendix A.4). We emphasize that this sweep reports test-set scores and is therefore a robustness report, not a tuning procedure; the default configuration was fixed in advance and never selected on the basis of test performance.

6. Discussion

Mittal et al. [27] showed that DAL techniques only slightly outperform random sampling when used alone, but that this gap widens considerably when SSL is incorporated. Our results reproduce both halves of that observation within a single experimental design. The semi-supervised component contributes a positive gain on all six datasets, with a mean of +0.033 and statistical significance with a large effect size on two of them. The acquisition rule, by contrast, separates from random sampling on exactly one dataset. The value of the framework therefore resides in the combination of components rather than in the acquisition rule, which is consistent with the view that AL and SSL are synergistic rather than merely additive [11,23].

The clearest evidence for the framework as a whole is its label efficiency. Relative to a fully supervised model trained on the entire pool with ground truth labels, the pipeline recovers between 84.8% and 99.8% of attainable performance while annotating 4.4–37% of the pool and involving no human annotator at any stage. On Ionosphere the 46 chatbot annotations are statistically indistinguishable from full supervision on 208 ground truth labels. Improvement over the initial seed set is observed on every dataset without exception, which corroborates the foundational argument of pool-based active learning [7] even where the particular choice of acquisition rule does not.

6.1. When does informed acquisition help?

The most informative result of this study is the sharp contrast between Optdigits and everything else. On Optdigits, all three informed acquisition rules outperform random sampling and all three survive correction for multiple comparisons, with effect sizes between 0.81 and 2.09. On the other five datasets, none does. Because the protocol, the oracle, the seeds, and the budget construction are identical throughout, the explanation must lie in a property of the datasets rather than of the method.

Our first hypothesis was representation quality. Optdigits presents 10 classes in 62 dimensions, and the model reaches 0.85 accuracy from the seed set alone, so its learned embedding is informative from the outset; MNIST under a multilayer perceptron presents ten classes in 784 raw pixels and the embedding is correspondingly poor. This predicts that a convolutional encoder, which extracts a far better representation from the same images, should restore the separation. The companion experiment of Section 5.1 tests exactly this, and the prediction fails. The convolutional encoder raises the fully supervised ceiling on MNIST with 10 classes from 0.9309 to 0.9719 and the active learning result from 0.7563 to 0.8246, and thus eliminates every collapsed run, and yet the acquisition rules still do not separate: The largest difference against random sampling is +0.0094 at a raw p of 0.268. Representation quality alone is therefore not the explanation.

What distinguishes Optdigits from MNIST is instead their position on the learning curve. On Optdigits, the seed set alone already reaches 86.8% of the fully supervised ceiling, and the annotation budget moves the model across the flat part of the curve, from 0.8521 to 0.9167. On MNIST with 10 classes, the seed set reaches only 60.1% of the ceiling, and the same budget moves the model across the steep part, from 0.5842 to 0.8246. In the steep regime, almost any additional label is informative, so the marginal value of choosing which labels to acquire is small relative to the value of acquiring more of them at all; random sampling captures most of the available gain by itself. In the flat regime the easy gains are exhausted, the remaining errors are concentrated on a smaller and more specific subset of the input space, and the choice of which instances to annotate becomes the binding constraint.

This reading is consistent with the cold-start literature [2, 29] but locates the difficulty more precisely. The usual statement is that acquisition fails when the model is trained on too few labels; our results suggest that what matters is not the absolute number of labels but how far the model still is from what its architecture can achieve on the data. It also yields a practical prescription: Informed acquisition is worth its computational cost once a model is within reach of its ceiling, and is largely wasted before that point.

6.2. Uncertainty in the embedding space

The comparison between the two uncertainty modes is narrower than we originally reported, but survives the corrected protocol in the direction we predicted. The embedding-space entropy H_{emb} exceeds the predictive entropy H_{pred} in mean performance on four of six datasets, and the difference is statistically significant on Ionosphere (+0.0585, paired t -test $p = 0.023$, Wilcoxon $p = 0.023$, $d = 0.86$). This is consistent with the well-documented calibration problem of predictive output distributions in low-data regimes [12]: When the labeled pool contains only a few tens of instances, the softmax layer is a poor estimator of epistemic uncertainty, whereas a proximity-based estimate computed in the learned embedding space can provide a more informative signal, in a spirit related to the manifold-preserving perspective of Ji et al. [34].

We must however report that on Optdigits, the one dataset where the acquisition rule demonstrably matters, the proposed criterion is the weakest of the three that separate from random sampling: +0.0168 against +0.0410 for MC-dropout and +0.0354 for predictive entropy. We therefore do not claim that embedding-space entropy is the strongest acquisition rule available. Its advantage appears to be specific to the very small labeled pools of the binary tabular benchmarks, where the softmax layer is least reliable; once the pool is large enough for the output distribution to be reasonably calibrated, the advantage disappears and reverses.

6.3. Annotation quality

Measuring rather than assuming oracle accuracy exposed an interaction we had not anticipated and which, to our knowledge, has not previously been quantified. The annotation accuracy realized inside the active learning loop is systematically lower than the accuracy the same oracle achieves on a randomly drawn probe set: 0.848 versus 0.787 on Optdigits, 0.762 versus 0.729 on MNIST with 10 classes, and 0.909 versus 0.863 on Ionosphere. The mechanism is straightforward once stated. An acquisition function selects instances that lie near the decision boundary, and those are precisely the instances a language model is most likely to mislabel. Informed acquisition therefore degrades the quality of the labels it obtains, which partially offsets the benefit of selecting informative instances. This coupling is absent when the annotator is a domain expert whose accuracy is approximately uniform over the input space, and it suggests a concrete design direction: Acquisition criteria that jointly score informativeness and expected annotator reliability.

The cost of annotation noise is modest but not negligible, and it depends on the encoder. On Optdigits, error rates of 21% and 28% cost 1.2 and 1.7 points of final accuracy relative to noiseless annotation. On MNIST with 10 classes under a convolutional encoder, the same comparison costs 5.4 and 7.0 points, whereas under a multilayer perceptron, the three oracles were within one point of each other. A weaker model is insensitive to label noise because it cannot exploit correct labels either; a stronger model extracts more from its labels and is correspondingly more exposed when they are wrong. Reports of robustness to annotation noise should therefore be read together with the capacity of the model that was trained, and we would caution against generalizing such claims across architectures.

The self-training loop nevertheless acts as a partial noise-mitigation mechanism, since pseudo-labels are recomputed from scratch at each iteration and an incorrect annotation can be overridden once the model's confidence in the true class exceeds τ . Per-class inspection confirms that the degradation is uniform across the label space rather than concentrated on particular classes.

6.4. Imbalanced data

The imbalanced multi-class setting is where accuracy is most misleading and where the benefit of informed acquisition is most visible. On ECG5000, the model achieves 0.90 accuracy, while the macro- F_1 is only 0.51, because the two frequent classes account for 94% of the instances. Per-class inspection shows that the entire benefit of informed acquisition is concentrated in the rare classes: Recall on the PVC (premature ventricular contraction) class in the ECG5000 dataset rises from 0.200 under random sampling to 0.322 under the proposed criterion, while recall on the dominant normal class is unchanged. The rarest class, with 14 instances in total, is not learned by any configuration at this budget; pseudo-labels for minority classes less frequently exceed the fixed confidence threshold $\tau = 0.75$, so the self-training stage reinforces the majority structure it is given. Adaptive thresholding strategies such as FlexMatch [18], which relax the threshold on a per-class basis according to the learning progress, offer a natural avenue for improving minority-class recall within the proposed framework.

6.5. Limitations

Several limitations should be acknowledged. First, the proposed acquisition criterion outperforms random sampling after correction for multiple comparisons on one of six datasets, and on that dataset, it is the weakest of the three rules that do so; the contribution of this work lies in the annotation framework and its label efficiency, not in the acquisition rule. Second, the embedding-based criterion exhibits a failure mode: On MNIST 3 versus 8, its standard deviation is 0.077 versus 0.028 for random sampling, and under the multilayer perceptron, one run in ten terminated at the majority-class floor. We attribute this to degeneracy of the inverse-distance posterior when the labeled anchor set is very small and the learned embedding has partially collapsed; a distance floor, or a fallback to predictive entropy when the anchor set is below a minimum size, would be a straightforward mitigation. Third, the rarest ECG5000 class is never learned at these budgets, so the framework should not be presented as suitable for extreme class imbalance without modification. Fourth, the main experiments use an annotation simulator calibrated to accuracies measured from the live models on probe sets, which reproduces the aggregate error rate and its concentration on ambiguous instances but cannot reproduce every systematic bias of a particular model version; the released implementation supports live API (Application Programming Interface) annotation, and a full replication under live querying would strengthen the claim. Fifth, the account we give of when informed acquisition helps rests on a contrast between two datasets and should be treated as a hypothesis to be tested directly, for example, by varying the initial pool size on a single dataset so that the model is placed at different points on its learning curve under an otherwise identical protocol.

7. Conclusions

In this study, we proposed DAL-Chat-SSL, a DAL framework that replaces human annotators with AI chatbot oracles and combines entropy-based query selection with pseudo-label self-training for classification tasks beyond the text domain. The framework was evaluated on six benchmark datasets spanning tabular, time-series, and image modalities, over 10 independent runs each, using a held-out test split that never participates in training, pseudo-labeling, acquisition, or model selection, and against

a fully supervised reference trained on the entire pool with ground truth labels.

The central finding is one of label efficiency: With no human annotator at any stage, the framework recovers between 84.8% and 99.8% of fully supervised performance while annotating 4.4–37% of the available pool, and on Ionosphere, 46 chatbot annotations are statistically indistinguishable from full supervision on 208 ground truth labels. Together with the robustness to annotation error and the measured contribution of the semi-supervised stage reported in Section 6, these results indicate that LLM-based annotation oracles are a viable and scalable alternative to human labeling in AL pipelines, provided that oracle noise is accounted for through robust SSL.

Two results run counter to our initial expectations and are stated as such. Acquisition strategies are separate from random sampling on one of the six benchmarks, and on that benchmark, the proposed embedding-space criterion is the weakest of the three rules that do so; we relate this pattern to how far the model still is from its attainable ceiling when the annotation budget is spent. The recurrent encoder of our original formulation was outperformed by an order-agnostic multilayer perceptron on every dataset on which it was tested. The contribution of this work therefore lies in the annotation framework and its label efficiency rather than in the acquisition rule or the encoder.

Future work will address acquisition criteria that jointly score the informativeness and expected annotator reliability, adaptive confidence thresholding in the self-training stage to improve recall on rare classes [18], and a full replication under live API querying, which the released implementation supports.

Use of generative AI tools declaration

AI chatbots (ChatGPT and Claude) were used in two distinct capacities in this work. First, they appear as an object of study: Within the experimental methodology described in Section 3.3, they serve as annotation oracles whose labeling accuracy is measured and reported. Second, during the preparation of this revised manuscript, the author used a generative AI assistant for language editing and for support in implementing the experimental pipeline and drafting the description of results. All experiments were designed, executed, and verified by the author, who reviewed and edited all resulting text and code and takes full responsibility for the content of the publication.

Acknowledgments

This research received no external funding. The datasets used in this study are publicly available benchmark datasets (Sonar, Ionosphere, Optdigits, ECG5000, and MNIST).

Conflict of interest

The author declares no conflicts of interest.

References

1. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei, *ImageNet: A large-scale hierarchical image database*, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, 248–255.

2. M. Wu, C. Li, Z. Yao, Deep active learning for computer vision tasks: Methodologies, applications, and challenges, *Appl. Sci.*, **12** (2022), 8103. <https://doi.org/10.3390/app12168103>
3. S. Budd, E. C. Robinson, B. Kainz, A survey on active learning and human-in-the-loop deep learning for medical image analysis, *Med. Image Anal.*, **71** (2021), 102062. <https://doi.org/10.1016/j.media.2021.102062>
4. Y. Tang, Y. Zhou, T. Wu, C. Wang, Z. Li, K. Li, AI-driven predictive maintenance for medical imaging equipment: A deep learning framework based on the IoMT data, *Reliab. Eng. Syst. Saf.*, **270** (2026), 112152. <https://doi.org/10.1016/j.ress.2025.112152>
5. C. Ma, H. Li, B. Xue, C. Wang, K. Li, Risk-oriented degradation modeling and maintenance decision support for computed tomography X-ray tubes using mechanism-informed learning, *Reliab. Eng. Syst. Saf.*, **275** (2026), 112837. <https://doi.org/10.1016/j.ress.2026.112837>
6. Y. Zhao, H. Li, Z. Li, K. Li, C. Wang, Hybrid Bayesian and deep learning-based degradation modeling for medical imaging equipment under multi-condition Internet of Medical Things data streams, *Reliab. Eng. Syst. Safe.*, **277** (2027), 112937. <https://doi.org/10.1016/j.ress.2026.112937>
7. B. Settles, *Active learning literature survey*, Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2010. Available from: <https://burrsettles.com/pub/settles.activelearning.pdf>.
8. P. Kumar, A. Gupta, Active learning query strategies for classification, regression, and clustering: A survey, *J. Comput. Sci. Technol.*, **35** (2020), 913–945. <https://doi.org/10.1007/s11390-020-9487-4>
9. D. Li, Z. Wang, Y. Chen, R. Jiang, W. Ding, M. Okumura, A survey on deep active learning: Recent advances and new frontiers, *IEEE Trans. Neural Netw. Learn. Syst.*, **36** (2025), 5879–5899. <https://doi.org/10.1109/TNNLS.2024.3396463>
10. P. Ren, Y. Xiao, X. Chang, P. Y. Huang, Z. Li, B. B. Gupta, et al., A survey of deep active learning, *ACM Comput. Surv.*, **54** (2021), 1–40. <https://doi.org/10.1145/3472291>
11. X. Zhan, Q. Wang, K. H. Huang, H. Xiong, D. Dou, A. B. Chan, A comparative survey of deep active learning, *arXiv preprint*, 2022. <https://doi.org/10.48550/arXiv.2203.13450>
12. Y. Gal, R. Islam, Z. Ghahramani, *Deep Bayesian active learning with image data*, In: Proceedings of the International Conference on Machine Learning (ICML), 2017, 1183–1192.
13. O. Sener, S. Savarese, *Active learning for convolutional neural networks: A core-set approach*, In: Proceedings of the International Conference on Learning Representations (ICLR), 2018, arXiv:1708.00489.
14. A. Kirsch, J. van Amersfoort, Y. Gal, BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning, *Adv. Neural Inf. Process. Syst.*, **32** (2019), 7026–7037.
15. D. Yoo, I. S. Kweon, *Learning loss for active learning*, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, 93–102.
16. S. Sinha, S. Ebrahimi, T. Darrell, *Variational adversarial active learning*, In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, 5972–5981.

17. K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, et al., FixMatch: Simplifying semi-supervised learning with consistency and confidence, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 596–608.
18. B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, et al., FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 18408–18419.
19. D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. A. Raffel, MixMatch: A holistic approach to semi-supervised learning, *Adv. Neural Inf. Process. Syst.*, **32** (2019), 5049–5059.
20. X. Yang, Z. Song, I. King, Z. Xu, A survey on deep semi-supervised learning, *IEEE Trans. Knowl. Data Eng.*, **35** (2023), 8934–8954. <https://doi.org/10.1109/TKDE.2022.3220219>
21. T. Fredriksson, J. Bosch, H. H. Olsson, An empirical evaluation of deep semi-supervised learning, *Int. J. Data Sci. Anal.*, **20** (2025), 4127–4148. <https://doi.org/10.1007/s41060-024-00713-8>
22. Z. Wen, O. Pizarro, S. Williams, Active self-semi-supervised learning for few labeled samples, *Neurocomputing*, **614** (2025), 128772. <https://doi.org/10.1016/j.neucom.2024.128772>
23. T. Wan, K. Xu, T. Yu, X. Wang, D. Feng, B. Ding, et al., A survey of deep active learning for foundation models, *Intell. Comput.*, **2** (2023), 0058. <https://doi.org/10.34133/icomputing.0058>
24. K. Wang, D. Zhang, Y. Li, R. Zhang, L. Lin, Cost-effective active learning for deep image classification, *IEEE T. Circ. Syst. Vid.*, **27** (2017), 2591–2600. <https://doi.org/10.1109/TCSVT.2016.2589879>
25. S. Song, D. Berthelot, A. Rostamizadeh, Combining MixMatch and active learning for better accuracy with fewer labels, *arXiv preprint*, 2019. <https://doi.org/10.48550/arXiv.1912.00594>
26. J. Guo, H. Shi, Y. Kang, K. Kuang, S. Tang, Z. Jiang, et al., *Semi-supervised active learning for semi-supervised models: Exploit adversarial examples with graph-based virtual labels*, In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, 2896–2905.
27. S. Mittal, J. Niemeijer, Ö. Çiçek, M. Tatarchenko, J. Ehrhardt, J. P. Schäfer, et al., Realistic evaluation of deep active learning for image classification and semantic segmentation, *Int. J. Comput. Vis.*, **133** (2025), 4294–4316. <https://doi.org/10.1007/s11263-025-02372-z>
28. Y. Xia, S. Mukherjee, Z. Xie, J. Wu, X. Li, R. Aponte, et al., *From selection to generation: A survey of LLM-based active learning*, In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, 14552–14569. <https://doi.org/10.18653/v1/2025.acl-long.708>
29. M. Bayer, J. Lutz, C. Reuter, ActiveLLM: Large language model-based active learning for textual few-shot scenarios, *Trans. Assoc. Comput. Linguist.*, **14** (2026), 1–22. <https://doi.org/10.1162/TACL.a.63>
30. Y. Qi, X. Yang, J. Lu, G. Guo, J. Enticott, G. Liu, et al., *Next generation active learning: Mixture of LLMs in the loop*, In: Proceedings of the AAAI Conference on Artificial Intelligence, **40** (2026), 24909–24917. <https://doi.org/10.1609/aaai.v40i29.39678>

-
31. K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., *Learning phrase representations using RNN encoder-decoder for statistical machine translation*, In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>
 32. C. E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.*, **27** (1948), 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
 33. D. P. Kingma, J. Ba, *Adam: A method for stochastic optimization*, In: Proceedings of the International Conference on Learning Representations (ICLR), 2015.
 34. Y. Ji, V. S. Kaza, N. Artham, T. Wang, Deep active learning with manifold-preserving trajectory sampling, *arXiv preprint*, 2024. <https://doi.org/10.48550/arXiv.2410.15605>

Appendix

A.1. Encoder comparison

Table A1. Encoder comparison under an identical protocol, $n = 10$ paired runs per cell. Primary metric is accuracy except on ECG5000, where it is macro- F_1 . MLP represents multilayer perceptron.

Dataset	Strategy	GRU	MLP	Δ	p
Sonar (floor: 0.5397)	Random sampling	0.5667 ± 0.0763	0.7413 ± 0.0613	+0.1746	< 0.001
	Uncertainty sampling	0.5476 ± 0.1160	0.7333 ± 0.0579	+0.1857	0.001
	MC-dropout (BALD)	0.5619 ± 0.0793	0.7778 ± 0.0725	+0.2159	< 0.001
	Coreset	0.5794 ± 0.0590	0.7556 ± 0.0695	+0.1762	< 0.001
	Proposed (embedding)	0.5413 ± 0.0692	0.7349 ± 0.0778	+0.1937	0.001
	<i>Pooled</i>	$+0.1892 \pm 0.0997$ over 50 paired observations			
Ionosphere (floor: 0.6415)	Random sampling	0.7189 ± 0.1070	0.8519 ± 0.0561	+0.1330	0.001
	Uncertainty sampling	0.6245 ± 0.0834	0.8406 ± 0.0653	+0.2160	< 0.001
	MC-dropout (BALD)	0.6462 ± 0.2134	0.8623 ± 0.0486	+0.2160	0.011
	Coreset	0.6226 ± 0.1812	0.8321 ± 0.0838	+0.2094	0.017
	Proposed (embedding)	0.6830 ± 0.1689	0.8991 ± 0.0298	+0.2160	0.005
	<i>Pooled</i>	$+0.1981 \pm 0.1704$ over 50 paired observations			
ECG5000 (macro- F_1)	Random sampling	0.3812 ± 0.0163	0.4863 ± 0.0658	+0.1051	0.001
	Predictive entropy	0.3922 ± 0.0544	0.4969 ± 0.0879	+0.1047	0.003
	Smallest margin	0.4082 ± 0.0355	0.5331 ± 0.0171	+0.1250	< 0.001
	MC-dropout (BALD)	0.2545 ± 0.1294	0.4495 ± 0.0770	+0.1950	0.001
	Coreset	0.3890 ± 0.0582	0.4807 ± 0.0624	+0.0917	0.006
	Proposed (embedding)	0.4080 ± 0.0275	0.5088 ± 0.0420	+0.1007	0.001
	<i>Pooled</i>	$+0.1204 \pm 0.0828$ over 60 paired observations			
MNIST 3 vs. 8 (floor: 0.5111)	Random sampling	0.5844 ± 0.1040	0.8968 ± 0.0302	+0.3123	< 0.001
	Uncertainty sampling	0.5694 ± 0.0696	0.9027 ± 0.0546	+0.3333	< 0.001
	MC-dropout (BALD)	0.5644 ± 0.1066	0.8973 ± 0.0864	+0.3328	< 0.001
	Coreset	0.5077 ± 0.0593	0.9223 ± 0.0191	+0.4147	< 0.001
	Proposed (embedding)	0.5520 ± 0.0826	0.8814 ± 0.1390	+0.3294	< 0.001
	<i>Pooled</i>	$+0.3445 \pm 0.1141$ over 50 paired observations			
MNIST (10) (floor: 0.1124)	Random sampling	0.3143 ± 0.1442	0.7691 ± 0.0083	+0.4547	< 0.001
	Predictive entropy	0.1668 ± 0.0662	0.7416 ± 0.0389	+0.5748	< 0.001
	MC-dropout (BALD)	0.1040 ± 0.0022	0.7409 ± 0.0431	+0.6369	< 0.001
	Coreset	0.1074 ± 0.0070	0.7541 ± 0.0124	+0.6466	< 0.001
	Proposed (embedding)	0.2167 ± 0.1209	0.7563 ± 0.0234	+0.5396	< 0.001
	<i>Pooled</i>	$+0.5611 \pm 0.1190$ over 60 paired observations			

A.2. Acquisition strategies under the recurrent encoder

Table A2. Final test performance under the GRU encoder ($n = 10$ runs, mean $\pm$ sd). Δ is measured against random sampling in the primary metric of each dataset.

Dataset	Strategy	Primary metric	Δ	Collapse
Sonar	Random sampling	0.5667 ± 0.0763	–	5/10
	Uncertainty sampling	0.5476 ± 0.1160	–0.0190	5/10
	MC-dropout (BALD)	0.5619 ± 0.0793	–0.0048	6/10
	Coreset	0.5794 ± 0.0590	+0.0127	4/10
	Proposed (embedding)	0.5413 ± 0.0692	–0.0254	7/10
Ionosphere	Random sampling	0.7189 ± 0.1070	–	5/10
	Uncertainty sampling	0.6245 ± 0.0834	–0.0943	8/10
	MC-dropout (BALD)	0.6462 ± 0.2134	–0.0726	6/10
	Coreset	0.6226 ± 0.1812	–0.0962	4/10
	Proposed (embedding)	0.6830 ± 0.1689	–0.0358	4/10
ECG5000	Random sampling	0.3812 ± 0.0163	–	0/10
	Predictive entropy	0.3922 ± 0.0544	+0.0110	0/10
	MC-dropout (BALD)	0.2545 ± 0.1294	–0.1267	6/10
	Coreset	0.3890 ± 0.0582	+0.0078	0/10
	Proposed (embedding)	0.4080 ± 0.0275	+0.0268	0/10
MNIST 3 vs. 8	Random sampling	0.5844 ± 0.1040	–	5/10
	Uncertainty sampling	0.5694 ± 0.0696	–0.0151	4/10
	MC-dropout (BALD)	0.5644 ± 0.1066	–0.0200	6/10
	Coreset	0.5077 ± 0.0593	–0.0768	9/10
	Proposed (embedding)	0.5520 ± 0.0826	–0.0325	6/10
MNIST (10)	Random sampling	0.3143 ± 0.1442	–	1/10
	Predictive entropy	0.1668 ± 0.0662	–0.1476	4/10
	MC-dropout (BALD)	0.1040 ± 0.0022	–0.2103*	10/10
	Coreset	0.1074 ± 0.0070	–0.2069*	9/10
	Proposed (embedding)	0.2167 ± 0.1209	–0.0976	3/10

* Significantly *worse* than random sampling after Holm–Bonferroni correction ($p = 0.006$ in both cases). Per-class inspection of these configurations shows a single digit that was predicted and zero recall on the other nine.

A.3. Rendered example prompts

Sonar (tabular), query index 102, ground truth = 1 (mine).

--- SYSTEM MESSAGE ---

You are an expert annotator for a tabular classification task on the Sonar (UCI) dataset. You will be shown one sample rendered as text. Assign exactly one class label. Answer with a single JSON object of the form {"label": <int>, "confidence": <float 0-1>} and nothing else.

--- USER MESSAGE ---

TASK

Classify one tabular sample from Sonar (UCI).

CLASS DEFINITIONS

0 = Rock

1 = Mine

INPUT FORMAT

normalized sonar return energy in 60 frequency bands

The sample is given as name=value pairs. Values are z-normalized over the whole dataset (mean 0, sd 1).

LABELED EXAMPLES

EXAMPLE 1 (label = 0, Rock)

b01=-0.400, b02=-0.041, b03=-0.027, b04=-0.715, b05=0.364, b06=-0.101, b07=0.522, b08=0.298, b09=1.125, b10=0.021, ... , b60=-0.659

QUERY SAMPLE

b01=1.288, b02=2.511, b03=2.164, b04=2.066, b05=1.234, b06=-0.822, b07=-0.625, b08=-0.796, b09=-0.345, b10=0.201, ... , b60=1.174

Respond with JSON only: {"label": <int>, "confidence": <float>}

ECG5000 (time series), query index 643, ground truth = 0 (normal).

--- SYSTEM MESSAGE ---

You are an expert annotator for a timeseries classification task on the ECG5000 (UCR) dataset. You will be shown one sample rendered as text. Assign exactly one class label. Answer with a single JSON object of the form {"label": <int>, "confidence": <float 0-1>} and nothing else.

--- USER MESSAGE ---

TASK

Classify one timeseries sample from ECG5000 (UCR).

CLASS DEFINITIONS

- 0 = Normal
- 1 = R-on-T PVC
- 2 = PVC
- 3 = SP/Escape
- 4 = Unclassified

INPUT FORMAT

z-normalized single heartbeat, 140 samples

The sample is given as a whitespace-separated sequence of z-normalized amplitudes in temporal order.

LABELED EXAMPLES

EXAMPLE 1 (label = 3, SP/Escape)

(uniformly decimated from 140 to 70 points)

+0.21 +0.42 +0.04 -1.53 -1.33 -1.18 -1.65 -0.11 -0.28 -0.25 -0.40 -0.06
-0.16 -0.07 +0.21 -0.06 +0.23 +0.24 +0.23 +0.12 ... +1.12 +1.53

QUERY SAMPLE

(uniformly decimated from 140 to 70 points)

+2.24 +1.13 +1.52 +3.09 +3.15 +3.71 +4.49 +4.43 +3.75 +3.53 +3.26 +2.89
+2.78 +2.43 +2.17 +2.01 +1.79 +1.51 +1.42 +1.55 ... -0.55 -1.86

Respond with JSON only: {"label": <int>, "confidence": <float>}

MNIST 3 vs. 8 (image), ground truth = 1 (digit 8).

--- SYSTEM MESSAGE ---

You are an expert annotator for an image classification task on the MNIST 3 vs 8 dataset. You will be shown one sample rendered as text. Assign exactly one class label. Answer with a single JSON object of the form {"label": <int>, "confidence": <float 0-1>} and nothing else.

--- USER MESSAGE ---

TASK

Classify one image sample from MNIST 3 vs 8.

CLASS DEFINITIONS

0 = digit 3

1 = digit 8

INPUT FORMAT

28x28 grayscale, [0,1]

The image is average-pooled to a 16x16 grid and quantized to 10 intensity levels, then rendered twice: as an ASCII picture and as a digit matrix. Both encode the same array.

QUERY SAMPLE

ASCII intensity map (16x16, ' '=dark, '@'=bright):

```

      :*@@@
    %@@@
  +@@@#
=@@@*
:%%@*
*@@=
.@@= .-#
=@@- *%%
:@@@@@@
#%@@@@
..:@@@#:
```

Quantized intensity matrix (0=dark ... 9=bright):

```

0000000000026999
0000000000089999
0000000000059997
0000000000499960
00000000028999600
0000000069994000
0000000199940137
0000000499936899
0000000299999999
0000000078999999
0000000011299972
```

Respond with JSON only: {"label": <int>, "confidence": <float>}

A.4. Sensitivity analysis

Table A3. One-at-a-time sensitivity, final test accuracy (mean $\pm$ SD, five seeds per cell). Sensitivity analysis was run on the four datasets.

Axis	Value	Sonar		Ionosphere		MNIST 3/8		ECG5000	
		GRU	MLP	GRU	MLP	GRU	MLP	GRU	MLP
Hidden width	32	0.486	0.746	0.762	0.859	0.491	0.919	0.760	0.901
	64	0.578	0.718	0.634	0.896	0.621	0.914	0.898	0.916
	128	0.546	0.765	0.849	0.889	0.575	0.931	0.893	0.883
Dropout	0.1	0.625	0.752	0.719	0.879	0.609	0.933	0.903	0.880
	0.3	0.578	0.718	0.634	0.896	0.575	0.931	0.898	0.916
	0.5	0.562	0.727	0.766	0.866	0.559	0.926	0.886	0.914
SSL threshold	0.60	0.613	0.771	0.709	0.874	0.575	0.865	0.785	0.907
	0.75	0.578	0.718	0.634	0.896	0.575	0.931	0.898	0.916
	0.90	0.673	0.724	0.730	0.885	0.575	0.933	0.888	0.902
Rounds	2	0.508	0.718	0.693	0.826	0.594	0.894	0.720	0.870
	4	0.597	0.730	0.655	0.859	0.518	0.933	0.776	0.895
	6	0.578	0.718	0.634	0.896	0.575	0.931	0.898	0.916
Initial pool / class	4	0.521	0.746	0.757	0.838	0.544	0.927	0.898	0.916
	8	0.578	0.718	0.634	0.896	0.524	0.921	0.878	0.903
	16	0.533	0.791	0.698	0.889	0.601	0.942	0.861	0.901

A.5. Per-class recall

Table A4. Per-class recall on ECG5000 with the MLP encoder ($n = 10$ runs). Class supports are given in parentheses. The benefit of informed acquisition is concentrated in the rare classes; the rarest class is not learned by any configuration at this budget.

Strategy	Normal (1751)	R-on-T (1060)	PVC (58)	SP/Escap (117)	Uncl. (14)
Random sampling	0.970 ± 0.022	0.899 ± 0.044	0.200 ± 0.218	0.311 ± 0.161	0.125 ± 0.212
Predictive entropy	0.980 ± 0.016	0.808 ± 0.171	0.317 ± 0.193	0.411 ± 0.198	0.050 ± 0.105
Smallest margin	0.981 ± 0.013	0.884 ± 0.068	0.306 ± 0.144	0.397 ± 0.128	0.100 ± 0.129
MC-dropout (BALD)	0.932 ± 0.072	0.894 ± 0.053	0.111 ± 0.148	0.283 ± 0.195	0.100 ± 0.129
Coreset	0.954 ± 0.075	0.909 ± 0.071	0.244 ± 0.227	0.263 ± 0.138	0.075 ± 0.169
Proposed (embedding)	0.969 ± 0.026	0.909 ± 0.061	0.322 ± 0.171	0.349 ± 0.184	0.025 ± 0.079
<i>GRU encoder, same protocol</i>					
Random sampling	0.985 ± 0.011	0.919 ± 0.031	0.033 ± 0.060	0.003 ± 0.009	0.025 ± 0.079
Proposed (embedding)	0.984 ± 0.008	0.916 ± 0.037	0.083 ± 0.112	0.046 ± 0.081	0.075 ± 0.169

Table A5. Per-class recall by annotation quality on MNIST with 10 classes with MLP encoder ($n = 10$ runs).

Oracle	Annot. acc.	0	1	2	3	4	5	6	7	8	9
Noiseless	1.00	0.89	0.87	0.79	0.77	0.73	0.65	0.85	0.79	0.63	0.67
ChatGPT	0.75	0.88	0.91	0.75	0.72	0.68	0.65	0.86	0.79	0.64	0.65
Claude	0.68	0.90	0.92	0.76	0.74	0.69	0.64	0.83	0.82	0.66	0.62



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)