# All models are wrong, some are useful: Model Selection with Limited Labels

**Patrik Okanovic**
ETH Zurich
patrik.okanovic@inf.ethz.ch

**Andreas Kirsch**
blackhc@gmail.com

**Jannes Kasper**
TU Delft
J.Kasper@student.tudelft.nl

**Torsten Hoefler**
ETH Zurich
htor@ethz.ch

**Andreas Krause**
ETH Zurich
krausea@ethz.ch

**Nezihe Merve Gürel**
TU Delft
n.m.gurel@tudelft.nl

## Abstract

We introduce MODEL SELECTOR[1], a framework for label-efficient selection of pretrained classifiers. Given a pool of unlabeled target data, MODEL SELECTOR samples a small subset of highly informative examples for labeling, in order to efficiently identify the *best* pretrained model for deployment on this target dataset. Through extensive experiments, we demonstrate that MODEL SELECTOR drastically reduces the need for labeled data while consistently picking the best or near-best performing model. Across 18 model collections on 16 different datasets, comprising over 1,500 pretrained models, MODEL SELECTOR reduces the labeling cost by up to $94.15\%$ to identify the best model compared to the cost of the strongest baseline. Our results further highlight the robustness of MODEL SELECTOR in model selection, as it reduces the labeling cost by up to $72.41\%$ when selecting a near-best model, whose accuracy is only within $1\%$ of the best model.

## 1 Introduction

The abundance of openly available machine learning models poses a dilemma: selecting the best pretrained model for domain-specific applications becomes increasingly challenging. As the development and deployment of large-scale machine learning models have been accelerating at a rapid pace (He et al., 2016; Chiu et al., 2018; Mann et al., 2020; Radford et al., 2019), a wide selection of pretrained models for natural language processing (Devlin et al., 2019; Liu et al., 2019) and computer vision (Krizhevsky et al., 2012; Xie et al., 2017), varying in architecture, type, and complexity, are now accessible through various open-source and academic platforms (HuggingFace, 2024; PyTorch, 2024; TensorFlow, 2024). AutoML platforms (AWS, 2024; GoogleCloud, 2024) further increase this variety by providing researchers and developers with instant access to powerful bespoke models through automated workflows and pretrained model repositories. Thanks to advances in training methods and model architecture, many of these models now support zero-shot learning (Brown, 2020; Xian et al., 2018; Pourpanah et al., 2022; Radford et al., 2021), and can tackle new tasks without requiring fine-tuning or updates to their weights. Traditionally, models are often picked by hand, involving intuition-based sampling and evaluation, which quickly becomes impractical as the number of models and potential evaluation samples increases, risking suboptimal choices and inefficient resource utilization.

To address this, several techniques perform automated model validation by selecting and labeling a small informative subset of data examples to assess model performance and facilitate label-efficient model selection and evaluation (Sawade et al., 2012; Matsuura & Hara, 2023; Kossen et al., 2021), which are also commonly referred to as *active model selection* (Karimi et al., 2021; Matsuura & Hara, 2023; Liang et al., 2020; Gardner et al., 2015). However, many of the methods in the pool-based setting (characterized by the availability of a large pool of unlabeled data) either assume a fixed number of available models (Sawade et al., 2012) or

---

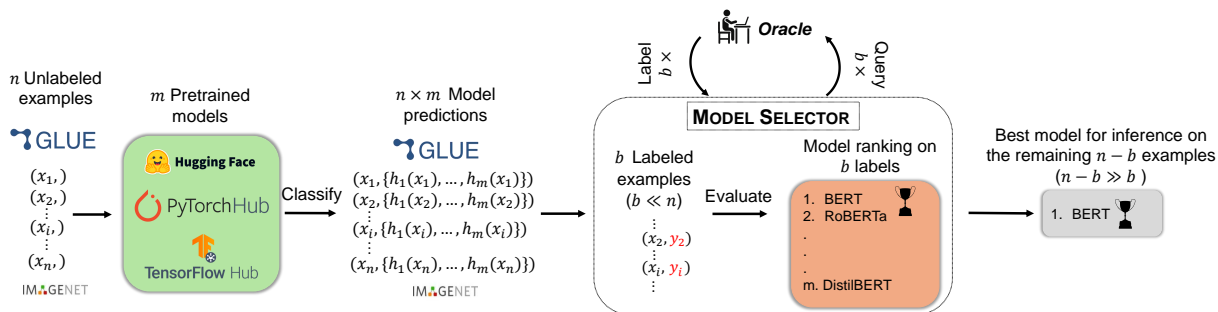[1]Source code: https://github.com/RobustML-Lab/model-selector

Figure 1: An overview of our label-efficient model selection pipeline with MODEL SELECTOR. Given a pool of $n$ unlabeled data examples and a set of $m$ pretrained classifiers, MODEL SELECTOR aims to select $b$ (with $b \ll n$) unlabeled examples that, once labeled, can identify the best pretrained model.

are restricted to specific model families (Gardner et al., 2015). Others require retraining the model each time a new example is labeled (Ali et al., 2014), which can be computationally expensive and potentially unnecessary for ready-to-deploy models.

In this work, we aim to extend these settings and ask: *Given a pool of unlabeled data, how can we identify the most informative examples to label in order to select the best classifier for this data,* both in a model-agnostic and efficient manner?

**Contributions**   In this paper, we introduce MODEL SELECTOR, a fully *model-agnostic* approach for identifying the best pretrained model using a limited number of labeled examples. Unlike existing methods, which often rely on assumptions about the model architectures or require detailed knowledge of model internals, our approach treats models purely as black-boxes requiring no additional training and anything beyond hard predictions, i.e., the predicted label without predictive distribution. To the best of our knowledge, this is the first work in the pool-based setting that is both model-agnostic and relies solely on hard predictions.

Formally, given a pool of $n$ newly collected, unlabeled data examples and a labeling budget $b$ with $b \ll n$, MODEL SELECTOR aims to select $b$ unlabeled examples that, once labeled, provide the maximum expected information gain about the best model (model with the highest utility) for this data. To achieve this, we employ the *most informative selection policy* (Chen et al., 2015; Kirsch et al., 2019; Treven et al., 2023; Chattopadhyay et al.; Ding et al., 2024), where we greedily label the most informative examples until the labeling budget $b$ is exhausted. We define informativeness in terms of mutual information between the best model and data labels, relying on a *single-parameter* model as proxy that nonetheless sufficiently captures the relationship between them. Despite its simplicity, the proposed single-parameter model is highly effective in identifying the best model under limited labels. Moreover, for such a simplified model, the most informative selection policy has a provable near-maximal utility, as shown by Chen et al. (2015).

We conduct extensive experiments comparing our method against a range of adapted methods over $1,500$ models across $18$ different model collections and $16$ different datasets. We observe that MODEL SELECTOR consistently outperforms all the baselines. In particular, MODEL SELECTOR reduces the labeling cost by up to $94.15\%$ compared to the best competing baseline in finding the best model. We also show that MODEL SELECTOR finds a near-best model with a reduction in labeling cost by up to $72.41\%$ within $1\%$ accuracy of the best model.

Once the best pretrained machine learning model is identified, we then deploy it for making predictions on the remaining unlabeled samples. An overview of MODEL SELECTOR is provided in Figure 1. Our approach is designed to reduce the expenses related to pretrained model selection for classifiers, marking a step forward in efficient machine learning practice.

2

## 2  Related Work

To date, label-efficient model selection has been studied primarily in the stream-based (online) setting (Piratla et al., 2021; Liu et al., 2022a; Kassraie et al., 2023; Madani et al., 2012; Xia et al., 2024a; Liu et al., 2022b; Li et al., 2024a,b; Xia et al., 2024b; Karimi et al., 2021). In the pool-based setting, where we have access to the entire collection of unlabeled examples, the literature often makes strong assumptions, such as predefined model families or learning tasks. For example, both Liang et al. (2020) and Zhao et al. (2008) assume specific learning tasks (binary time series classification or graph-based semi-supervised learning, respectively), while Gardner et al. (2015) limit model selection to Gaussian processes. Other works including Ali et al. (2014) consider a general class of hypotheses but rely on sequential training. Bhargav et al. (2024) introduce a misclassification penalty framework for hypothesis testing in a batched setting, and Kumar & Raj (2018) perform classifier risk estimation on stratified batches without strong assumptions.

Several works align closely with our setting, focusing on model selection without the previously mentioned assumptions. These works focus on evaluating the risk of a single model (Sawade et al., 2010; Katariya et al., 2012; Kossen et al., 2021), comparing two models (Sawade et al., 2012; Leite & Brazdil, 2010), or, like ours, comparing multiple models (Matsuura & Hara, 2023). We adapt the method of Sawade et al. (2012) for pairwise model comparison in our setting (AMC) and use VMA (Matsuura & Hara, 2023) as a baseline. While AMC minimizes the asymptotic variance of the estimated loss for each model, VMA minimizes the variance of the estimated test loss conditioned on previously queried examples. Due to the lack of applicable baselines, we also employ uncertainty and margin sampling (Dagan & Engelson, 1995; Seung et al., 1992; Freund et al., 1997).

Unlike previous works, we propose a fully agnostic method, making no assumptions about model families, tasks, or the use of soft predictions. Additionally, our work is related to optimal information-gathering strategies, with a discussion of related work deferred to the Appendix C, in order to prioritize works with the same objective of label-efficient model selection.

## 3  Model Selector

In this section, we present our label-efficient model selection algorithm, MODEL SELECTOR. We start by describing the problem setup, where we connect the unknown best model to the true data labels using a single-parameter likelihood model. Then, we introduce the MODEL SELECTOR algorithm, which is designed to identify the best pretrained model with limited labels. We end the section by providing further details on the algorithm and explain how this single parameter can be learned directly from the data without requiring any labels.

### 3.1  Problem Setting

Consider a pool of $n$ freshly collected unlabeled examples denoted by $\mathcal{D} = \{(x_i, Y_i) \in \mathcal{X} \times \mathcal{Y} \,|\, i \in [n]\}$, whose labels $Y_i$ are unobserved[2]. We denote the true labels by $y_i$.

For a given set of $m$ pretrained, ready-to-deploy classifiers $\mathcal{M} = \{h_j : \mathcal{X} \mapsto \mathcal{Y} \,|\, j \in [m]\}$, and a labeling budget $b$ with $b \ll n$, our aim is to identify the best pretrained model $h(\cdot)$ among $\mathcal{M}$ by querying a small number of at most $b$ labels, and perform inference for the remaining $n - b$ examples using this model. The *best* model here is defined as the model that fits the data $\mathcal{D}$ best if all $n$ labels $y_{i \in [n]}$ were available. In this work, we consider the model with the highest utility (accuracy) on $n$ labels as the best. That is, $h^* = \arg\max_{h_j \in \mathcal{M}} 1/n \sum_{i=[n]} \mathbb{1}[h_j(x_i) = y_i]$.

The best model is unknown to us without labels. We represent this unknown by using a random variable $H$, with a known prior distribution $H \sim \mathbb{P}(H = h_j)$ over the set of candidate models $\mathcal{M} = \{h_j \,|\, j \in [m]\}$. Our goal is to uncover the identity of $H$ for the target data $\mathcal{D}$ by using as few labels as possible. That is, we want $h^* = \arg\max_{h_j \in \mathcal{M}} \mathbb{P}(H = h_j | \mathcal{L})$ where $\mathcal{L}$ denotes the observed labels. To achieve this, we will sequentially label new data examples that reveals information about $H$.

---

[2]In here and what follows $[n] := \{1, 2, \dots n\}$.

Given that accuracy is our chosen utility measure, we focus on the binary outcome of correct versus incorrect predictions made by $H$ among the candidate models $\mathcal{M}$ on these labeled examples. With this intention, we use a parameter that represents the probability of this binary outcome. Formally, we characterize the behavior of the best model $h^*$ by an error probability $\epsilon$ when predicting the true label as follows:

$$\mathbb{P}(H(x) \neq y | H = h^*) = \epsilon, \tag{1}$$
$$\mathbb{P}(H(x) = y | H = h^*) = 1 - \epsilon$$

where $\epsilon \in [0, 1]$. We also assume that the true (unobserved) labels $\{Y_i \, | i \in [n]\}$ are conditionally independent given $H$. Thus, they can be interpreted as generated according to the best model $h^*$, and flipped with probability $\epsilon$ independently at random. We learn $\epsilon$ prior to model selection process, as detailed in Section 3.3.

Our primary motivation for this formulation is to establish a highly compact yet effective and interpretable relationship between the best model and the true labels. Although this problem setting closely aligns with the one studied by Chen et al. (2015), which inspired our approach, representing all class conditional relationships with a single parameter is a major simplification of the problem. Next, we introduce the MODEL SELECTOR algorithm and explain how it uses Equation 1 to select informative examples to label, followed by algorithmic details concerning the choice of $\epsilon$.

## 3.2   The Algorithm

Given an unlabeled data pool $\mathcal{D}$, we want to identify the $b$ most informative examples to label for model selection. We characterize the information contained in the labels towards $H$ using Shannon's mutual information with respect to Equation 1.

Entropy, as a measure of uncertainty, given a discrete random variable $X$ distributed according to $\mathbb{P} : \mathcal{X} \to [0, 1]$ is defined as $\mathbb{H}(X) := -\sum_{x \in \mathcal{X}} \mathbb{P}(X = x) \log \mathbb{P}(X = x)$. The mutual information between $X$ and $Y$ measures the expected information gain that $Y$ provides about $X$, and is defined as: $\mathbb{I}(X; Y) = \mathbb{H}(X) - \mathbb{H}(X|Y)$.

Our objective can be formalized as finding the set of labeled data examples $\mathcal{L}_{\text{OPT}[b]}$ (of size at most $b$) that gives us maximum information about $H$. That is,

$$\mathcal{L}_{\text{OPT}[b]} := \underset{\substack{\mathcal{L} \subset \{(x_i, y_i) \, | i \in [n]\} \\ s.t. \; |\mathcal{L}| \leq b}}{\arg \max} \; \mathbb{I}(H; \mathcal{L}). \tag{2}$$

Starting from the initial pool of unlabeled examples $\mathcal{D}$, the strategy of MODEL SELECTOR is to greedily pick the unlabeled example that provides the maximal information gain about the true value of $H$ and request its label. At each greedy step $t$, MODEL SELECTOR queries the label of $x_t$ where:

$$x_t = \underset{x \in \mathcal{U}_t}{\arg \max} \; \mathbb{I}(H; Y | x, \mathcal{L}_t) \tag{3}$$

where $\mathcal{L}_t$ and $\mathcal{U}_t$ respectively denote the disjoint sets of labeled and unlabeled examples at step $t$.

To compute the information gain for the unlabeled data example $x$, we can express Equation 3 in terms of differential entropies as follows:

$$\begin{aligned} x_t &= \underset{x \in \mathcal{U}_t}{\arg \max} \; \mathbb{H}(H|\mathcal{L}_t) - \mathbb{E}_Y[\mathbb{H}(H|\mathcal{L}_t \cup \{(x, Y)\})] \\ &= \underset{x \in \mathcal{U}_t}{\arg \min} \; \mathbb{E}_Y[\mathbb{H}(H|\mathcal{L}_t \cup \{(x, Y)\})] \end{aligned} \tag{4}$$

where the expectation in $\mathbb{E}_Y[\cdot]$ is taken over $Y$, since the true label $y$ for the corresponding $x$ is unknown[3].

Equation 4 is equivalent to minimizing the model posterior uncertainty (Nguyen et al., 2021). It suggests that the most informative sampling policy for unlabeled data is equivalent to minimizing the entropy of the

---

[3]To compute this expectation, we approximate the posterior over the classes as uniform over the model predictions, motivated by its robust performance observed in the results.

posterior of $H$. To compute this entropy, we require the expected entropy of the model posterior given the example $x$ and its hypothetical label $c \in \mathcal{Y}$ as well as the labeled examples $\mathcal{L}_t$, which we refer to as the *hypothetical model posterior* at $t$ given $x$. By applying Bayes' rule, we obtain the following expression of the hypothetical model posterior:

$$\mathbb{P}(H = h_j | \mathcal{L}_t \cup \{(x, Y = c)\}) \propto \tag{5}$$
$$\mathbb{P}(\mathcal{L}_t \cup \{(x, Y = c)\}) | H = h_j) \, \mathbb{P}(H = h_j).$$

Consider a uniform prior over models with $P(H = h_j) = 1/m$ for simplicity. We have $\mathbb{P}(H = h_j | \mathcal{L}_t \cup \{(x, Y = c)\}) \propto \mathbb{P}(\mathcal{L}_t \cup \{(x, Y = c)\}) | H = h_j)$. Further applying the model in Equation 1 to $\mathbb{P}(\mathcal{L}_t \cup \{(x, Y = c)\}) | H = h_j)$, the hypothetical model posterior can be computed as:

$$\mathbb{P}(H = h_j | \mathcal{L}_t \cup \{(x, Y = c)\}) \propto \tag{6}$$
$$(1 - \epsilon)^{h_{j,(t,x)}} \epsilon^{t - h_{j,(t,x)}}$$

where $h_{j,(t,x)}$ denotes the number of correct predictions of classifier $h_j$ on $\mathcal{L}_t \cup \{(x, Y = c)\}$.

Upon selecting the most informative example $x_t$ at round $t$ using Equation 4, and receiving its label $y_t$ from the *oracle*, MODEL SELECTOR updates the labeled set $\mathcal{L}_t$ with $\mathcal{L}_{t+1} = \mathcal{L}_t \cup \{x_t, y_t\}$ and unlabeled set $\mathcal{U}_t$ with $\mathcal{U}_{t+1} = \mathcal{U}_t \backslash \{x_t\}$ as well as the model posterior with

$$\mathbb{P}(H = h_j | \mathcal{L}_{t+1}) \propto \tag{7}$$
$$\mathbb{P}(H = h_j | \mathcal{L}_t)(1 - \epsilon)^{\mathbb{1}[h_j(x_t) = y_t]} \epsilon^{\mathbb{1}[h_j(x_t) \neq y_t]}$$

It then selects another example to label from the pool of remaining unlabeled examples $\mathcal{U}_t$. After a number of these labels are requested up to the budget $b$, MODEL SELECTOR returns the model with the highest accuracy on the labeled set $\mathcal{L}_b$. The pseudocode of this algorithm is depicted in Algorithm 1.

---

**Algorithm 1** MODEL SELECTOR

**Require:** models $\mathcal{M} = \{h_1, \ldots, h_m\}$, unlabeled examples $\mathcal{U}_0$, parameter $\epsilon$, labeling budget $b$, *oracle*
1: $\mathcal{L}_0 \leftarrow \{\}$
2: **for** $t = 0$ to $b - 1$ **do**
3:     **for** $c \in \mathcal{Y}$ **do**
4:         $\mathbb{P}(H = h_j \mid \mathcal{L}_t \cup \{(x, Y = c)\}) \leftarrow \frac{1}{Z} \mathbb{P}(H = h_j | \mathcal{L}_t)(1 - \epsilon)^{\mathbb{1}[h_j(x_t) = c]} \epsilon^{\mathbb{1}[h_j(x_t) \neq c]}$     ▷ hypothetical model posterior
5:     **end for**
6:     $x_t \leftarrow \arg\min_{x \in \mathcal{U}_t} \mathbb{E}_Y[\mathbb{H}(H | \mathcal{L}_t \cup \{(x, Y)\})]$
7:     $y_t \leftarrow oracle(x_t)$
8:     $\mathcal{L}_{t+1} \leftarrow \mathcal{L}_t \cup \{(x_t, y_t)\}$
9:     $\mathcal{U}_{t+1} \leftarrow \mathcal{U}_t \backslash \{x_t\}$
10:    $\mathbb{P}(H = h_j | \mathcal{L}_{t+1}) \leftarrow \frac{1}{Z} \mathbb{P}(H = h_j | \mathcal{L}_t)(1 - \epsilon)^{\mathbb{1}[h_j(x_t) = y_t]} \epsilon^{\mathbb{1}[h_j(x_t) \neq y_t]}$    ▷ update model posterior
11: **end for**
12: **return** $\arg\max_{h_j \in \mathcal{M}} \frac{1}{b} \sum_{i \in \mathcal{L}_b} \mathbb{1}[h_j(x_i) = y_i]$    ▷ select the best model

---

### 3.3 Algorithmic Details

We determine the value of $\epsilon$ prior to the model selection process by conducting a grid search over a range of values and choosing $\epsilon$ that yields the best performance in terms of accurately identifying the best model (with details deferred to Section 4.3.1) *without requiring labels*. In many active learning scenarios, it is a standard practice to allocate an initial budget for exploration (Lewis, 1995; McCallum et al., 1998; Zhang & Chen, 2002; Hoi et al., 2006; Zhan et al., 2022), which typically involves randomly sampling some examples and querying their labels to obtain a rough estimate of dynamics of interest. This would yield an $\epsilon$ for our work. However, upon exploring this option, we discovered that $\epsilon$ can also be effectively learned by generating noisy labels using pretrained models. Specifically, we construct a noisy oracle, denoted as $\{\hat{y}_i\}_{i \in [n]}$, for the

unlabeled examples $\{x_i\}_{i\in[n]}$ by leveraging the distribution of predicted classes from the candidate models. For tasks with a limited number of classes, we label each example with the most frequently predicted class; for tasks with a larger set of classes, we sample a class based on this distribution. We then carry out our grid search using these noisy labels as if they were the true oracle, and choose the best-performing $\epsilon$.

For all datasets and model collections, we estimate $\epsilon$ for our problem without the need for any initial labeled data. Quantitatively, our estimation of $\epsilon$ has an error margin of only $0.01$ compared to the values obtained using the ground truth oracle. We refer to Appendix D for more details on this process.

According to Equation 1, an immediate expectation is that the ideal value for $\epsilon$ should reflect the the error rate of the best model for the target data, guiding us to the data examples where our epistemic uncertainty about the best model is highest. However, our experiments indicate that this can sometimes lead to overfitting to the evidence, possibly due to our single-parameter model in Equation 1 being a further simplification of the model proposed by Chen et al. (2015) and misaligned with their modeling assumptions. To illustrate this, consider a scenario in which the best model has an accuracy of $0.9$ on the target dataset, which implies that $\epsilon = 0.1$. In this case, at each step, the probability that a model $h_j \in \mathcal{M}$ is the best model, represented by the posterior $\mathbb{P}(H = h_j | \mathcal{L}_t)$, is scaled by a factor of $(1-\epsilon)/\epsilon$ if $h_j$ correctly predicts the label for $x_t$, as specified in Equation 7. For $\epsilon = 0.1$, this scaling factor becomes $9$, which heavily weights the evidence and leads to overfitting. Our grid search results support this observation: larger values of $\epsilon$ lead to more conservative updates to the model posterior, providing a regularization effect that mitigates overfitting to the evidence and improves overall performance.

The parameter $\epsilon$ can also be interpreted in terms of the model disagreement within the exploration-exploitation tradeoff. When $\epsilon$ is high, it yields better results in scenarios where the models exhibit high disagreement, encouraging exploration over exploitation to build sufficient confidence on the best model. In contrast, a lower $\epsilon$ increases the belief in the labeled examples when model disagreement is low. In this regard, the effectiveness of tuning $\epsilon$ using noisy labels is intuitive: in our simplified single-parameter model, $\epsilon$ serves as a proxy for measuring disagreement among competing models regarding the data labels. By learning $\epsilon$ directly from these noisy labels generated by model predictions, we can accurately gauge the extend of it without requiring any labeled data. Although these noisy labels alone may not reliably identify the best or nearly best model (as shown in Appendix D), they still offer valuable insights that guide MODEL SELECTOR in making more informed model selection decisions.

# 4 Experiments

We conduct a comprehensive set of experiments to evaluate the performance of MODEL SELECTOR for model selection. Our experiments span $8$ vision tasks and $8$ text classification tasks, each varying in number of classes. We test across 18 model collections with more than 1,500 pretrained models.

## 4.1 Datasets and Model Collections

For image classification tasks, we use $3$ ImageNet V2 datasets (Recht et al., 2019) with $1,000$ classes and $10,000$ examples. We download more than $100$ pretrained models for each dataset from PyTorch (2024), and further fine-tune them with different hyperparameters, resulting in varying accuracies. The model architectures range from ResNet (He et al., 2016) and MobileNet (Howard et al., 2017) to EfficientNet (Tan & Le, 2020), with accuracy ranging from $43\%$ to $87\%$.

We also consider the PACS dataset (Li et al., 2017), in order to test model selection in the domain adaptation setting, which includes four domains. We train $30$ models using the architectures mentioned above, resulting in a range of model accuracy from $73\%$ to $94\%$.

For text classification tasks, we add $8$ datasets from the GLUE benchmark (Wang et al., 2019), with test set sizes ranging from $71$ to $40,430$. We use pretrained models from HuggingFace (2024) without further fine-tuning. The model collections include architectures such as BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2020), and RoBERTa (Liu et al., 2019). The number of models in each collection varies from $80$ to $110$, and accuracy ranges from $5\%$ to $95\%$. The GLUE benchmark predicts $2$ or $3$ classes depending on the dataset.

Additionally, we use the same datasets and pretrained models as Karimi et al. (2021), who study the

online setting. The datasets we use include CIFAR-10 (Krizhevsky et al., 2009), ImageNet (Deng et al., 2009), Drift (Vergara, 2012), and EmoContext (SemEval, 2024).

We refer to Appendix A for additional details on datasets and model collections.

## 4.2   Baselines

To evaluate MODEL SELECTOR we compare against random sampling (RANDOM), as well as several adapted strategies listed below.

- **Uncertainty Sampling.** We adapt the method of Dagan & Engelson (1995) for model selection by creating a probability distribution over classes based on model predictions for each example. We then rank examples by their entropy and select the top $b$ example with the highest entropy.

- **Margin Sampling.** Adapted by Seung et al. (1992); Freund et al. (1997). The MARGIN method is non-adaptive and selects examples with the smallest margin, defined as the difference between the highest and second-highest class probabilities.

- **Active Model Comparison.** We adapt AMC from Sawade et al. (2012), which optimizes sampling to reduce the likelihood of selecting a worse model. While originally proposed for comparing two models within a labeling budget, we extend it to evaluate all pairs of pretrained models for best model selection.

- **Variance Minimization Approach.** We implement VMA from Matsuura & Hara (2023), which samples examples to minimize the variance of estimated risk. While VMA requires softmax predictions, our setup treats models as black boxes, so we substitute softmax predictions with a probability distribution based on model predictions.

Notably, none of these methods are explicitly designed for pretrained model selection with limited labels, yet we adapt them as baseline comparisons within this newly formalized model selection framework.

## 4.3   Experimental Setup

### 4.3.1   Evaluation Protocol and Learning $\epsilon$

We employ the following evaluation protocol. We uniformly sample $n$ i.i.d. instances from the entire test data. Each algorithm sequentially queries up to a labeling budget $b$ and selects the model with the highest accuracy on these $b$ labeled examples. We consider the model achieving the highest accuracy across all $n$ labels as the *true* best model. We then evaluate each method by comparing its selected model based on $b$ labels to the true best model. We call this process *realization* and repeat it multiple times to obtain a performance estimate for each method.

We learn the parameter $\epsilon$ for MODEL SELECTOR directly from the model predictions, as mentioned in Section 3.3. We first perform noisy labeling by assigning each data example the class most predicted by our models. For datasets with a large number of classes, we assign labels randomly based on the probability distributions formed by the model predictions. We then run our evaluation protocol treating our noisy labels as oracle labels and perform a grid search over different values of $\epsilon$. Finally, we select the $\epsilon$ that makes MODEL SELECTOR perform best in identifying the best model, solely with the noisy labels.

Our experiments yield several insightful observations regarding the selected values of the parameter $\epsilon$. For a comprehensive analysis of these findings, we refer to Appendix D.

### 4.3.2   Performance Metrics

For a given labeling budget, we compare different baselines using the following key performance metrics: *Identification Probability*, defined as the fraction of realizations where a method successfully identifies the true best model of that realization, *Label Efficiency*, the reduction in number of labels (%) required to select the best or a near-best model over all realizations, specifically one within $\delta$ vicinity of the best model, and *95th Percentile Accuracy Gap*, represents the 95th percentile of the accuracy gap across all realizations, which is measured relative to the accuracy of the best model in each realization.

## 4.4 Experimental Results

We present our numerical results for each of the previously introduced performance metrics. Extended results can be found in Appendix E. Our observations for each metric are summarized as follows:

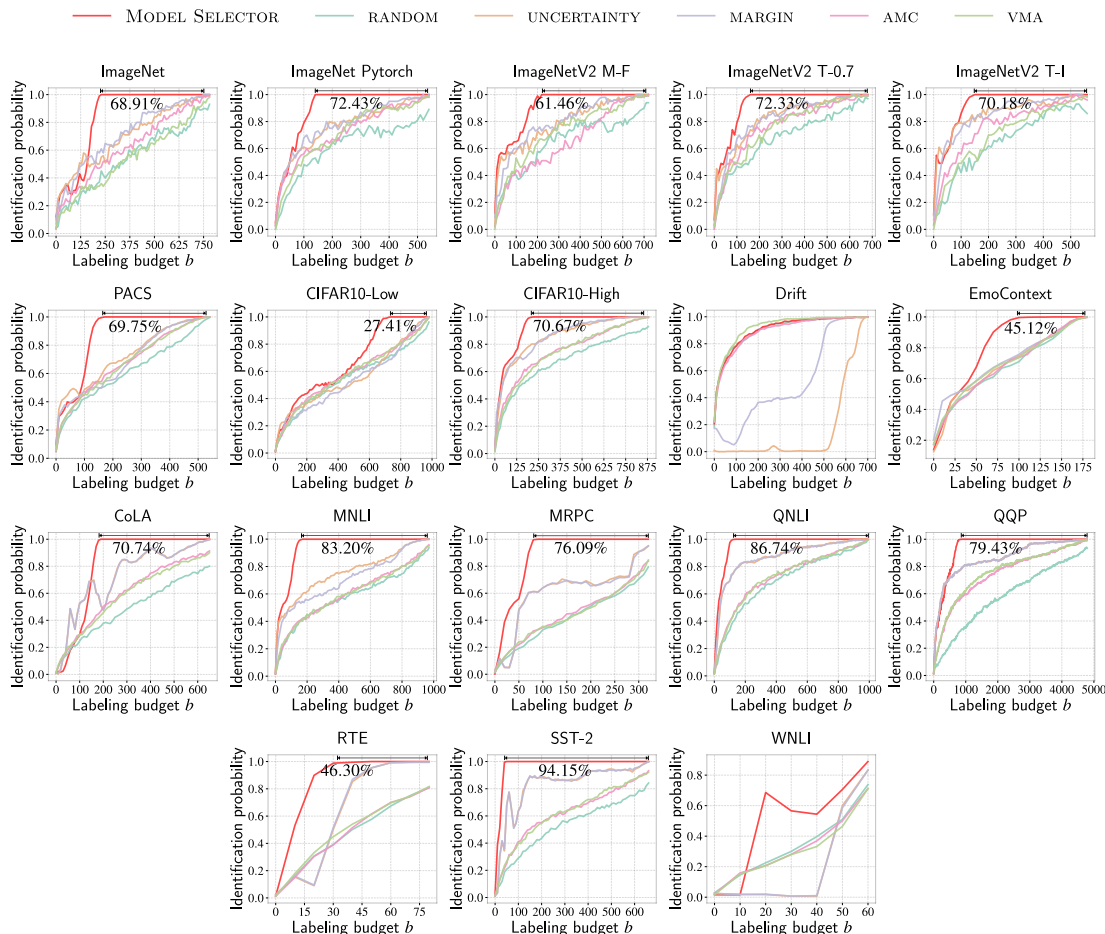### 4.4.1 Best Model Identification Probability



Figure 2: Best model identification probability of MODEL SELECTOR and the baselines on $18$ model collections. MODEL SELECTOR is capable of reducing the labeling cost by up to $94.15\%$ for identifying the best model.

Figure 2 shows identification probabilities of MODEL SELECTOR compared to baselines across 18 model collections, with labeling budgets extending until the best baseline (often UNCERTAINTY) reaches identification probability $100\%$ (see Appendix A for details). MODEL SELECTOR requires substantially fewer labels to reach a high identification probability than the baseline methods. For instance, our method reduces the labeling cost by up to $94.15\%$ to identify the true best model with $100\%$ identification probability, compared to the best competing baseline (primarily UNCERTAINTY). Compared to methods on ImageNet model collections, MODEL SELECTOR reduces the labeling cost by $68.91\%, 69.35\%, 70.56\%, 71.56\%$, and $74.38\%$ for MARGIN, UNCERTAINTY, AMC, VMA, and RANDOM, respectively. We also examine a challenging scenario using the model collections on the Drift dataset, where there is a significant distribution shift between the training data for each model in the collection, and the model performs very differently on the unlabeled data pool. In such a setting, RANDOM is considered the strongest baseline compared to selective sampling methods (Settles,

2009; Ayed & Hayou, 2023). By learning the parameter $\epsilon$ in a self-supervised manner from model predictions, MODEL SELECTOR automatically detects the drift and adjusts $\epsilon$ to 0.5, effectively mimicking random sampling in these settings. Additionally, MODEL SELECTOR outperforms existing baselines on the WNLI dataset using fewer than 70 examples, demonstrating its ability to identify the best model even on datasets with a limited number of labels. Across different labeling cost, MODEL SELECTOR can consistently match the identification probability of the baselines at a significantly lower labeling cost.

| Dataset | $\delta = 1\%$ | $\delta = 0.5\%$ | $\delta = 0.1\%$ |
|---|---|---|---|
| CIFAR10-High | ↓ **48.04**% | ↓ **58.40**% | ↓ **72.23**% |
| CIFAR10-Low | ↓ **21.07**% | ↓ **21.82**% | ↓ **25.67**% |
| EmoContext | ↓ **20.56**% | ↓ **34.19**% | ↓ **39.89**% |
| PACS | ↓ **62.73**% | ↓ **66.81**% | ↓ **68.62**% |
| Drift | ↑ 23.79% | ↑ 7.96% | ↑ 11.18% |
| ImageNet | ↓ **53.62**% | ↓ **63.80**% | ↓ **69.81**% |
| ImageNet Pytorch | ↓ **40.94**% | ↓ **64.07**% | ↓ **73.36**% |
| ImageNetV2 T-I | ↑ 6.12% | ↓ **49.12**% | ↓ **70.61**% |
| ImageNetV2 T-0.7 | ↓ **57.58**% | ↓ **57.79**% | ↓ **73.39**% |
| ImageNetV2 M-F | ↓ **48.18**% | ↓ **61.39**% | ↓ **56.72**% |
| MRPC | ↓ **72.41**% | ↓ **73.62**% | ↓ **74.54**% |
| CoLA | ↓ **45.89**% | ↓ **53.75**% | ↓ **71.01**% |
| QNLI | ↓ **46.88**% | ↓ **78.39**% | ↓ **85.75**% |
| QQP | ↑ 11.90% | ↓ **26.55**% | ↓ **73.36**% |
| SST-2 | ↓ **7.89**% | ↓ **39.66**% | ↓ **93.33**% |
| WNLI | 0.00% | 0.00% | 0.00% |
| MNLI | ↓ **69.42**% | ↓ **79.83**% | ↑ 3.95% |
| RTE | ↓ **40.96**% | ↓ **40.96**% | ↓ **40.96**% |

Table 1: Label efficiency for near-best models: MODEL SELECTOR consistently reduces labeling cost to reach the $\delta$ vicinity of the true best model compared to the best competing method.

### 4.4.2 Label Efficiency for Near-Best Models

We examine labeling efficiency from a different perspective. We evaluate the budget required to select the best *or* a near-best model with accuracy within the $\delta$ vicinity of that of the best model. Specifically, we measure the required number of labels where, in all realizations, the selected models are within 1%, 0.5%, and 0.1% of the best model accuracy.

Table 1 shows the percentage reduction in the number of labels required by MODEL SELECTOR to select a near-best model over all realizations. To reach the same $\delta$ vicinity of the accuracy of the best model, our method requires fewer labels than the best competing baselines (mainly UNCERTAINTY and MARGIN). Quantitatively, for MRPC, MODEL SELECTOR reduces the labeling cost by 72.41%, 73.62%, and 74.54% for $\delta$ values of 1%, 0.5% and 0.1%, respectively. Moreover, our method approaches the performance of the near-best method on ImageNetV2 for all model collections. Specifically, MODEL SELECTOR with $\delta = 0.1\%$ reduces the labeling cost by 70.61%, 73.39%, and 56.72% for ImageNetV2 T-I, ImageNetV2 T-0.7, and ImageNetV2 M-F, respectively. Our results show that MODEL SELECTOR is consistently more label-efficient, not only for identifying the best model but also for selecting near-best models across different settings.

### 4.4.3 Robustness Analysis

We compute the 95-th percentile of the accuracy gap to assess the robustness of MODEL SELECTOR and perform worst case scenario analysis for each method. Specifically, we calculate the accuracy gap between the selected

| Dataset Identification probability | MODEL SELECTOR (70%/80%/90%/100%) | RANDOM (70%/80%/90%/100%) | MARGIN (70%/80%/90%/100%) | UNCERTAINTY (70%/80%/90%/100%) | AMC (70%/80%/90%/100%) | VMA (70%/80%/90%/100%) |
|---|---|---|---|---|---|---|
| CIFAR10-High | **1.90/0.80/0.40/0.00** | 5.00/3.90/3.50/3.00 | 2.00/1.30/1.30/1.10 | 2.50/1.50/1.00/0.70 | 3.80/2.60/2.10/1.80 | 4.00/3.00/2.60/1.90 |
| CIFAR10-Low | 1.40/0.90/0.50/0.00 | 2.00/1.80/1.40/1.30 | 2.10/1.80/1.60/1.40 | 2.10/1.80/1.50/1.30 | 1.70/1.40/1.20/1.10 | 2.00/1.60/1.50/1.30 |
| EmoContext | 1.30/0.60/0.30/0.00 | 1.10/1.00/0.90/0.70 | 2.00/1.00/0.80/0.50 | 1.50/1.10/0.70/0.50 | 1.40/1.00/0.80/0.50 | 1.20/1.00/0.90/0.50 |
| PACS | 1.40/1.10/0.40/0.00 | 1.90/1.70/1.70/1.70 | 1.80/1.80/1.80/1.80 | 1.70/1.60/1.50/1.50 | 1.70/1.60/1.50/1.40 | 1.80/1.60/1.70/1.50 |
| Drift | 11.33/8.27/5.87/0.00 | 11.47/7.87/6.27/6.40 | 16.67/16.67/13.87/7.60 | 18.00/17.33/10.00/10.00 | 11.87/8.13/6.53/0.00 | 11.60/9.47/3.60/0.00 |
| ImageNet | 0.90/0.90/0.80/0.00 | 2.30/2.20/2.10/2.10 | 1.20/1.20/1.20/1.10 | 1.10/1.50/1.30/1.30 | 1.70/1.70/1.30/1.40 | 1.70/1.70/1.70/1.70 |
| ImageNet Pytorch | 0.80/0.50/0.20/0.00 | 3.70/3.30/3.00/2.60 | 1.30/0.90/0.80/0.70 | 1.00/1.00/1.00/0.80 | 2.20/1.90/1.30/1.20 | 3.60/2.40/1.90/1.20 |
| ImageNetV2 T-I | 1.20/0.70/0.10/0.00 | 4.30/4.50/3.00/2.20 | 1.30/1.30/1.10/0.50 | 1.70/1.40/0.60/0.60 | 3.50/2.80/1.90/1.80 | 3.10/2.40/2.30/1.60 |
| ImageNetV2 T-0.7 | 1.00/0.50/0.20/0.00 | 4.20/3.70/3.50/2.50 | 1.50/1.30/1.10/1.10 | 1.50/1.50/1.30/1.00 | 2.60/2.40/1.80/1.30 | 2.80/2.70/2.30/1.80 |
| ImageNetV2 M-F | 0.90/0.40/0.30/0.00 | 4.10/2.60/2.60/2.60 | 1.10/1.00/0.90/0.60 | 1.10/1.10/0.90/0.60 | 3.10/1.10/1.10/0.90 | 3.70/1.70/1.60/1.60 |
| MRPC | 1.14/1.14/0.29/0.00 | 5.71/5.43/5.14/4.86 | 2.00/1.43/1.14/0.86 | 1.71/1.14/0.86/1.14 | 5.43/5.14/5.14/4.86 | 5.43/5.14/4.86/4.29 |
| CoLA | 0.88/0.62/0.25/0.00 | 3.38/3.37/3.37/3.12 | 1.12/0.88/1.12/1.37 | 1.00/0.88/1.12/1.37 | 2.62/2.50/2.50/2.37 | 2.50/2.50/2.38/2.25 |
| QNLI | 1.00/0.60/0.30/0.00 | 4.60/4.20/3.90/3.80 | 2.10/1.40/1.00/0.80 | 2.40/1.40/1.00/0.80 | 4.60/4.00/3.80/3.60 | 4.40/4.20/3.90/3.60 |
| QQP | 0.46/0.24/0.12/0.00 | 1.50/1.44/1.36/1.30 | 0.40/0.30/0.26/0.22 | 0.38/0.30/0.26/0.24 | 1.08/0.96/0.80/0.72 | 1.10/0.90/0.80/0.72 |
| SST-2 | 0.40/0.27/0.13/0.00 | 6.80/6.40/6.27/6.40 | 0.40/0.40/0.40/0.40 | 0.40/0.40/0.40/0.40 | 5.87/5.60/5.60/5.47 | 5.73/5.47/5.33/5.33 |
| WNLI | 3.08/3.08/1.54/0.00 | 12.31/4.62/1.54/1.54 | 6.15/3.08/1.54/1.54 | 6.15/3.08/1.54/1.54 | 9.23/3.08/3.08/1.54 | 9.23/3.08/3.08/1.54 |
| MNLI | 1.00/0.80/0.40/0.00 | 4.70/4.30/3.90/2.90 | 1.20/1.20/1.10/1.00 | 1.10/1.10/1.10/1.00 | 4.40/4.00/3.20/2.70 | 4.00/4.30/3.70/2.90 |
| RTE | 10.40/10.00/4.40/0.00 | 22.40/21.20/16.40/11.20 | 21.20/16.80/17.20/6.80 | 20.80/16.80/19.20/7.20 | 22.80/20.80/16.80/10.80 | 22.00/19.20/14.80/10.80 |

Table 2: Robustness analysis: 95-th Percentile Accuracy Gap (%) at budget needed for MODEL SELECTOR to reach identification probability $70\%, 80\%, 90\%$, and $100\%$. Compared to baselines, MODEL SELECTOR achieves a smaller accuracy gap from the best model. Best method bolded; Next best underlined.

model and the true best model for all realizations and select the accuracy gap that is larger than $95\%$ of accuracy gaps across all realizations. We evaluate this for different budgets for each dataset, determined as the budget required for MODEL SELECTOR to achieve $70\%, 80\%, 90\%$, and $100\%$ identification probability.

As shown in Table 2, MODEL SELECTOR achieves significantly smaller accuracy gaps compared to baseline methods. For example, the best competing methods (MARGIN and UNCERTAINTY) on the RTE dataset with high disagreement among the model predictions, for identification probability of $70\%, 80\%$, and $90\%$, return a model with $20.80\%, 16.80\%$, and $17.20\%$ accuracy gaps, while MODEL SELECTOR returns models with accuracy gaps that are $10.40\%, 10.00\%$, and $0.40\%$. Quantitatively, these are $2\times, 1.7\times$, and $43\times$ smaller accuracy gaps. Compared to each method for MNLI at identification probability $90\%$, MODEL SELECTOR selects the model with $0.4\%$ accuracy gap, while MARGIN, UNCERTAINTY, AMC, VMA, and RANDOM select models with accuracy gaps of $1.1\%, 1.1\%, 3.2\%, 3.7\%$, and $3.9\%$, which is $2.8\times, 2.8\times, 8\times$, and $9.8\times$ larger compared to our method. Even in settings with a small number of examples, such as WNLI, and datasets with a large number of classes, such as ImageNet, MODEL SELECTOR outperforms the baselines. Specifically, when MODEL SELECTOR reaches an identification probability of $100\%$, the best competing baselines still select models with accuracy gaps of $1\%$ and $1.1\%$ for WNLI and ImageNet, respectively.

Our findings highlight the robustness of MODEL SELECTOR for consistently returning a near-best model even at its lowest performance.

## 5 Discussions

MODEL SELECTOR delivers competitive performance across various settings by capturing the discrepancy between the candidate models and the true labels with just a single parameter $\epsilon$. Additionally, it relies solely on hard predictions, without requiring access to the internal workings of pretrained models. Our future work will extend to settings where soft predictions are available and explore model selection with limited demonstrations for generative models.

## Acknowledgements

# References

Alnur Ali, Rich Caruana, and Ashish Kapoor. Active learning with model selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

AWS. Cloud computing services - amazon web services (aws), 2024. URL `https://aws.amazon.com/`.

Fadhel Ayed and Soufiane Hayou. Data pruning and neural scaling laws: fundamental limitations of score-based algorithms. *arXiv preprint arXiv:2302.06960*, 2023.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 65–72, 2006.

Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *International Conference on Computational Learning Theory*, pp. 35–50. Springer, 2007.

Jayanth Bhargav, Mahsa Ghasemi, and Shreyas Sundaram. Submodular information selection for hypothesis testing with misclassification penalties. *arXiv preprint arXiv:2405.10930*, 2024.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Marat Valievich Burnashev. Data transmission over a discrete channel with feedback. random transmission time. *Problemy peredachi informatsii*, 12(4):10–30, 1976.

Venkatesan T Chakaravarthy, Vinayaka Pandit, Sambuddha Roy, Pranjal Awasthi, and Mukesh Mohania. Decision trees for entity identification: Approximation algorithms and hardness results. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 53–62, 2007.

Aditya Chattopadhyay, Benjamin David Haeffele, Rene Vidal, and Donald Geman. Performance bounds for active binary testing with information maximization. In *Forty-first International Conference on Machine Learning*.

Yuxin Chen, S Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Conference on Learning Theory*, pp. 338–363. PMLR, 2015.

Herman Chernoff. *Sequential design of experiments*. Springer, 1992.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4774–4778. IEEE, 2018.

Ian Connick Covert, Wei Qiu, Mingyu Lu, Na Yoon Kim, Nathan J White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In *International Conference on Machine Learning*, pp. 6424–6447. PMLR, 2023.

Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pp. 150–157. Elsevier, 1995.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17, 2004.

Sanjoy Dasgupta and J Langford. Active learning. *Encyclopedia of Machine Learning*, pp. 6, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Amol Deshpande, Lisa Hellerstein, and Devorah Kletenik. Approximation algorithms for stochastic boolean function evaluation and stochastic submodular set cover. In *Proceedings of the twenty-fifth annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1453–1466. SIAM, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL `https://arxiv.org/abs/1810.04805`.

Zixin Ding, Si Chen, Ruoxi Jia, and Yuxin Chen. Learning to rank for active learning via multi-task bilevel optimization. *arXiv preprint arXiv:2310.17044*, 2023.

Zixin Ding, Si Chen, Ruoxi Jia, and Yuxin Chen. Learning to rank for one-round active learning. 2024.

Nicolas Emmenegger, Mojmir Mutny, and Andreas Krause. Likelihood ratio confidence sets for sequential decision making. *Advances in Neural Information Processing Systems*, 36, 2024.

Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 2013.

Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28:133–168, 1997.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pp. 1183–1192. PMLR, 2017.

Jacob Gardner, Gustavo Malkomes, Roman Garnett, Kilian Q Weinberger, Dennis Barbour, and John P Cunningham. Bayesian active model selection with an application to automated audiometry. *Advances in neural information processing systems*, 28, 2015.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. Efficient active learning of halfspaces: an aggressive approach. In *International Conference on Machine Learning*, pp. 480–488. PMLR, 2013.

GoogleCloud. Google cloud: Automl, 2024. URL `https://cloud.google.com/automl`.

Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 353–360, 2007.

Steve Hanneke and Liu Yang. Minimax analysis of active learning. *J. Mach. Learn. Res.*, 16(1):3487–3602, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Steven CH Hoi, Rong Jin, and Michael R Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, pp. 633–642, 2006.

M Horstein. Sequential transmission using noise-less feedback. *IEEE Trans. on Information Theory*, 12: 448–455, 1966.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. URL `https://arxiv.org/abs/1704.04861`.

Jonas Hübotter, Bhavya Sukhija, Lenart Treven, Yarden As, and Andreas Krause. Information-based transductive active learning. *arXiv preprint arXiv:2402.15898*, 2024.

HuggingFace. Huggingface: Natural language processing tools, 2024. URL https://huggingface.co.

Jonas Hübotter, Bhavya Sukhija, Lenart Treven, Yarden As, and Andreas Krause. Transductive active learning: Theory and applications, 2024. URL https://arxiv.org/abs/2402.15898.

Haim Kaplan, Eyal Kushilevitz, and Yishay Mansour. Learning with attribute costs. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pp. 356–365, 2005.

Mohammad Reza Karimi, Nezihe Merve Gürel, Bojan Karlaš, Johannes Rausch, Ce Zhang, and Andreas Krause. Online active model selection for pre-trained classifiers. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 307–315. PMLR, April 2021.

Parnian Kassraie, Nicolas Emmenegger, Andreas Krause, and Aldo Pacchiano. Anytime model selection in linear bandits. In *Proc. Neural Information Processing Systems (NeurIPS)*, December 2023.

Namit Katariya, Arun Iyer, and Sunita Sarawagi. Active evaluation of classifiers on large datasets. In *2012 IEEE 12th International Conference on Data Mining*, pp. 329–338, 2012. doi: 10.1109/ICDM.2012.161.

Andreas Kirsch. Advancing deep active learning & data subset selection: Unifying principles with information-theory intuitions. *arXiv preprint arXiv:2401.04305*, 2024.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning, 2019. URL https://arxiv.org/abs/1906.08158.

S Rao Kosaraju, Teresa M Przytycka, and Ryan Borgstrom. On an optimal split tree problem. In *Workshop on Algorithms and Data Structures*, pp. 157–168. Springer, 1999.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, pp. 5753–5763. PMLR, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Anurag Kumar and Bhiksha Raj. Classifier risk estimation under limited labeling resources. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 3–15. Springer, 2018.

Rui Leite and Pavel Brazdil. Active testing strategy to predict the best classification algorithm via sampling and metalearning. In *ECAI 2010*, pp. 309–314. IOS Press, 2010.

David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pp. 13–19. ACM New York, NY, USA, 1995.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization, 2017. URL https://arxiv.org/abs/1710.03077.

Junfan Li, Zenglin Xu, Zheshun Wu, and Irwin King. On the necessity of collaboration in online model selection with decentralized data. *arXiv preprint arXiv:2404.09494*, 2024a.

Po-han Li, Oyku Selin Toprak, Aditya Narayanan, Ufuk Topcu, and Sandeep Chinchali. Online foundation model selection in robotics. *arXiv preprint arXiv:2402.08570*, 2024b.

Shen Liang, Yanchun Zhang, and Jiangang Ma. Active model selection for positive unlabeled time series classification. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 361–372, 2020. doi: 10.1109/ICDE48307.2020.00038.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Xuefeng Liu, Fangfang Xia, Rick L Stevens, and Yuxin Chen. Contextual active online model selection with expert advice. In *ICML2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*. ICML, 2022a.

Xuefeng Liu, Fangfang Xia, Rick L Stevens, and Yuxin Chen. Cost-effective online contextual model selection. *arXiv preprint arXiv:2207.06030*, 2022b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4): 590–604, 1992.

Omid Madani, Daniel J. Lizotte, and Russell Greiner. Active model selection, 2012. URL https://arxiv.org/abs/1207.4138.

Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1, 2020.

Mitsuru Matsuura and Satoshi Hara. Active model selection: A variance minimization approach. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023. URL https://openreview.net/forum?id=vBwfTUDTtz.

Andrew Kachites McCallum, Kamal Nigam, et al. Employing em and pool-based active learning for text classification. In *ICML*, volume 98, pp. 350–358. Citeseer, 1998.

Mohammad Naghshvar and Tara Javidi. Active sequential hypothesis testing. 2013.

Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. An information-theoretic framework for unifying active learning problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9126–9134, 2021.

Robert Nowak. Noisy generalized binary search. *Advances in neural information processing systems*, 22, 2009.

Vihari Piratla, Soumen Chakrabarti, and Sunita Sarawagi. Active assessment of prediction services as accuracy surface over attribute combinations. *Advances in Neural Information Processing Systems*, 34:23140–23151, 2021.

Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4051–4070, 2022.

PyTorch. Pytorch hub: Reproducible research and pre-trained models, 2024. URL https://pytorch.org/hub/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL `https://arxiv.org/abs/1910.01108`.

Christoph Sawade, Niels Landwehr, Steffen Bickel, and Tobias Scheffer. Active risk estimation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 951–958, 2010.

Christoph Sawade, Niels Landwehr, and Tobias Scheffer. Active comparison of prediction models. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper_files/paper/2012/file/92fb0c6d1758261f10d052e6e2c1123c-Paper.pdf`.

SemEval. Emocontext, 2024. URL `https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer`.

Burr Settles. Active learning literature survey. 2009.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL `https://arxiv.org/abs/1905.11946`.

TensorFlow. Tensorflow hub: A library for reusable machine learning modules, 2024. URL `https://www.tensorflow.org/hub`.

Lenart Treven, Cansu Sancaktar, Sebastian Blaes, Stelian Coros, and Andreas Krause. Optimistic active exploration of dynamical systems. *Advances in Neural Information Processing Systems*, 36:38122–38153, 2023.

Alexander B Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1): 135–166, 2004.

Alexander Vergara. Gas Sensor Array Drift at Different Concentrations. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5MK6M.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL `https://arxiv.org/abs/1804.07461`.

Yu Xia, Fang Kong, Tong Yu, Liya Guo, Ryan A. Rossi, Sungchul Kim, and Shuai Li. Convergence-aware online model selection with time-increasing bandits. In *The Web Conference 2024*, 2024a. URL `https://openreview.net/forum?id=2IwSOTWvXu`.

Yu Xia, Fang Kong, Tong Yu, Liya Guo, Ryan A Rossi, Sungchul Kim, and Shuai Li. Which llm to play? convergence-aware online model selection with time-increasing bandits. In *Proceedings of the ACM on Web Conference 2024*, pp. 4059–4070, 2024b.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2017. URL `https://arxiv.org/abs/1611.05431`.

Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B Chan. A comparative survey of deep active learning. *arXiv preprint arXiv:2203.13450*, 2022.

Cha Zhang and Tsuhan Chen. An active learning framework for content-based information retrieval. *IEEE transactions on multimedia*, 4(2):260–268, 2002.

Bin Zhao, Fei Wang, Changshui Zhang, and Yangqiu Song. Active model selection for graph-based semi-supervised learning. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1881–1884. IEEE, 2008.

# Appendix

## A   Datasets and Model Collections

Table 3 summarizes the details of the $18$ model collections used in Section 4. These collections vary in dataset size, realization pool size, range of pretrained model accuracies, number of pretrained models, and number of classes. The dataset sizes vary from as few as 71 examples for WNLI to $50,000$ for ImageNet. The number of predicted classes ranges from binary classification tasks in the GLUE benchmark to $1,000$ classes for ImageNet. We have experimented with settings involving as few as $8$ models for EmoContext up to $118$ models for WNLI. We choose the realization pool size to be practical from a practitioner's point of view; specifically, we avoid labeling the entire test dataset. For most datasets, the realization pool size is $1,000$. For model collections on ImageNet and ImageNetV2, we do not use larger realization pool sizes to enable comparison with baseline methods, which otherwise would cause out-of-memory exceptions and have long execution times (mostly VMA). For every model collection in Section 4 we evaluate for all budgets $b$ up to the size of the realization pool.

In Table 3 we also show the best found $\epsilon$ parameter across $1,000$ realizations for every model collection. Further details on selection of $\epsilon$ can be found in Appendix D.

In conclusion, Table 3 highlights the diverse range of settings considered in our experiments.

| Dataset | Best $\epsilon$ across 1,000 realizations | Dataset size | Realization pool size | Model accuracy | Number of models | Number of classes |
|---|---|---|---|---|---|---|
| CIFAR10-High | 0.47 | 10,000 | 1,000 | 55% - 92% | 80 | 10 |
| CIFAR10-Low | 0.47 | 10,000 | 1,000 | 40% - 70% | 80 | 10 |
| EmoContext | 0.47 | 5509 | 1,000 | 88% - 92% | 8 | 4 |
| PACS | 0.45 | 9991 | 1,000 | 73% - 94% | 30 | 7 |
| Drift | 0.50 | 3600 | 1,000 | 25% - 60% | 9 | 6 |
| ImageNet | 0.45 | 50,000 | 1,000 | 50% - 82% | 102 | 1,000 |
| ImageNet Pytorch | 0.45 | 50,000 | 1,000 | 55% - 87% | 114 | 1,000 |
| ImageNetV2 T-I | 0.46 | 10,000 | 1,000 | 58% - 89% | 114 | 1,000 |
| ImageNetV2 T-0.7 | 0.45 | 10,000 | 1,000 | 51% - 86% | 114 | 1,000 |
| ImageNetV2 M-F | 0.48 | 10,000 | 1,000 | 43% - 81% | 114 | 1,000 |
| MRPC | 0.37 | 408 | 350 | 31% - 91% | 95 | 2 |
| CoLA | 0.45 | 1043 | 800 | 14% - 87% | 109 | 2 |
| QNLI | 0.44 | 5463 | 1,000 | 16% - 90% | 90 | 2 |
| QQP | 0.47 | 40430 | 5,000 | 8% - 85% | 101 | 2 |
| SST-2 | 0.36 | 872 | 750 | 7% - 97% | 97 | 2 |
| WNLI | 0.47 | 71 | 65 | 8% - 59% | 118 | 2 |
| MNLI | 0.43 | 9815 | 1,000 | 5% - 91% | 82 | 3 |
| RTE | 0.39 | 277 | 250 | 41% - 88% | 87 | 2 |

Table 3: Summary of the $18$ model collections and datasets used in our experiments, including dataset sizes, realization pool sizes, ranges of pretrained model accuracies, numbers of pretrained models, and numbers of classes.

Figure 3 displays the accuracies of the models used in Section 4, evaluated on the entire dataset. Our experiments encompass model collections ranging from those where the majority of models perform equally well, such as QQP and CoLA, to settings where there is an equal distribution of high-performing and low-performing models, like CIFAR10 and ImageNet. We also consider settings where most models do not perform
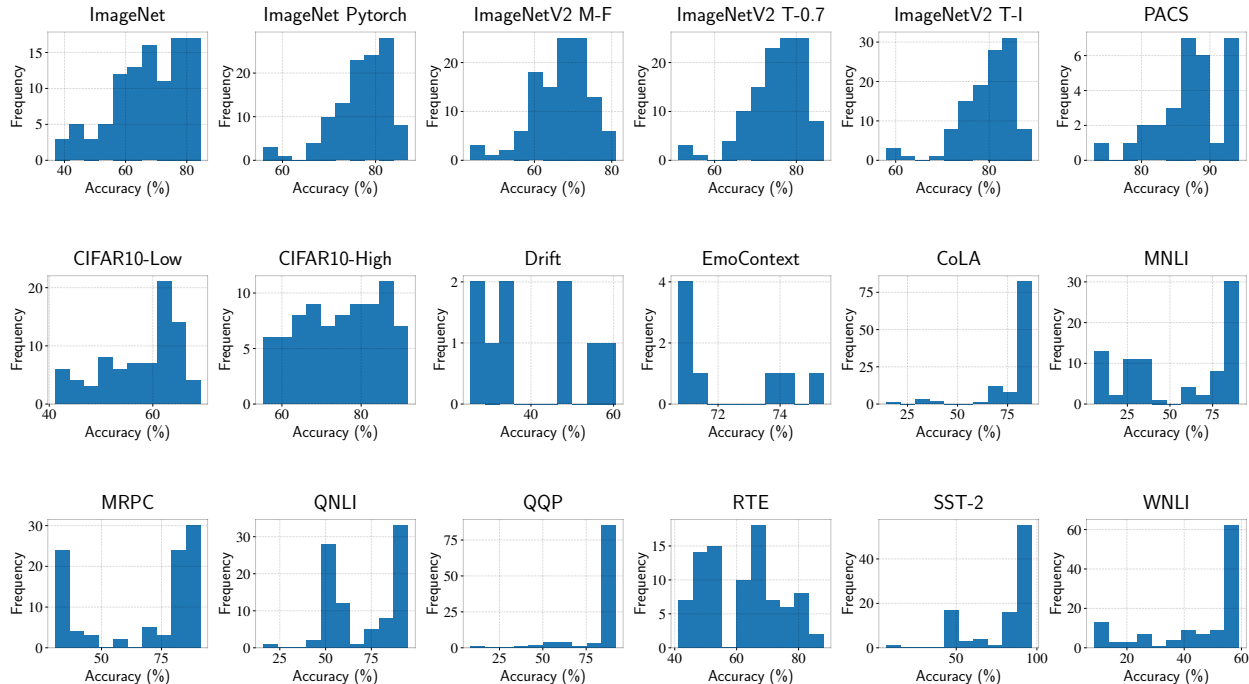
Figure 3: Accuracies of the models used in Section 4, evaluated on the entire dataset. Our experiments cover different scenarios across a wide range of model accuracies.

well and have high accuracy, such as Drift and RTE.

This demonstrates that we capture a wide range of possible model accuracies without making any assumptions about their distribution.

# B Baselines

To evaluate MODEL SELECTOR we compare against random sampling (RANDOM), as well as uncertainty sampling (UNCERTAINTY) (Dagan & Engelson, 1995) and margin sampling (MARGIN) (Seung et al., 1992; Freund et al., 1997). Additionally, we use the variance minimization approach (VMA) (Matsuura & Hara, 2023) and active model comparison (AMC) (Sawade et al., 2012).

**Uncertainty Sampling.** We adapt the method of Dagan & Engelson (1995) to our context. For each example, we create a probability distribution over the possible classes by counting predictions of all models. We sort the examples according to the probability distribution with maximal uncertainty, that is, entropy. Then we select $b$ examples with highest entropy. Note, that this is a non-adaptive baseline, meaning after querying a label it does not change the decision for the next example.

**Margin Sampling.** In a similar manner to UNCERTAINTY, we adapt the method for margin sampling, as proposed by Seung et al. (1992); Freund et al. (1997). The MARGIN method selects examples with the largest margin, where margin is defined as the difference between the highest probability of a certain class and the second-highest probability of a different class. As for UNCERTAINTY, it selects $b$ examples with largest margin in a non-adaptive manner.

**Active Model Comparison.** We implement the Active Comparison of Predictive Models method proposed by Sawade et al. (2012). AMC samples from a distribution that maximizes the power of a statistical test, and with that minimizes the likelihood of selecting a worse model. While AMC is originally designed to compare the risks of two predictive models within a fixed labeling budget, we extend the approach to evaluate all possible pairs of pretrained models, allowing for the selection of the best model within the given budget.

**Variance Minimization Approach.** We implement the algorithm proposed by Matsuura & Hara (2023). VMA samples examples such that the variance of the estimated risk is minimized. Although VMA is designed for active model selection, it assumes to have a test loss estimator. Specifcally, it requires softmax predictions for each example. However, in our setup models are treated as black boxes. Therefore, instead of softmax predictions we use probability distribution over the possible classes from counting predictions of all models.

It is important to note that none of these methods are specifically tailored for our setting of pretrained model selection with limited labels. Nonetheless, we adapt these methods as baseline comparisons within this newly formalized model selection framework.

## C   Related Work

**Optimal Information Gathering** How to gather information optimally has been studied from different angles and for different settings. The majority of works do not consider the model selection setting. For example, information gathering has been explored for active learning (MacKay, 1992; Dasgupta & Langford, 2011; Ding et al., 2023, 2024), experimental design (Lindley, 1956; Fedorov, 2013), reinforcement learning (Treven et al., 2023), evaluation of (stochastic) Boolean functions (Kaplan et al., 2005; Deshpande et al., 2014), active hypothesis testing (Chernoff, 1992; Nowak, 2009; Naghshvar & Javidi, 2013), channel coding with feedback (Horstein, 1966; Burnashev, 1976), and feature selection (Covert et al., 2023). Furthermore, Hübotter et al. (2024) discuss how to optimally gather information most efficiently. Smith & Gal (2018) explore modes of failure for measuring uncertainty in adversarial settings. Emmenegger et al. (2024) discuss gathering information in the sequential setting as we do, yet not specifically for model selection. In the context of minimizing epistemic uncertainty about an unknown target, several methods utilize the most informative selection policy using mutual information for active learning setting (Gal et al., 2017; Houlsby et al., 2011; Nguyen et al., 2021; Hübotter et al., 2024; Kirsch, 2024). Our work falls into this category. In particular, our motivation is to exploit the enjoyable properties of the most informative selection policy in the binary symmetric channel (Chen et al., 2015) for model selection under the uncertainty of the channel, where we consider the channel model as our epistemic uncertainty model. The provable near-optimality guarantees therein generally apply to our case. In instances not, our learned $\epsilon$ would be highest (simply $0.5$), indicating that simple random sampling should be the most competitive baseline. Furthermore, works can be categorized based on the assumptions of underlying noise into the noise-free settings (Freund et al., 1997; Kosaraju et al., 1999; Dasgupta, 2004; Chakaravarthy et al., 2007; Golovin & Krause, 2011) and works with noisy observations (Balcan et al., 2006; Hanneke, 2007; Gonen et al., 2013; Balcan et al., 2007; Tsybakov, 2004; Hanneke & Yang, 2015). Our work falls into the latter category. Many works have analyzed active learning with noisy observations, similarly to our setting with noisy prediction estimates for the pool-based setting. Yet, they limit their theoretical insights to limited hypotheses classess (Balcan et al., 2007; Gonen et al., 2013), and restricted noise settings (Tsybakov, 2004; Hanneke & Yang, 2015). In all of the previous studies, mutual information repeats itself as a key concept for both label-efficient model selection and optimal information gathering. Chattopadhyay et al. provide theoretical bounds on using greedy mutual information, and others show it in practice (Treven et al., 2023; Chen et al., 2015). In conclusion, our work falls into the category of works for sequential pool-based noisy information gathering by maximizing mutual information with provable near-optimality guarantees.

# D The Parameter $\epsilon$

In this chapter we present the results for choosing the $\epsilon$ parameter. We select the best $\epsilon$ as described in Section 3.3, that is, by constructing the noisy oracle and observing the identification probability for different $\epsilon$ values.

## D.1 Selecting the Parameter $\epsilon$

In Figure 5 we show the identification probability (see Section 4.4.1) of MODEL SELECTOR with different $\epsilon$ values on 18 model collections using the noisy labels. For comparison, in Figure 4 we show the same ranges of $\epsilon$ using the oracle labels. Both figures do not visualize the entire realization, and are cut at budgets for demonstrations. Figures 4 and 5 confirm our statement that appropriate values for $\epsilon$ lie in range $[0.35, 0.49]$. Smaller values than $\epsilon = 0.35$ lead to overfitting (see Section 3.3). When comparing model collections that use noisy labels with those that use oracle labels, we find that, fewer labels are needed to achieve a high identification probability at the same $\epsilon$ values. For example, $\epsilon = 0.45$ requires fewer labels to reach high identification probability than $\epsilon = 0.35$ for CIFAR10-Low in both Figure 5 and Figure 4. Also, we observe that for all ImageNet model collections MODEL SELECTOR with $\epsilon = 0.49$ requires more labels to reach high identification probabilities when compared to smaller $\epsilon$ values using both noisy and oracle labels. In addition, this experiment demonstrates the robustness of MODEL SELECTOR to different $\epsilon$ values. Quantitatively, QQP and QNLI demonstrate that all observed $\epsilon$ values $(0.35, 0.40, 0.45, 0.49)$ achieve high identification probability requiring approximately equal amount of labels from the entire realization.

Having identified the $\epsilon$ value ranges that reduce the labeling cost required to achieve the same identification probability, we perform the same experiment using more finely grained $\epsilon$ ranges. In Figure 6 and Figure 7 we show the identification probability of MODEL SELECTOR with finely grained $\epsilon$ values on 18 model collections using the noisy labels and oracle labels, respectively. Figures 6 and 7 further show the robustness of MODEL SELECTOR to $\epsilon$ values. Both from noisy labels and oracle labels one can see that MODEL SELECTOR with $\epsilon \in [0.45, 0.49]$ reaches high identification probability with approximately the same amount of required labels for both CIFAR10 model collections, CoLA, and QQP. The same holds for $\epsilon \in [0.41, 0.45]$ on MNLI and QNLI with both noisy and oracle labels.

As discussed in Section 4 for Drift model collection, MODEL SELECTOR automatically adjusts $\epsilon$ to $0.5$, effectively mimicking random sampling. MODEL SELECTOR is capable detecting that using only noisy labels.

When the figures do not clearly indicate which $\epsilon$ value has the highest reduction in labeling cost, we calculate it numerically to find the best $\epsilon$ values. Selected best $\epsilon$ values for every model collection using noisy labels and oracle labels can be found in Table 4. Quantitatively, our estimation of $\epsilon$ has an error margin of only $0.01$ compared to the values obtained using the oracle labels.

## D.2 Choosing $\epsilon$ with Initial Oracle Labels

As discussed in Section 3.3, it is a standard practice to allocate an initial budget for exploration (Lewis, 1995; McCallum et al., 1998; Zhang & Chen, 2002; Hoi et al., 2006; Zhan et al., 2022). In Figure 8 and Figure 9 we show how identification probability changes as we increase the number of oracle labels for RTE and WNLI, respectively. For both RTE and WNLI the identification probability differs when evaluated on noisy labels and on oracle labels. As the number of oracle labels increases, the identification probability increasingly resembles that obtained when evaluated entirely on oracle labels. However, the choice for best $\epsilon$ remains the same for both datasets when evaluated on noisy labels and on oracle labels, supporting the claim we do not any initial labeled data.
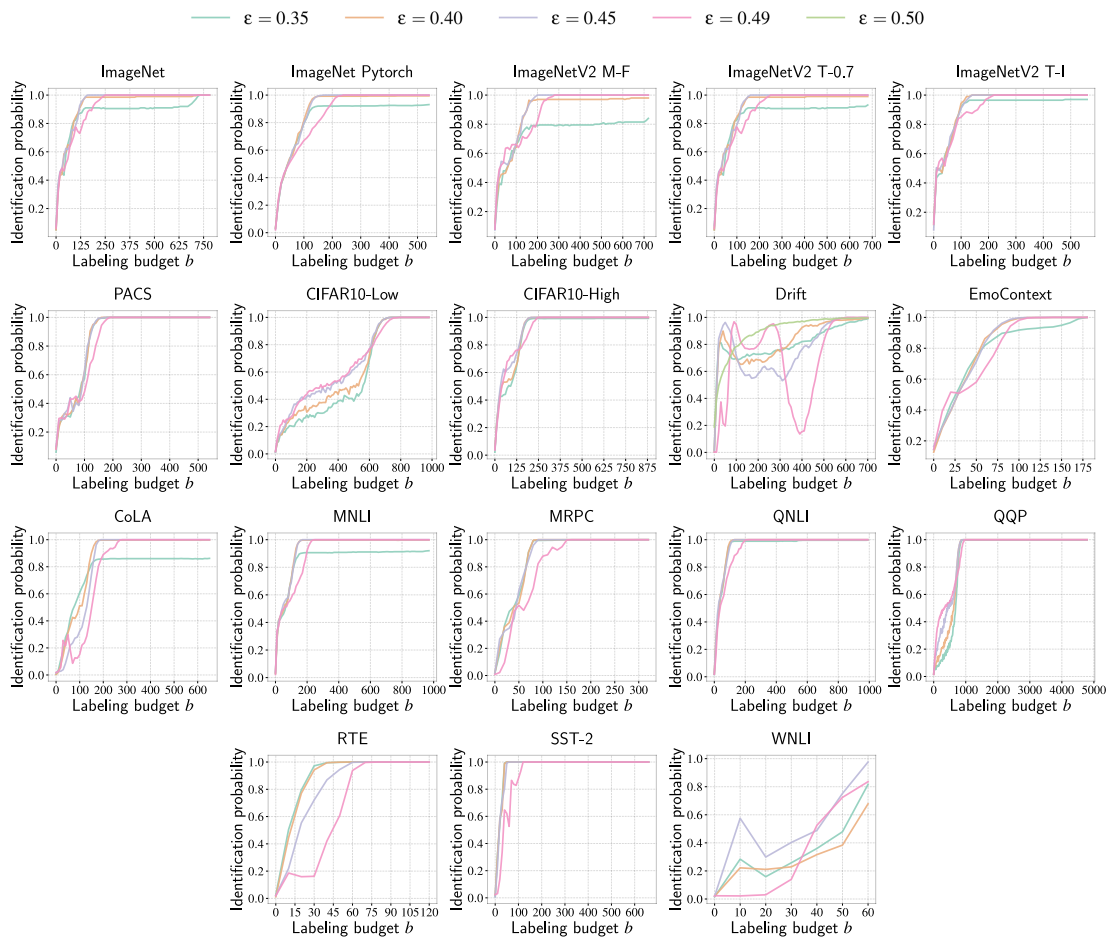
Figure 4: Best model identification probability of MODEL SELECTOR for $\epsilon \in \{0.35, 0.40, 0.45, 0.49, 0.50\}$ on $18$ model collections using oracle labels.
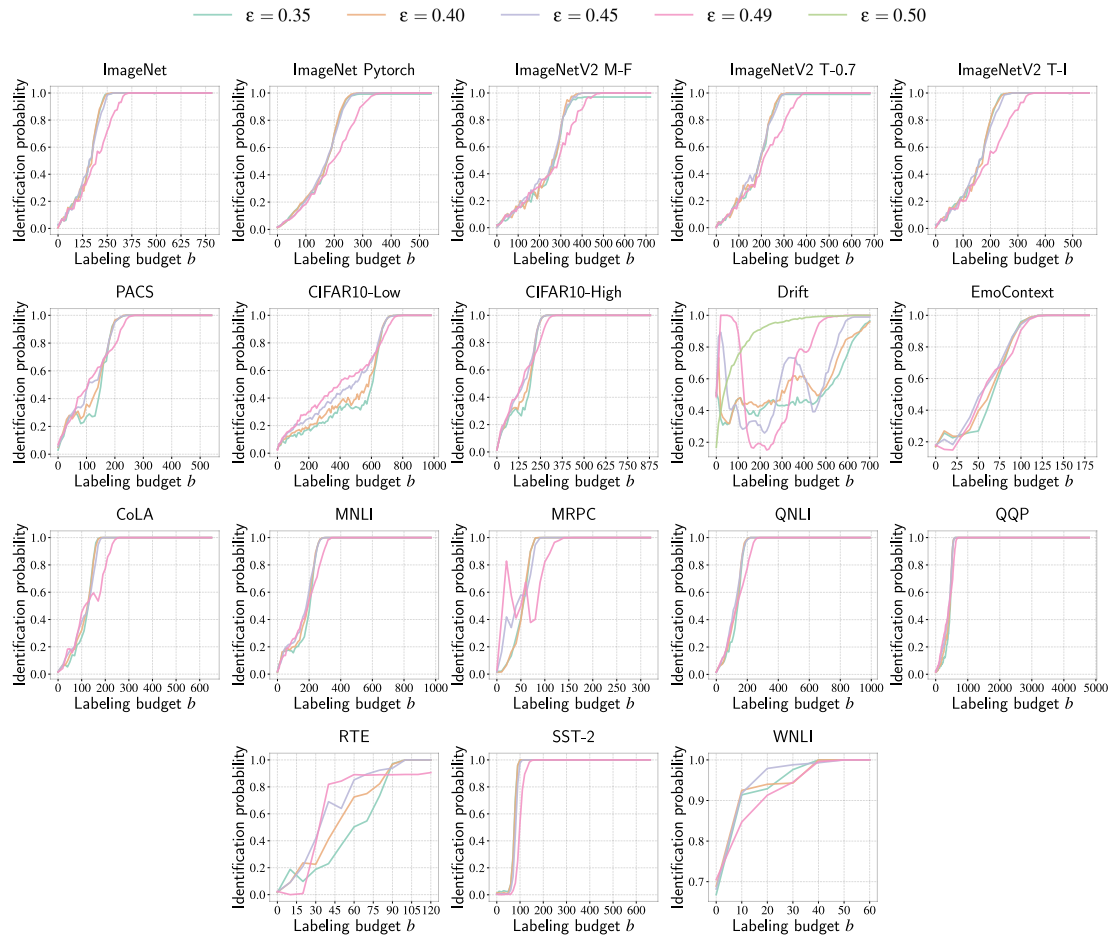
Figure 5: Best model identification probability of MODEL SELECTOR for $\epsilon \in \{0.35, 0.40, 0.45, 0.49, 0.50\}$ on 18 model collections using noisy labels.
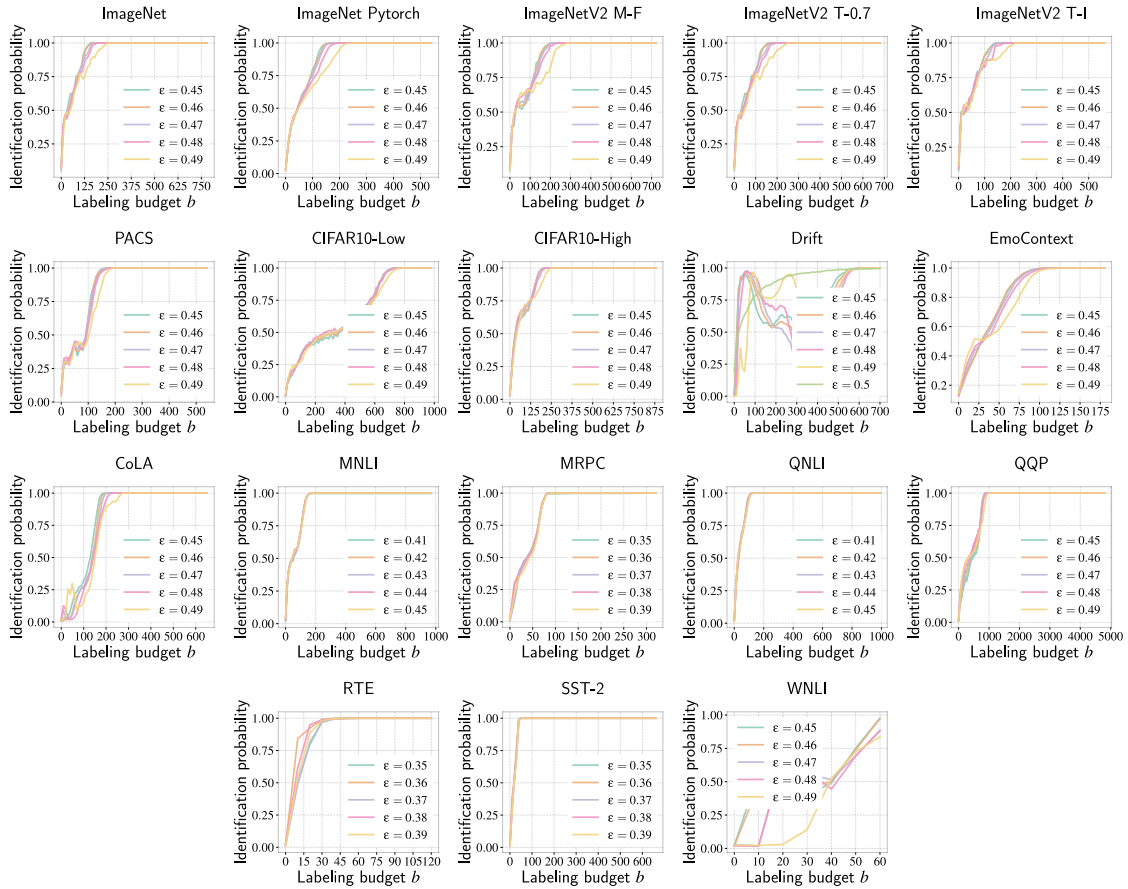
Figure 6: Best model identification probability of MODEL SELECTOR for finely grained $\epsilon$ values on $18$ model collections using oracle labels.
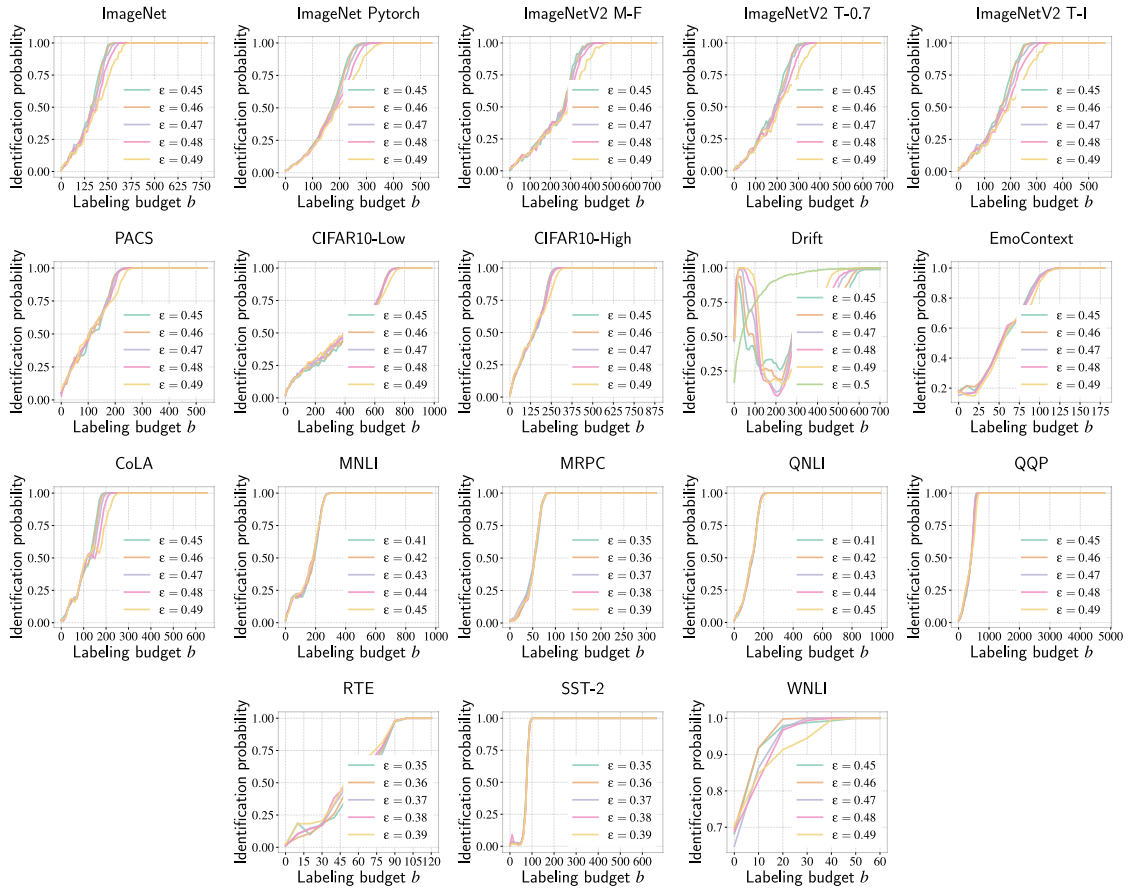
Figure 7: Best model identification probability of MODEL SELECTOR for finely grained $\epsilon$ values on $18$ model collections using noisy labels.

| Dataset | $\epsilon$ on noisy labels | $\epsilon$ on oracle labels |
|---|---|---|
| CIFAR10-High | 0.47 | 0.47 |
| CIFAR10-Low | 0.47 | 0.47 |
| EmoContext | 0.47 | 0.46 |
| PACS | 0.45 | 0.45 |
| Drift | 0.50 | 0.50 |
| ImageNet | 0.45 | 0.46 |
| ImageNet Pytorch | 0.45 | 0.46 |
| ImageNetV2 T-I | 0.46 | 0.45 |
| ImageNetV2 T-0.7 | 0.45 | 0.45 |
| ImageNetV2 M-F | 0.48 | 0.47 |
| MRPC | 0.37 | 0.36 |
| CoLA | 0.45 | 0.45 |
| QNLI | 0.44 | 0.45 |
| QQP | 0.47 | 0.47 |
| SST-2 | 0.36 | 0.36 |
| WNLI | 0.47 | 0.47 |
| MNLI | 0.43 | 0.44 |
| RTE | 0.39 | 0.39 |

Table 4: Selected $\epsilon$ using noisy labels compared to selected $\epsilon$ using oracle labels. Our estimation of $\epsilon$ has an error margin of only 0.01 compared to the values obtained using the ground truth oracle.
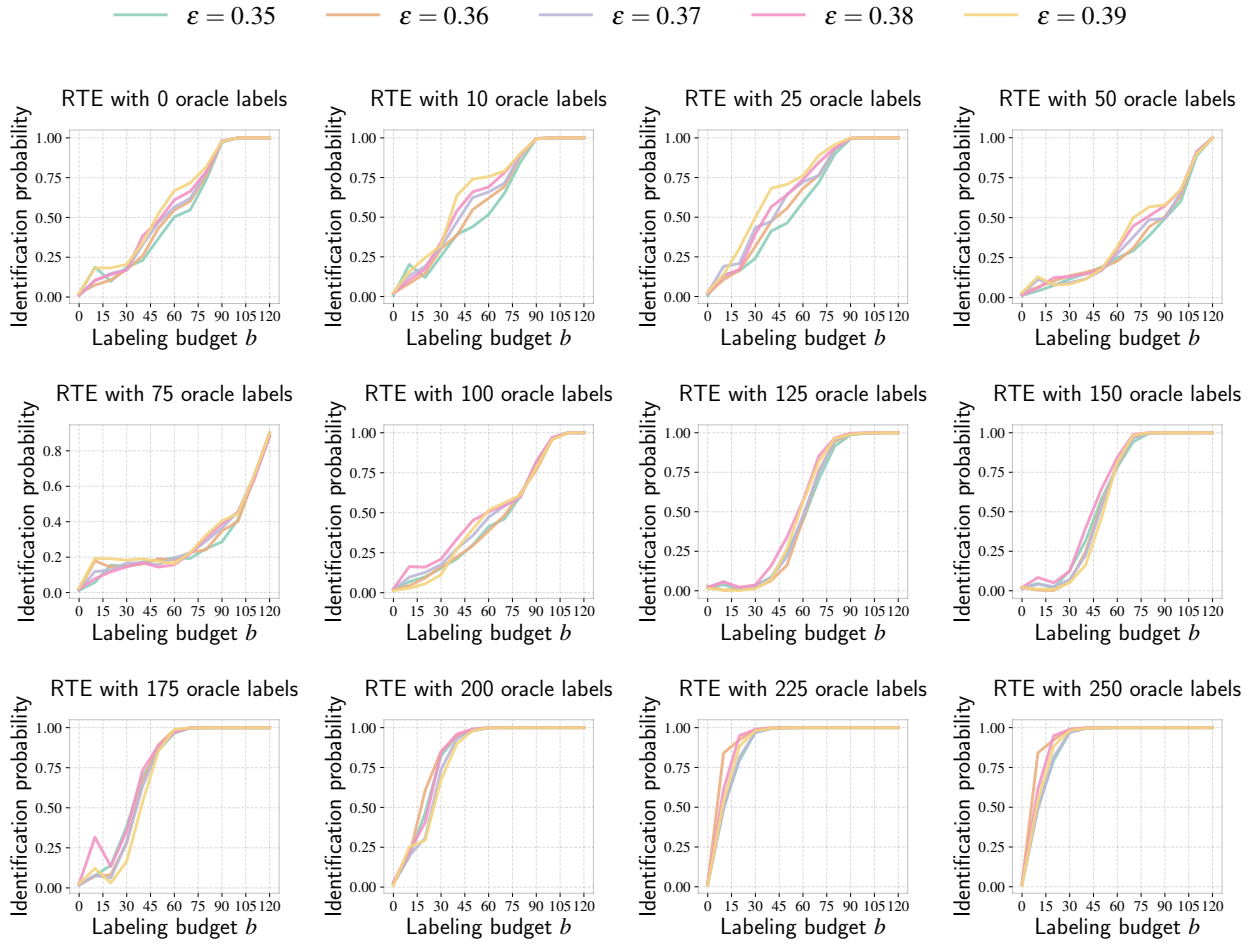
Figure 8: Best model identification probability of MODEL SELECTOR for $\epsilon \in [0.35, 39]$ on RTE dataset, as the number of oracle labels increases.
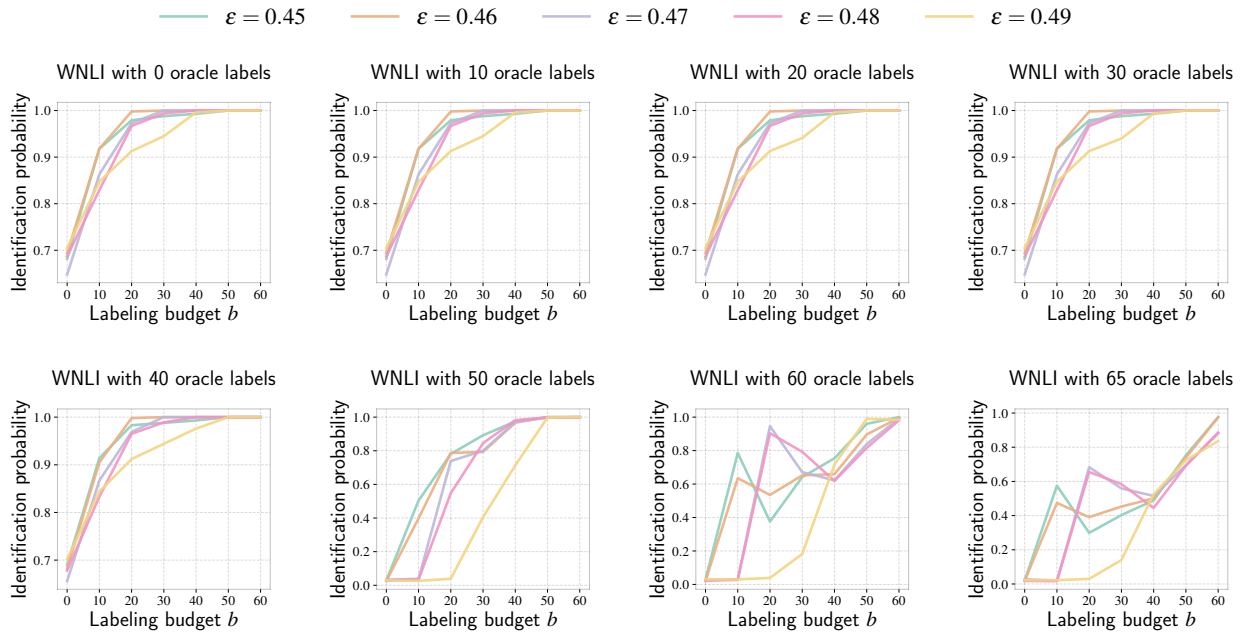
Figure 9: Best model identification probability of MODEL SELECTOR for $\epsilon \in [0.45, 49]$ on WNLI dataset, as the number of oracle labels increases.

### D.3 Can We Use Noisy Labels to Identify the Best Model?

Although we can use the noisy oracle, constructed as explained in Section 3.3, to choose the best $\epsilon$, we cannot use it to select the best model. Table 5 shows the accuracy gap between the best model evaluated using the noisy oracle and the true best model. For all model collections, this accuracy gap is significantly greater than $0\%$. For example, the gap reaches up to $15.93\%$ for RTE.

| Dataset | Best model accuracy gap |
|---|---|
| CIFAR10-High | 4.28% |
| CIFAR10-Low | 2.64% |
| EmoContext | 0.96% |
| PACS | 1.56% |
| Drift | 13.78% |
| ImageNet | 3.49% |
| ImageNet Pytorch | 4.47% |
| ImageNetV2 T-I | 4.53% |
| ImageNetV2 T-0.7 | 5.72% |
| ImageNetV2 M-F | 7.27% |
| MRPC | 1.29% |
| CoLA | 5.22% |
| QNLI | 3.25% |
| QQP | 1.08% |
| SST-2 | 3.93% |
| WNLI | 3.49% |
| MNLI | 3.48% |
| RTE | 15.93% |

Table 5: Accuracy gaps between the best models evaluated using the noisy oracle and the true best models across all model collections. Labels are required to identify the best model.

## E  Extended Results

In this section, we extend the results from Section 4.

Table 6 presents the label efficiency for near-best models, as explained in Section 4.4.2, considering larger values of $\delta$. To achieve accuracy within the same $\delta$ vicinity of the best model, our method requires fewer labels than the best competing baselines (mostly UNCERTAINTY and MARGIN). For ImageNetV2 T-0.7, MODEL SELECTOR reduces the labeling cost by $64.71\%$, $9.09\%$, and $16.67\%$ for $\delta$ values of $5\%$, $3\%$, and $2\%$, respectively. However, by examining the distribution of model accuracies in Figure 3, we observe that many models are within $5\%$, $3\%$, and $2\%$ of the best model's accuracy. Therefore, selecting a model within these vicinities cannot truly be considered selecting a near-best model, and we present these results for completeness. Importantly, as $\delta$ decreases, MODEL SELECTOR achieves greater reductions in labeling costs.

Table 7 extends the robustness analysis from Section 4.4.3. Instead of calculating the 95th percentile of the accuracy gap, we report the 90th percentile of the accuracy gap across all realizations. As previously, we evaluate this for different budgets for each dataset, determined as the budget required for MODEL SELECTOR to achieve $70\%$, $80\%$, $90\%$, and $100\%$ identification probability. As shown in Table 7, MODEL SELECTOR achieves significantly smaller accuracy gaps compared to baseline methods. For example, the best competing methods (MARGIN and UNCERTAINTY) on the RTE dataset with high disagreement among the model predictions, for identification probability of $70\%$, and $80\%$, return a model with $15.60\%$, and $14.40\%$ accuracy gaps, while MODEL SELECTOR returns models with accuracy gaps that are $8.40\%$, and $6.80\%$. Quantitatively, these are $1.9\times$ and $2.1\times$ smaller accuracy gaps. Furthermore, when MODEL SELECTORS 90th percentile accuracy gaps become $0\%$, the best competing baselines still select models with significant accuracy gaps. For example,

| Dataset | $\delta = 5\%$ | $\delta = 3\%$ | $\delta = 2\%$ |
|---|---|---|---|
| CIFAR10-High | ↑ 19.47% | ↓ **20.99**% | ↓ **25.81**% |
| CIFAR10-Low | ↓ **8.18**% | ↓ **15.79**% | ↓ **19.29**% |
| EmoContext | ↑ 16.67% | ↑ 27.27% | ↑ 5.06% |
| PACS | ↑ 17.39% | ↑ 27.16% | ↓ **49.80**% |
| Drift | ↑ 15.52% | ↑ 16.09% | ↑ 12.93% |
| ImageNet | ↓ **35.00**% | ↑ 67.35% | ↑ 0.62% |
| ImageNet Pytorch | ↑ 33.33% | ↑ 10.94% | ↑ 11.94% |
| ImageNetV2 T-I | ↑ 18.75% | ↓ **3.23**% | ↓ **18.75**% |
| ImageNetV2 T-0.7 | ↓ **64.71**% | ↓ **9.09**% | ↓ **16.67**% |
| ImageNetV2 M-F | ↑ 12.50% | ↓ **28.09**% | ↓ **13.48**% |
| MRPC | ↑ 2.90% | ↑ 5.33% | ↑ 2.60% |
| CoLA | ↑ 32.50% | ↓ **9.60**% | ↓ **41.99**% |
| QNLI | ↓ **18.97**% | ↓ **9.38**% | ↓ **42.52**% |
| QQP | ↑ 41.73% | ↑ 62.58% | ↑ 35.22% |
| SST-2 | ↑ 59.26% | ↑ 14.81% | ↓ **3.57**% |
| WNLI | ↓ **1.59**% | ↓ **1.59**% | 0.00% |
| MNLI | ↓ **29.27**% | ↑ 54.74% | ↑ 10.56% |
| RTE | ↓ **40.96**% | ↓ **40.96**% | ↓ **40.96**% |

Table 6: Label efficiency for near-best models: MODEL SELECTOR consistently reduces labeling cost to reach the $\delta$ vicinity of the true best model compared to the best competing method as $\delta$ decreases.

MARGIN and UNCERTAINTY select a model with 1.54% accuracy gap for CoLA, and 6.40% accuracy gap for RTE.

We confirm our findings highlight the robustness of MODEL SELECTOR in consistently returning the near-best model, even at its lowest performance.

| Dataset Identification probability | MODEL SELECTOR (70%/80%/90%/100%) | RANDOM (70%/80%/90%/100%) | MARGIN (70%/80%/90%/100%) | UNCERTAINTY (70%/80%/90%/100%) | AMC (70%/80%/90%/100%) | VMA (70%/80%/90%/100%) |
|---|---|---|---|---|---|---|
| CIFAR10-High | **1.00**/0.40/0.10/0.00 | 4.20/2.90/2.60/2.10 | <u>1.30</u>/<u>0.70</u>/0.80/0.80 | 1.50/0.90/<u>0.60</u>/<u>0.40</u> | 3.10/1.80/1.60/1.30 | 3.30/2.10/1.70/1.30 |
| CIFAR10-Low | **1.00**/0.50/0.10/0.00 | 1.40/1.20/1.00/0.90 | 1.60/1.40/1.30/1.00 | 1.60/1.40/1.20/0.90 | <u>1.30</u>/<u>1.00</u>/<u>0.80</u>/<u>0.70</u> | 1.40/1.10/1.00/0.80 |
| EmoContext | **0.60**/0.30/0.10/0.00 | <u>0.80</u>/0.70/0.60/0.40 | 1.00/<u>0.60</u>/<u>0.50</u>/<u>0.30</u> | 0.90/0.70/<u>0.50</u>/<u>0.30</u> | 0.90/<u>0.60</u>/<u>0.50</u>/0.40 | <u>0.80</u>/<u>0.60</u>/<u>0.50</u>/0.40 |
| PACS | **1.10**/0.60/0.10/0.00 | 1.50/1.40/1.30/1.30 | 1.50/1.50/1.50/1.50 | <u>1.30</u>/<u>1.30</u>/<u>1.20</u>/1.20 | 1.40/<u>1.30</u>/<u>1.20</u>/<u>1.10</u> | 1.50/1.40/1.30/1.20 |
| Drift | <u>8.67</u>/<u>6.40</u>/<u>2.93</u>/0.00 | 8.53/<u>6.40</u>/3.87/0.00 | 15.73/15.73/12.80/7.07 | 16.93/16.40/9.33/9.20 | 9.47/6.80/4.00/**0.00** | 9.60/5.87/**0.00**/**0.00** |
| ImageNet | **0.70**/0.70/0.20/0.00 | 2.00/1.60/1.70/1.70 | <u>0.80</u>/<u>0.80</u>/<u>0.80</u>/<u>0.50</u> | 0.90/1.10/1.10/0.90 | 1.30/1.30/1.10/1.10 | 1.40/1.40/1.50/1.40 |
| ImageNet Pytorch | **0.50**/0.30/0.20/0.00 | 2.80/3.00/2.50/2.10 | 1.00/<u>0.60</u>/<u>0.60</u>/<u>0.50</u> | <u>0.80</u>/0.80/0.70/<u>0.50</u> | 1.40/1.00/0.90/0.80 | 2.60/2.00/1.30/0.90 |
| ImageNetV2 T-I | **1.00**/0.30/0.00/0.00 | 3.50/3.20/2.20/1.80 | **1.00**/1.10/0.70/0.40 | 1.10/<u>0.90</u>/<u>0.50</u>/<u>0.20</u> | 2.70/2.20/1.80/1.20 | 2.40/1.60/1.80/1.00 |
| ImageNetV2 T-0.7 | **0.70**/0.30/0.00/0.00 | 3.80/2.90/2.90/1.90 | <u>1.10</u>/<u>0.90</u>/0.80/<u>0.60</u> | <u>1.10</u>/1.00/0.90/0.70 | 2.20/1.50/1.20/0.80 | 2.00/1.40/1.50/1.20 |
| ImageNetV2 M-F | **0.60**/0.30/0.10/0.00 | 3.30/1.50/1.50/1.10 | <u>0.90</u>/0.70/<u>0.50</u>/0.40 | <u>0.90</u>/<u>0.60</u>/0.60/<u>0.40</u> | 2.50/0.90/0.90/0.80 | 2.70/1.50/1.10/0.90 |
| MRPC | **0.86**/0.57/0.29/0.00 | 4.86/4.57/4.00/4.00 | <u>1.14</u>/<u>0.86</u>/<u>0.86</u>/<u>0.86</u> | <u>1.14</u>/<u>0.86</u>/<u>0.86</u>/<u>0.86</u> | 4.57/4.29/4.00/4.00 | 4.57/4.29/4.00/3.43 |
| CoLA | **0.62**/0.38/0.12/0.00 | 2.87/2.75/2.75/2.62 | <u>0.75</u>/<u>0.75</u>/<u>0.87</u>/1.12 | <u>0.75</u>/<u>0.75</u>/<u>0.88</u>/<u>1.12</u> | 2.25/2.25/2.12/2.00 | 2.12/2.13/2.00/1.88 |
| QNLI | **0.70**/0.40/0.10/0.00 | 4.00/3.70/3.50/3.30 | <u>1.40</u>/<u>0.90</u>/0.70/0.60 | 1.90/<u>0.90</u>/<u>0.60</u>/<u>0.60</u> | 4.20/3.60/3.30/3.20 | 3.90/3.60/3.30/3.00 |
| QQP | <u>0.32</u>/<u>0.14</u>/0.02/0.00 | 1.36/1.24/1.16/1.10 | **0.28**/<u>0.18</u>/<u>0.16</u>/<u>0.14</u> | **0.28**/<u>0.18</u>/<u>0.16</u>/<u>0.14</u> | 0.88/0.78/0.60/0.54 | 0.90/0.76/0.66/0.54 |
| SST-2 | **0.27**/0.13/0.00/0.00 | 5.87/5.73/5.20/5.20 | <u>0.40</u>/<u>0.40</u>/<u>0.40</u>/<u>0.40</u> | <u>0.40</u>/<u>0.40</u>/<u>0.40</u>/<u>0.40</u> | 4.53/4.53/4.13/4.13 | 4.40/4.27/4.13/4.00 |
| WNLI | **3.08**/1.54/1.54/0.00 | 9.23/<u>3.08</u>/1.54/<u>1.54</u> | <u>4.62</u>/**1.54**/1.54/0.00 | 6.15/**1.54**/1.54/0.00 | 6.15/<u>3.08</u>/1.54/<u>1.54</u> | 6.15/<u>3.08</u>/**1.54**/0.00 |
| MNLI | **0.70**/0.40/0.10/0.00 | 3.30/3.00/2.70/1.90 | <u>0.90</u>/<u>0.90</u>/0.90/<u>0.80</u> | 1.00/<u>0.90</u>/<u>0.80</u>/<u>0.80</u> | 3.20/2.90/2.20/1.80 | 3.20/3.30/2.60/2.10 |
| RTE | **8.40**/6.80/0.00/0.00 | 18.80/17.20/12.00/10.00 | 16.00/15.60/15.60/**6.40** | <u>15.60</u>/15.60/15.60/**6.40** | 19.20/15.60/11.60/10.00 | 18.40/<u>14.40</u>/<u>11.20</u>/10.00 |

Table 7: Robustness analysis: 90-th Percentile Accuracy Gap (%) at budget needed for MODEL SELECTOR to reach identification probability 70%, 80%, 90%, and 100%. Compared to baselines, MODEL SELECTOR achieves a smaller accuracy gap from the best model. Best method bolded; Next best underlined.

# F   Scaling and Computation Cost

All methods, including MODEL SELECTOR and the baseline algorithms, are implemented in Python and use the following Python libraries: Pandas, Matplotlib, Numpy, Scipy, Torch, HuggingFace, and Seaborn. To optimize runtime, we execute the realizations in parallel on a 128-core cluster.

The total runtime for the baseline methods across all realizations ranges from 60 seconds (for WNLI) to 30 hours (for ImageNet) on the cluster. MODEL SELECTOR execution times range from 1.1 seconds (for WNLI) to 61.2 minutes (for ImageNet).