

Consensus-Driven Active Model Selection

Justin Kay*
MIT

Grant Van Horn
UMass Amherst

Subhransu Maji
UMass Amherst

Daniel Sheldon
UMass Amherst

Sara Beery
MIT

Abstract

The widespread availability of off-the-shelf machine learning models poses a challenge: which model, of the many available candidates, should be chosen for a given data analysis task? This question of model selection is traditionally answered by collecting and annotating a validation dataset—a costly and time-intensive process. We propose a method for active model selection, using predictions from candidate models to prioritize the labeling of test data points that efficiently differentiate the best candidate. Our method, **CODA**, performs **consensus-driven active model selection** by modeling relationships between classifiers, categories, and data points within a probabilistic framework. The framework uses the consensus and disagreement between models in the candidate pool to guide the label acquisition process, and Bayesian inference to update beliefs about which model is best as more information is collected. We validate our approach by curating a collection of 26 benchmark tasks capturing a range of model selection scenarios. CODA outperforms existing methods for active model selection significantly, reducing the annotation effort required to discover the best model by upwards of 70% compared to the previous state-of-the-art. Code and data are available at: <https://github.com/justinkay/coda>.

1. Introduction

The availability of off-the-shelf machine learning models is growing rapidly. As of this writing there are over 1.9M pre-trained models available for download from the HuggingFace Models repository [22], ranging from small specialized models to large general-purpose foundation models. Application-specific *model zoos* are growing as well, curating sets of models for everything from wildlife monitoring [18] to medicine [8, 46]. These zoos potentially enable accurate data analysis without the need for custom ML development, but introduce a new challenge: **which model, of the many available, performs best for a given set of data?** Traditionally *model selection* decisions are made by collecting labels for a subset of the data in question and

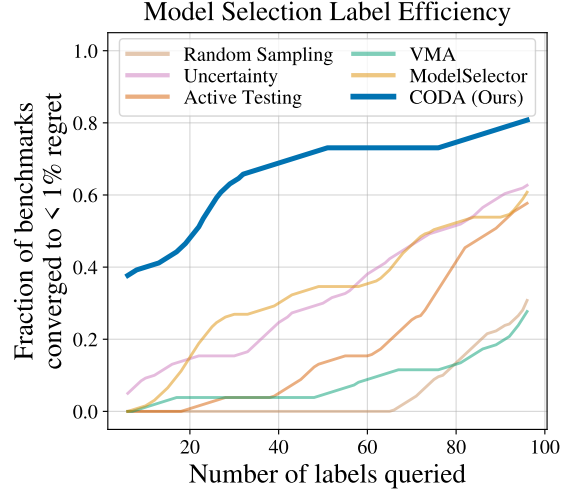


Figure 1. **We introduce CODA, a consensus-driven method for active model selection.** This figure shows the number of labels needed to converge to the optimal or near-optimal (within 1% accuracy) model in a benchmark suite of 26 model selection tasks. CODA is significantly more label-efficient than prior work, identifying a near-optimal model with fewer than 25 labeled examples over 50% of the time, and with fewer than 100 labeled examples over 80% of the time.

evaluating the performance of each model on that subset. To ensure results are robust, these datasets need to be large, representing significant human effort for each new dataset.

While reducing human effort during *training* has been well-studied [35, 59], efficient model selection at test time is relatively unexplored. Progress on this challenge will be beneficial for both users of pre-trained models and for researchers designing label-efficient algorithms. In particular, the field of *unsupervised domain adaptation (UDA)* proposes to adapt algorithms to new data without any human labels whatsoever, yet successful UDA methods are highly dependent on the use of human-labeled validation sets for model selection [15, 24, 28, 29, 42]. This contradiction has motivated work in *unsupervised model selection* [43, 56, 70], but so far these methods have proven to be unreliable, especially in challenging real-world conditions [15, 24, 28, 42].

Recently, methods for *active model selection* have been

*Correspondence to: kayj@mit.edu

proposed to identify an optimal model from a candidate set with fewer labels than required by traditional fully-labeled validation [25, 40, 45, 58]. Active methods use model predictions to guide the label acquisition process, iteratively querying a human expert for labels on specific data points that are expected to be most informative. Though promising, prior work remains label-inefficient, often requiring several hundred to several thousand labels to reliably perform model selection [40, 45, 58]. There are two key limitations that lead to this inefficiency: (1) models are largely treated independently of each other, both before and during the label collection process, ignoring valuable information captured by model agreement and disagreement; (2) categories are also treated independently, ignoring correlations between data points that can be deduced from category-specific model errors.

In this paper we propose a novel **consensus-driven** active model selection method, **CODA**, to address these limitations. Our approach models relationships between classifiers, categories, and data points in order to make more informed label queries. To do this we revisit classical probabilistic models of the classification data generating process. Specifically, we propose a framework inspired by the Dawid and Skene model of annotator agreement [14, 47], whereby each classifier is represented by a *confusion matrix* that captures its per-category performance characteristics. We adapt this framework for active model selection by constructing a probabilistic estimate over which model is best that accounts for per-category classifier consensus and uncertainty. We then iteratively query for ground-truth labels on data points that are expected to provide maximal information about the probability that each model is the best.

We validate our approach by curating a benchmarking suite of 26 model selection tasks representing a variety of real-world use cases across computer vision and natural language processing, which we publish alongside our method to support future model selection research. Our approach exceeds, often significantly, the performance of the previous state-of-the-art on 18 out of these 26 tasks. In addition, our method is exceptionally label-efficient, often requiring fewer than 25 labeled examples to identify the best or near-best model (see Fig. 1).

In summary, our main contributions are the following:

1. We introduce CODA, a novel method for active model selection. CODA leverages model consensus and Bayesian inference to identify the most informative labels for performing model selection at test time.
2. We curate a benchmarking suite of 26 active model selection tasks to validate our approach and compare with prior work. We release the data publicly to support future research in active model selection.
3. We demonstrate that CODA achieves state-of-the-art performance on 18 out of 26 tasks in our benchmark.

Additionally, though not the main focus of our work, we show that CODA’s initialization routine allows us to match or exceed state-of-the-art *unsupervised* model selection results on 20 out of 26 tasks.

2. Related work

Model selection in machine learning is typically performed using a held-out “validation” set to select between different candidate algorithms, hyperparameters, and/or training checkpoints. Differences between training, validation, and test distributions create challenges for model selection [3, 27, 30]. While in some cases out-of-distribution accuracy has been observed to be highly linearly correlated with in-distribution accuracy [41, 51, 52], in others it has been observed to be uncorrelated or even negatively correlated [41, 57, 64], indicating that reliable model selection for a dataset of interest cannot in general be performed using a validation set from a different data distribution. This has implications both for pre-trained models sourced from model zoos, where little or nothing is known about the training data distribution, as well as for models trained on one distribution and deployed on another. Our experiments and benchmarking suite evaluate both settings.

Unsupervised model selection methods perform model selection without the use of test labels. These methods have largely been proposed in the context of *unsupervised domain adaptation*, where *unlabeled* test-domain data is available for training. Unsupervised validation methods in this setting typically compute a heuristic such as entropy based on model predictions on this unlabeled test-domain data, under the assumption that these measures are correlated with accuracy [43, 53, 56, 69]. Alternatively, methods may use the accuracy on labeled in-distribution examples, weighted by their “similarity” to out-of-distribution samples, as a proxy [70]. Unfortunately, many of these methods have been shown to be poorly or even negatively correlated with test-domain accuracy [15, 24, 42].

One key limitation of this family of methods is that they consider the predictions of individual model checkpoints in isolation from any other models being evaluated; in contrast, recent work has identified the possibility to utilize the predictions of all models concurrently to better estimate their individual performance [20, 60]. Our method also harnesses this consensus information, but uses a probabilistic framework to aggregate and update our beliefs about the prediction set over time.

Active learning and active testing methods intelligently select informative data points to reduce annotation effort for training and evaluating machine learning models [31, 32, 35, 44, 59]. Active testing is related to our setup but differs in that the goal is to estimate the test loss of one model, rather than select the best from a set of candidates. Existing methods function by constructing unbiased importance

sampling estimators of model performance [31]. While it is possible to perform active model selection by performing active testing concurrently for all models and selecting the one with the lowest loss estimate, we will demonstrate this is significantly less label-efficient than specific targeted strategies for active model selection.

Active model selection methods have focused predominantly on the *online* setting, where data points are observed in a stream [25, 26, 37]. In contrast, we focus on the *pool-based* setting where a static collection of unlabeled data is available from the beginning. Early work in the pool-based setting resembles work in active testing, using importance-weighted loss estimates for each model [40, 58]. As pointed out by Kossen et al. [32], these importance-sampling-based approaches exhibit high variance early in the sampling process, since metrics are computed solely from the labels collected so far. Practically, this means model selection remains unreliable until a significant number of annotations have been collected.

Recent work from Okanovic et al. [45] performs active model selection without importance sampling estimators by defining a simple single-parameter distribution over which model is best at any given time. At each time step, the posterior probability for each model being best is updated according to $\text{Posterior} = \frac{1-\epsilon}{\epsilon} \times \text{Prior}$ if the model gets the label correct, where ϵ is a hyperparameter determining the learning rate. This simple probabilistic approach has been shown to be more label-efficient than prior work, but still requires significant annotation effort to overcome both its uninformative priors and independence assumptions between data points. Our method addresses these limitations by constructing informative unsupervised priors and by modeling correlated errors across the test data pool.

Probabilistic models of agreement aggregate annotations created by a group of human annotators on a dataset. They do so by modeling a “data generating process” that describes how annotations are created according to latent random variables like per-annotator accuracy. The Dawid-Skene model [14] (which we describe in more detail in Sec. 4.1) is an early example that remains popular today. Initially proposed for aggregating the predictions of doctors regarding patient outcomes, it has since found success in cleaning crowd-sourced annotations [50, 68] and merging human- and AI-generated predictions [6, 63, 65]. Many extensions have been proposed that incorporate Bayesian inference [47, 48] or more complex data generating processes [7, 19, 68]. We extend the general latent variable framework for active model selection. We do not fit the model directly to predictions as in prior work; instead, we use the framework as a starting point and update it iteratively to incorporate actively-collected information.

3. Active model selection problem formulation

Models and data We assume that we have a hypothesis set $H = \{h_k\}_{k=1}^{|H|}$ consisting of candidate models from which we want to select. Each model has generated predictions on some unlabeled test set $D = \{x_i\}_{i=1}^{|D|}$ that we care about but cannot exhaustively annotate. We assume the predictive task is a C -way multi-class classification problem with $\hat{c}_{k,i} = \arg \max_c h_k(x_i)$, where $\hat{c}_{k,i} \in \{1, \dots, C\}$ and $h_k(x_i) \in [0, 1]^C$ are the class prediction and the C -dimensional prediction vector of model h_k on data point x_i , respectively. Our setup is agnostic to whether $h_k(x_i)$ is a soft score vector or one-hot labels.

Active data point selection At every time step t , we query a human expert for a single ground truth label y_i . The choice of which y_i to query for is up to the model selection algorithm. We partition D into disjoint unlabeled and labeled subsets: $D = D_U \cup D_L$. Once a point has been queried for a ground-truth label it moves from D_U to D_L .

Model selection and evaluation At each time step t , a model selection algorithm returns its choice $\hat{h}^{(t)}$ of the model it currently believes is best. To evaluate these choices we assume that we know the form of the loss function L that we would use to evaluate each model in H if we had test labels. Our true best model, h^* (the one we hope to select), is the one that minimizes this loss empirically over D :

$$h^* = \arg \min_{h \in H} \frac{1}{|D|} \sum_{i=1}^{|D|} L(h(x_i), y_i), \quad (1)$$

where y_i are the true labels corresponding to x_i . In this paper we focus on accuracy-based loss functions; extending our framework to additional metrics is an interesting direction for future work.

We evaluate the efficacy of model selection algorithms based on the *regret* incurred at each time step, defined as the difference in loss between our chosen model $\hat{h}^{(t)}$ and the true best model h^* :

$$\text{Regret}_t = \frac{1}{|D|} \sum_{i=1}^{|D|} \left[L(\hat{h}^{(t)}(x_i), y_i) - L(h^*(x_i), y_i) \right] \quad (2)$$

We also measure *cumulative regret* at each time step, defined as the sum of all previous regrets:

$$\text{Cuml. Regret}_t = \sum_{s=1}^t \text{Regret}_s \quad (3)$$

Note that we can only measure these values while benchmarking as it requires oracle access to ground truth labels.

4. Method

4.1. A Dawid-Skene model for classifier predictions

The Dawid-Skene (DS) model is a probabilistic representation of how human annotators generate predictions on a set

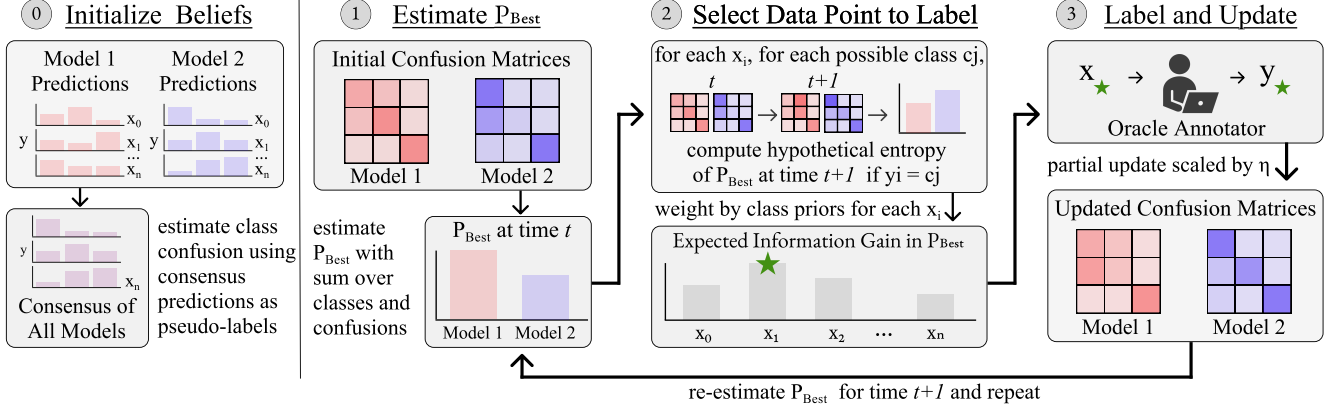


Figure 2. **CODA for active model selection.** Simplified example with two models. At time step 0 we instantiate a Bayesian framework for tracking model performance over time (Sec. 4.1), using the consensus of all model predictions to instantiate per-model priors (Sec. 4.2). At each time step we perform three actions: (1) We estimate P_{Best} , the current probability that each model is best, by integrating over our current beliefs (Sec. 4.3); (2) We compute the expected information gain in the P_{Best} distribution that would result from labeling each point in our dataset, and select the argmax as the most informative point (Sec. 4.4); (3) We query for the ground truth label of our selected data point, evaluate whether each model correctly predicts the true label, and update our beliefs (Sec. 4.5).

of data [14]. We adapt this model for the purpose of instead modeling the prediction process of machine learning models. Unlike prior work, we do not fit the model directly to the set of predictions; rather, we iteratively fit parameters to actively-collected ground-truth labels over time.

In particular, we base our approach off of the Bayesian instantiation of the DS model introduced by Passonneau and Carpenter [47]. We model the prediction process of each classifier h_k using a confusion matrix M_k of size (C, C) . Each row corresponds to a *true class* $c \in \{1, \dots, C\}$ and each column to a *predicted class* $c' \in \{1, \dots, C\}$. Thus each cell in the matrix represents the conditional probability

$$M_{k,c,c'} = P(\hat{c}_{k,i} = c' \mid y_i = c). \quad (4)$$

Our goal will be to perform statistical inference on the parameters of this confusion matrix. As such, the components of the data generating process are represented by random variables with latent parameters. The data generating process proceeds as follows:

1. Each data point’s true class label y_i is drawn randomly from per-data-point prior distributions over which class that data point could be, $y_i \sim \text{Cat}(\pi(x_i))$.
2. Each row of the classifier’s confusion matrix is drawn randomly from per-row distributions, $M_{k,c,\cdot} \sim \theta_{k,c}$, where $\theta_{k,c}$ is the prior distribution over what the row of the confusion matrix could be. To accommodate Bayesian updates, we initialize each $\theta_{k,c}$ to be a Dirichlet prior.
3. The sampled true class indexes into the corresponding row of the classifier’s confusion matrix, M_{k,y_i} .
4. The classifier’s prediction for that data point is sampled from the distribution over that row’s cells, $\hat{c}_{k,i} \sim \text{Cat}(M_{k,y_i})$.

See the supplemental material Fig. 9 for an illustrative view.

4.2. Constructing consensus priors (Fig. 2, Step 0)

We begin by collecting each model’s prediction vectors over the unlabeled dataset D . We take advantage of the “wisdom of the crowd” (of classifiers) to form initial consensus labels by summing these probabilities across all models,

$$s_{i,c} = \sum_{k=1}^H h_{k,c}(x_i) \quad \forall c = 1, \dots, C,$$

and then define the consensus label $c_i^* = \arg \max_c s_{i,c}$. For each model h_k , we then compare its predictions $h_k(x_i)$ against the consensus labels c_i^* for all i to initialize empirical confusion matrices:

$$\hat{M}_{k,c,c'} = \sum_{i=1}^{|D|} (\mathbf{1}[c_i^* = c] \cdot h_{k,c'}(x_i)) \quad (5)$$

We then use these empirical estimates to create Dirichlet priors θ over our beliefs in each row as:

$$\theta_{k,c,c'} = (\beta_{c,c'} + \alpha \hat{M}_{k,c,c'}) / T, \quad (6)$$

$$\beta_{c,c'} = \begin{cases} 1, & \text{if } c' = c, \\ \frac{1}{C-1}, & \text{otherwise.} \end{cases} \quad (7)$$

Where α is a hyperparameter controlling a blend between our empirical estimates and a static prior β representing 50% macro-accuracy, and T is a temperature parameter controlling the number of initial “pseudo-counts”. We use $\alpha = 0.1$ and $T = 0.5$ by default for all experiments.

4.3. Computing P_{Best} (Fig. 2, Step 1)

Throughout the active label collection process, we will create and update a probability distribution representing our belief in which model is best,

$$P_{\text{Best}} = (P(h = h^*))_{h \in H} \quad (8)$$

This distribution is the key component of our model selection algorithm. At each time step, we return $\arg \max_h P_{\text{Best}}$ as our choice of model, and (when benchmarking) evaluate our choice by computing the regret and cumulative regret of this choice at each time step.

We focus on accuracy as our target metric. In this case a simple option would be to use the posterior means of each classifier’s overall accuracy, computed by summing each row’s diagonal probability weighted by an estimated marginal class prevalence $\hat{\pi}(c)$ derived from the consensus:

$$\text{Acc}_{\text{mean}}(h_k) = \sum_{c=1}^C \hat{\pi}(c) M_{k,c,c} \quad (9)$$

$$\hat{\pi}(c) = \frac{1}{|D||H|} \sum_{i=1}^{|D|} \sum_{k=1}^{|H|} \sum_{c'=1}^C h_{k,c'}(x_i) M_{k,c',c} \quad (10)$$

Eq. (9) has the benefit of being efficient to compute but ignores the *probabilistic* nature of our estimates, failing to account for differences in uncertainty between classes/classifiers. Remember that we have access to more than just point estimates of the confusion matrices: we model our *beliefs* in what these confusion matrix entries *could* be based on what we have observed so far. We do this with per-row Dirichlet distributions, allowing us to incorporate additional uncertainty into accuracy estimates as follows. First, see that the marginal distribution for the c th class of the c th row’s Dirichlet distribution (*i.e.* the diagonal entry) is a Beta distribution with parameters:

$$\alpha_{k,c} = M_{k,c,c}, \quad \beta_{k,c} = \sum_{c' \neq c}^C M_{k,c,c'} \quad (11)$$

Then, to compute P_{Best} , we can integrate over the mixtures of all models’ per-row Beta distributions weighted by the class marginal $\hat{\pi}(c)$ as defined in Eq. (10). Supposing each classifier h_k ’s performance X_k is drawn independently from some distribution with PDF $f_k(x)$ and CDF $F_k(x)$, the integral that computes the probability that h_k ’s draw exceeds those of all others is:

$$P_{\text{Best}}(h_k) = P(X_k = \max_l X_l) \quad (12)$$

$$= \int_0^1 f_k(x) \prod_{l \neq k} F_l(x) dx, \quad (13)$$

Where the PDFs and CDFs are mixtures over the per-model per-row Beta distributions from Eq. (11):

$$f_k(x) = \sum_{c=1}^C \hat{\pi}(c) f_{k,c}(x) \quad (14)$$

$$F_l(x) = \sum_{c=1}^C \hat{\pi}(c) F_{l,c}(x) \quad (15)$$

where $f_{k,c}(x)$ is the PDF of $\text{Beta}(\alpha_{k,c}, \beta_{k,c})$ and $F_{l,c}$ is the CDF of $\text{Beta}(\alpha_{l,c}, \beta_{l,c})$.

Intuitively, to have X_k be the largest draw, we can “fix” X_k at some value x (with probability density $f_k(x)$), and then require that all X_l for $l \neq k$ lie at or below x (which happens with probability $F_l(x)$ each). See supplemental Fig. 10 for an illustrative view. In our implementation, we discretize $[0, 1]$ and approximate the above integral using a trapezoidal rule to integrate. Finally we obtain $P_{\text{Best}} = (P_{\text{Best}}(h_1), \dots, P_{\text{Best}}(h_{|H|}))$.

4.4. Selecting points to label (Fig. 2, Step 2)

To decide which point to label next at each time step, we aim to pick the one that, on average, reduces our uncertainty over P_{Best} the most. We quantify our uncertainty using the Shannon entropy $\mathcal{H}(P_{\text{Best}})$. For a candidate point x_i , let $\hat{\pi}(c | x_i)$ be the probability that x_i belongs to class c under our current beliefs (as in Eq. (10), without marginalizing). For each hypothetical label c , we perform a *virtual* update of all confusion matrix rows according to the update procedure defined in the next section (Sec. 4.5, Eq. (18)). This yields a *hypothetical* distribution P_{Best}^c . We measure the new entropy $\mathcal{H}(P_{\text{Best}}^c)$ and then revert to our original state. Weighting by $\hat{\pi}(c | x_i)$, the *expected posterior entropy* is

$$\sum_{c=1}^C \hat{\pi}(c | x_i) \mathcal{H}(P_{\text{Best}}^c). \quad (16)$$

Hence, the expected information gain (EIG) for point x_i is

$$\text{EIG}(x_i) = \mathcal{H}(P_{\text{Best}}) - \sum_{c=1}^C \hat{\pi}(c | x_i) \mathcal{H}(P_{\text{Best}}^c). \quad (17)$$

At each iteration, we compute $\text{EIG}(x_i)$ for all unlabeled points and query the label for the one with the highest EIG, then perform the *real* partial update to our Dirichlet parameters as in Eq. (18).

4.5. Updating beliefs (Fig. 2, Step 3)

As true labels become available, we update our confusion matrix estimates to incorporate new information. Consider we have just observed the label for x_i is $y_i = c$. Recall from Sec. 4.1 that we model each classifier’s class prediction for x_i as a random draw according to row c of its confusion matrix, $\hat{c}_{k,i} \sim \text{Cat}(M_{k,c})$. We use the fact that our Dirichlet prior for this row is the conjugate prior for this categorical distribution to update the Dirichlet for the next time step as:

$$\theta_{k,c,\hat{c}_{k,i}} \leftarrow \theta_{k,c,\hat{c}_{k,i}} + \eta \quad (18)$$

Where η is a learning rate hyperparameter allowing for partial updates. When $\eta = 1$ this reduces to the standard Dirichlet-categorical update rule. In practice, we find that partial updates with $\eta < 1$ are useful for stability. We use $\eta = 0.01$ by default for all experiments.

5. Datasets

Models are not re-trained during model selection. Therefore a *model selection benchmark* can be represented as a tuple (\mathbf{p}, \mathbf{y}) , where $\mathbf{p} \in \mathbb{R}^{|H| \times |D| \times C}$ is the set of predictions for each model $h \in H$, data point $x_i \in D$, and class $c \in C$, and $\mathbf{y} = \{y_1, \dots, y_{|D|}\}$ are the ground-truth labels for each data point in D . We perform model selection *directly on the test set*, i.e. there is no validation/test set split. This matches the pool-based setting of prior work in active testing and active model selection [31, 45].

We curate a suite of 26 diverse model selection benchmarking tasks from 3 different existing benchmarks along with over 3500 pre-trained models. This benchmark suite represents the largest empirical study of active model selection to date. In this section we describe the models in the candidate pool as well as how they were trained. In some cases, we train these models ourselves; in others, we source public pre-trained models for which we do not have any information about the training process. We publish our benchmark suite, both the curated set of datasets and the pretrained models, to support future research.

ModelSelector [45] is a benchmark suite focused on pre-trained models sourced from online repositories such as HuggingFace Models and PyTorch hub. We source the prediction files directly from the ModelSelector GitHub repository. We curate tasks for which at least 100 test data points are available for easy comparison with other datasets in our benchmarking suite. In total we include ten image and text based classification tasks: CIFAR10 (low accuracy models) [33], CIFAR10 (high accuracy models), PACS [34], and seven tasks from the GLUE language classification benchmark [67]. We refer to the vision tasks collectively as MSV (“ModelSelector Vision”). MSV and GLUE involve between 9–114 models per task totaling 851 models.

WILDS [30] is a benchmark of in-the-wild distribution shifts which provides an opportunity to study model selection in the domain generalization setting, where models have been specifically trained to generalize to new test data. We focus on all classification tasks in WILDS where performing standard model selection using the provided validation sets results in a regret greater than 1%, i.e. where active model selection would be beneficial (experiments included in supplemental). These tasks are: iWildCam [4], which involves classifying wildlife in imagery; FMoW [10], which involves classification of land use in remote sensing imagery; CivilComments [5], which involves toxicity classification of text data; and Camelyon17 [2], which involves tumor classification in histopathology data. These tasks range from binary to 182-way classification. We train all baseline algorithms from Koh et al. [30] using their publicly-available codebase: empirical risk minimization with in-distribution validation (ERM) [66], CORAL [62], IRM [1], and GroupDRO [21]. We train each method for the default

number of epochs used by WILDS, saving a checkpoint every epoch, resulting in between 20 and 240 models per task and 348 models total.

DomainNet126 [49, 55] is an unsupervised domain adaptation benchmark where the task is 126-way classification of objects in real and synthetic imagery. We follow Peng et al. [49] and construct 12 adaptation tasks across 4 domains: real imagery, paintings, clipart, and sketches, and use the standard UDA training protocol outlined used in prior work [42, 49, 55]. We use the Powerful-Benchmark codebase [43] to train 10 popular unsupervised domain algorithms on DomainNet126: ATDOC [36], BNM [11], BSP [9], CDAN [39], DANN [17], GVB [12], IM [61], MCC [23], MCD [54], and MMD [38]. We train each method for 40 epochs, saving a checkpoint every 2 epochs. In total, this gives us 10 algorithms \times 20 checkpoints = 200 models for each transfer task (2400 models overall).

6. Baselines

We compare our method against five other active model selection methods, ranging from classic approaches to recent state-of-the-art methods. The methods are:

Random sampling The simplest baseline samples a point uniformly at random each time step and maintains an empirical risk estimate for each model over time. We perform model selection by returning the model with the lowest empirical risk estimate at every time step.

Uncertainty sampling [13] We follow Okanovic et al. [45] to adapt classic committee-based uncertainty sampling techniques from active learning [13] to the active model selection scenario. At each time step, we greedily sample the point that most models in H disagree on, defined as the entropy of the mean prediction of all models. We perform model selection with empirical risk estimation.

Active Testing [31] Active Testing aims to obtain label-efficient unbiased estimates of model performance through importance-weighted sampling. To do so, they use a *surrogate model*, which is assumed to be more accurate than any candidate models in H , to guide the data sampling process. Points are sampled stochastically in proportion to the estimated loss of the model of interest with respect to the surrogate’s predictions. We adapt the framework to the active model selection setting as follows: we instantiate the surrogate model as the same ensemble we use to form our initial consensus estimates. We implement a naive extension of the Active Testing acquisition function that simply sums the acquisition probabilities from all models in the hypothesis set. We perform model selection with empirical risk estimation using unbiased risk estimators [16, 31].

VMA [40] VMA is an active model selection extension to the Active Testing framework. Their acquisition function is based on minimizing the pairwise variance of the difference between model loss estimates, where the loss estimates are the same importance-weighted estimates as [31]. Again we

	Task	Random Sampling	Uncertainty	Active Testing	VMA	Model Selector	CODA (Ours)
DomainNet26	real→sketch	147.1	197.7	269.8	119.2	88.8	<u>101.2</u>
	real→painting	143.6	167.9	118.9	139.9	<u>92.1</u>	87.2
	real→clipart	237.7	217.6	152.1	206.9	<u>153.2</u>	231.7
	sketch→real	280.4	252.7	<u>236.4</u>	273.4	268.4	11.9
	sketch→painting	<u>156.6</u>	181.9	237.7	157.1	173.9	13.0
	sketch→clipart	228.2	224.6	<u>162.0</u>	227.9	38.1	432.4
	painting→real	364.8	224.0	<u>215.6</u>	358.9	293.4	2.4
	painting→sketch	<u>179.3</u>	440.7	202.5	211.5	209.8	72.3
	painting→clipart	222.7	296.6	251.4	271.8	<u>73.2</u>	43.1
	clipart→real	322.1	177.1	159.1	306.4	<u>72.7</u>	25.3
	clipart→sketch	<u>247.2</u>	924.8	282.6	291.3	532.7	51.3
	clipart→painting	147.0	162.9	222.6	167.9	<u>131.7</u>	122.2
WILDS	iwildcam	287.0	380.4	392.1	440.6	459.0	201.7
	camelyon	<u>175.2</u>	311.6	206.1	160.1	198.3	288.7
	fmow	189.7	191.9	<u>153.0</u>	189.2	211.7	70.0
	civilcomments	140.9	13.3	76.7	<u>50.1</u>	125.3	318.6
MSV	cifar10-low	410.6	629.7	<u>399.9</u>	476.5	567.2	58.7
	cifar10-high	346.2	281.7	154.9	383.7	<u>90.9</u>	74.1
	pacs	216.6	6.9	101.8	116.5	97.9	<u>57.9</u>
GLUE	cola	368.1	169.5	239.8	317.8	<u>207.8</u>	2226.7
	mnli	237.4	<u>80.8</u>	234.2	312.8	148.5	23.5
	qnli	222.7	231.6	246.6	283.0	<u>185.7</u>	120.4
	qqp	136.7	388.9	<u>127.8</u>	169.2	186.8	4.8
	rte	375.7	390.3	424.4	674.7	243.6	<u>283.8</u>
	sst2	219.9	<u>89.6</u>	174.9	373.6	202.0	51.7
	mrpc	318.2	301.1	235.2	332.3	<u>173.1</u>	49.0

Table 1. **Active model selection main results: cumulative regret at step 100.** Best method per task in bold, second best underlined (lower is better). CODA is state-of-the-art on 18 out of 26 tasks, often significantly outperforming the next-best method, *e.g.* by $90\times$ on painting→real. Variances reported in supplemental.

instantiate the surrogate model as the ensemble of all models in H and return the model with the lowest unbiased risk estimate every time step.

ModelSelector [45] ModelSelector is currently the state-of-the-art method for active model selection. It utilizes a probability distribution similar to our P_{Best} , but computes this independently of any per-model performance metrics. Their P_{Best} is updated according to the following update rule: $P_{\text{Best},t+1}(h_k) = \frac{1-\epsilon}{\epsilon} \times P_{\text{Best},t}(h_k)$ if h_k gets the label at time t correct. ϵ is a learning rate hyperparameter that is set with a self-supervised protocol. We follow this protocol using their codebase to find the optimal ϵ for any datasets that they do not benchmark in their paper.

7. Experiments

Experimental settings All results are reported as the mean over five random seeds. We do not tune the hyperparameters of our method on each dataset; instead we select fixed values $\{T = 0.5, \alpha = 0.1, \eta = 0.01\}$ based on a limited set of initial experiments.

Active model selection results Our main results for active model selection are shown in Tab. 1. We use accuracy as our loss function and cumulative regret at step 100 as our main point of comparison for all methods, as this provides a summary of overall performance in the few-label regime where active selection is most impactful. We report additional metrics in the supplemental. For a more detailed look at performance within this time window, we visualize the re-

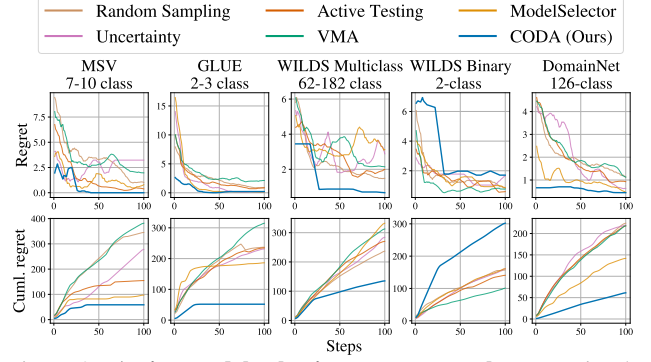


Figure 3. **Active model selection average results.** We visualize regret (top row) and cumulative regret (bottom row) from time steps 1 to 100, median value across all tasks within benchmarks. Lower is better. CODA is consistently the best performer over time for all settings except the binary classification tasks in WILDS. Full per-task results in supplemental.

gret and cumulative regret over time in more detail in Fig. 3 and for all tasks separately in the supplemental.

CODA outperforms all prior work on 18 out of the 26 datasets tested, often significantly, resulting in an over 80% reduction in regret compared to the next-best method on 5 datasets, greater than 50% reduction compared to the next-best on 11 datasets, and greater than 25% reduction from the next-best on 15 datasets. The next-best method is inconsistent across benchmarks. Of the eight datasets where ours is not the best method, uncertainty-based sampling and ModelSelector are best on three each while Active Testing and VMA are best on one each. Our method performs worst on CoLA and CivilComments, where we underperform random sampling by $6.1\times$ and $2.3\times$, respectively. We analyze these successes and failures in more detail in the remainder of this section.

Ablation studies First we ablate the priors used by our method in Fig. 5. We see that the consensus-informed priors introduced in Eq. (5) are a key component of our good performance. Removing them increases regret significantly during the early parts of the label acquisition process. We also see that the diagonal-weighted prior introduced in Eq. (7) to regularize the consensus priors performs better than uniform on some, but not all, datasets regardless of the number of classes. Note that in binary classification, the uniform and diagonal settings are equivalent.

In Fig. 6, we ablate our acquisition function (expected information gain w.r.t. P_{Best} , Eq. (17)). We compare with random sampling as well as uncertainty-based sampling [13] which greedily selects the data point with the largest Shannon entropy in the mean prediction of all models. We see that expected information gain results in the lowest cumulative regret in most cases. However we also see that uncertainty-based sampling also performs well in combination with the CODA probabilistic framework, sometimes outperforming expected information gain.

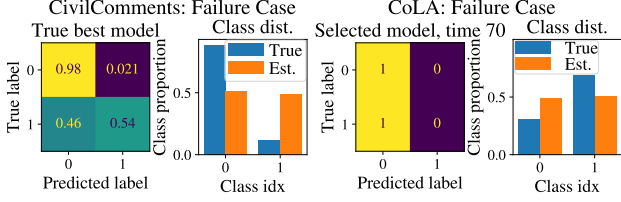


Figure 4. **Failure analysis on CivilComments and CoLA.** CODA may underestimate the performance of very biased classifiers in early steps (CivilComments, left), but overestimate them in later steps (CoLA, right) when there is also data imbalance present (blue bars). More details in Sec. 7.

Limitations and failure analysis We investigate CODA’s poor performance on CivilComments and CoLA in Fig. 4. We find that poor performance in these cases is caused by a combination of data imbalance and model bias. CODA uses the predicted class marginal in its updates (Eq. (17)), but regularizes this to counteract overconfident predictions (Eq. (7)). In CivilComments this causes us to upweight the contribution of minority class predictions early on, requiring a moderate amount of samples to identify the imbalance. In addition, the best model on CivilComments is extremely biased, with 98% accuracy on the majority class but only 54% accuracy on the minority class, exacerbating the problem when selecting based on micro-accuracy. Similarly, in CoLA, CODA selects a very biased model—one that only predicts the negative class—later in the model selection process (step 70) again because the dataset is estimated to be more balanced than it actually is. We note that these failures are exceptional cases, as CODA performs well on other datasets with significant imbalance (e.g. iWildCam). This points to interesting directions for future work to address data imbalance, model bias, and use-case specific metrics.

Additional unsupervised results Though not the main focus of this paper, our method can also be used to perform *unsupervised* model selection by using only the consensus-informed priors to compute P_{Best} . We show that this is remarkably effective, matching or outperforming the previous state-of-the-art in unsupervised model selection in 20 of 26 benchmarks, in the supplemental material.

8. Conclusion and future work

We have introduced a new method for active model selection that can identify the best model in a pool of candidates with significantly fewer human labels than prior work. The ability to do so has implications for users of pre-trained machine learning models, who can use CODA to efficiently select the best model for their dataset, as well as for researchers in fields such as domain adaptation who face model selection challenges during model development.

Our work opens up several exciting areas for future research. Within the active model selection framework there are several interesting directions: (1) How to better construct and utilize informative priors, whether from unsuper-

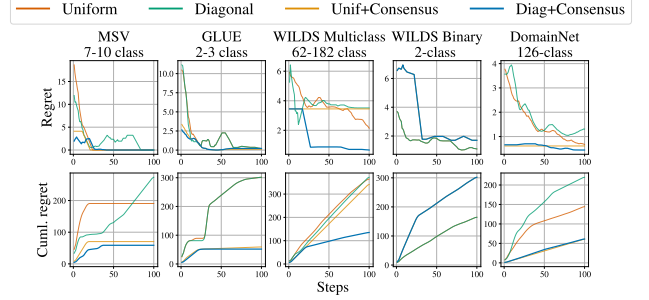


Figure 5. **Ablation of CODA prior design.** We compare a uniform prior on the confusion matrices (top row) with diagonal upweighting (“Diag.” column; Eq. (7)) and consensus prior (“Cons.” column; Eq. (5)) we introduce. For binary classification tasks, uniform and diagonal weighting are equivalent. In most cases, both the consensus prior and diagonal upweighting provide benefits, and are complementary.

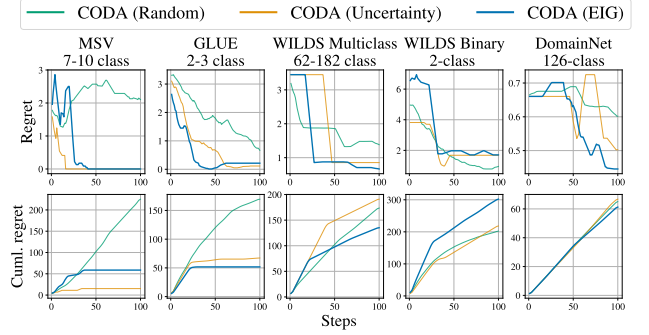


Figure 6. **Ablation of CODA acquisition function.** We use the CODA probabilistic framework and compare different data point acquisition functions: random sampling, uncertainty-based sampling, and expected information gain (EIG, the default). We see that EIG typically improves upon other sampling approaches, however uncertainty-based sampling is also a strong acquisition function in combination with the rest of the CODA framework.

vised procedures or by incorporating human-provided domain knowledge; (2) Extending the active model selection framework beyond accuracy and classification to support more tasks and target metrics; (3) Exploring more sophisticated probabilistic models that better capture predictive quantities such as model confidence.

More broadly, active model selection can be seen as one facet of a larger research goal of how to best utilize human effort in the development and deployment of machine learning systems. We show that active model selection is a particularly effective use of annotation effort, but there are many things that *could* be done with collected labels, either in serial or in parallel. For example, an interesting research direction is how to perform active learning and/or active testing concurrently with active model selection, and how to efficiently allocate effort to the different tasks. We hope our work can provide a strong starting point for investigating these questions.

Acknowledgments

We would like to thank Julia Chae, Mark Hamilton, Timm Haucke, Michael Hobley, Neha Hulkund, Rupa Kurinchi-Vendhan, Evan Shelhamer, and Edward Vendrow for feedback on early drafts, and Serge Belongie, Emma Pierson, Manish Raghavan, Shuvom Sadhuka, and Divya Shanmugam for helpful discussions. This work was supported in part by NSF awards #2313998, #2330423, and #2329927, NSERC award #585136, and MIT J-WAFS seed grant #2040131.

References

- [1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 6
- [2] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermesen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018. 6
- [3] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 2
- [4] Sara Beery, Grant Van Horn, Oisín Mac Aodha, and Pietro Perona. The iwildcam 2018 challenge dataset. *arXiv preprint arXiv:1904.05986*, 2019. 6
- [5] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019. 6
- [6] Steve Branson, Grant Van Horn, and Pietro Perona. Lean crowdsourcing: Combining humans and machines in an online system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7474–7483, 2017. 3
- [7] Alexander Braylan, Madalyn Marabella, Omar Alonso, and Matthew Lease. A general model for aggregating annotations across simple, complex, and multi-object annotation tasks. *Journal of Artificial Intelligence Research*, 78:901–973, 2023. 3
- [8] M Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, et al. Monai: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*, 2022. 1
- [9] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pages 1081–1090. PMLR, 2019. 6
- [10] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2018. 6
- [11] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3941–3950, 2020. 6, 13
- [12] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12455–12464, 2020. 6
- [13] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine learning proceedings 1995*, pages 150–157. Elsevier, 1995. 6, 7
- [14] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979. 2, 3, 4, 15
- [15] Linus Ericsson, Da Li, and Timothy Hospedales. Better practices for domain adaptation. In *International Conference on Automated Machine Learning*, pages 4–1. PMLR, 2023. 1, 2
- [16] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. On statistical bias in active learning: How and when to fix it. *arXiv preprint arXiv:2101.11665*, 2021. 6
- [17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016. 6
- [18] Andres Hernandez, Zhongqi Miao, Luisa Vargas, Rahul Dodhia, Pablo Arbelaez, and Juan M Lavista Ferres. Pytorch-wildlife: A collaborative deep learning framework for conservation. *arXiv preprint arXiv:2405.12930*, 2024. 1
- [19] Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. Learning whom to trust with mace. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, 2013. 3
- [20] Dapeng Hu, Mi Luo, Jian Liang, and Chuan-Sheng Foo. Towards reliable model selection for unsupervised domain adaptation: An empirical study and a certified baseline. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. 2, 13
- [21] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018. 6
- [22] HuggingFace. HuggingFace Models, 2025. 1
- [23] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 464–480. Springer, 2020. 6
- [24] Tarun Kalluri, Sreyas Ravichandran, and Manmohan Chandraker. Uda-bench: Revisiting common assumptions in un-

- supervised domain adaptation using a standardized framework. In *European Conference on Computer Vision*, pages 199–220. Springer, 2024. 1, 2
- [25] 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*, pages 307–315. PMLR, 2021. 2, 3
- [26] Parnian Kassraie, Nicolas Emmenegger, Andreas Krause, and Aldo Pacchiano. Anytime model selection in linear bandits, 2023. 3
- [27] Justin Kay, Peter Kulits, Suzanne Stathatos, Siqi Deng, Erik Young, Sara Beery, Grant Van Horn, and Pietro Perona. The caltech fish counting dataset: A benchmark for multiple-object tracking and counting. In *European Conference on Computer Vision*, pages 290–311. Springer, 2022. 2
- [28] Justin Kay, Suzanne Stathatos, Siqi Deng, Erik Young, Pietro Perona, Sara Beery, and Grant Van Horn. Unsupervised domain adaptation in the real world: A case study in sonar video. In *NeurIPS 2023 Computational Sustainability: Promises and Pitfalls from Theory to Deployment*, 2023. 1
- [29] Justin Kay, Timm Haucke, Suzanne Stathatos, Siqi Deng, Erik Young, Pietro Perona, Sara Beery, and Grant Van Horn. Align and distill: Unifying and improving domain adaptive object detection. *arXiv preprint arXiv:2403.12029*, 2024. 1
- [30] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021. 2, 6, 16
- [31] Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, pages 5753–5763. PMLR, 2021. 2, 3, 6
- [32] Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Thomas Rainforth. Active surrogate estimators: An active learning approach to label-efficient model evaluation. *Advances in Neural Information Processing Systems*, 35:24557–24570, 2022. 2, 3
- [33] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6
- [34] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 6
- [35] Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. A survey on deep active learning: Recent advances and new frontiers. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1, 2
- [36] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16632–16642, 2021. 6
- [37] 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, 2022. 3
- [38] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 6
- [39] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 6
- [40] 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*. 2, 3, 6
- [41] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International conference on machine learning*, pages 7721–7735. PMLR, 2021. 2
- [42] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Unsupervised domain adaptation: A reality check. *arXiv preprint arXiv:2111.15672*, 2021. 1, 2, 6
- [43] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Three new validators and a large-scale benchmark ranking for unsupervised domain adaptation. *arXiv preprint arXiv:2208.07360*, 2022. 1, 2, 6, 13, 15
- [44] Phuc Nguyen, Deva Ramanan, and Charless Fowlkes. Active testing: An efficient and robust framework for estimating accuracy. In *International Conference on Machine Learning*, pages 3759–3768. PMLR, 2018. 2
- [45] Patrik Okanovic, Andreas Kirsch, Jannes Kasper, Torsten Hoefer, Andreas Krause, and Nezihe Merve Gürel. All models are wrong, some are useful: Model selection with limited labels. *arXiv preprint arXiv:2410.13609*, 2024. 2, 3, 6, 7, 13, 16
- [46] Wei Ouyang, Fynn Beuttenmueller, Estibaliz Gómez-de Mariscal, Constantin Pape, Tom Burke, Carlos Garcia-López-de Haro, Craig Russell, Lucía Moya-Sans, Cristina De-La-Torre-Gutiérrez, Deborah Schmidt, et al. Bioimage model zoo: a community-driven resource for accessible deep learning in bioimage analysis. *BioRxiv*, pages 2022–06, 2022. 1
- [47] Rebecca J Passonneau and Bob Carpenter. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326, 2014. 2, 3, 4
- [48] Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. Comparing bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585, 2018. 3
- [49] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 6

- [50] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, page 269, 2017. 3
- [51] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018. 2
- [52] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 2
- [53] Luca Robbiano, Muhammad Rameez Ur Rahman, Fabio Galasso, Barbara Caputo, and Fabio Maria Carlucci. Adversarial branch architecture search for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2918–2928, 2022. 2
- [54] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 6
- [55] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058, 2019. 6
- [56] Kuniaki Saito, Donghyun Kim, Piotr Teterwak, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9184–9193, 2021. 1, 2
- [57] Amartya Sanyal, Yaxi Hu, Yaodong Yu, Yian Ma, Yixin Wang, and Bernhard Schölkopf. Accuracy on the wrong line: On the pitfalls of noisy data for out-of-distribution generalisation. *arXiv preprint arXiv:2406.19049*, 2024. 2
- [58] Christoph Sawade, Niels Landwehr, and Tobias Scheffer. Active comparison of prediction models. *Advances in neural information processing systems*, 25, 2012. 2, 3
- [59] Burr Settles. Active learning literature survey. 2009. 1, 2
- [60] Divya Shanmugam, Shuvom Sadhuka, Manish Raghavan, John Gutttag, Bonnie Berger, and Emma Pierson. Evaluating multiple models using labeled and unlabeled data. *arXiv preprint arXiv:2501.11866*, 2025. 2
- [61] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. *arXiv preprint arXiv:1206.6438*, 2012. 6
- [62] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer vision—ECCV 2016 workshops: Amsterdam, the Netherlands, October 8–10 and 15–16, 2016, proceedings, part III 14*, pages 443–450. Springer, 2016. 6
- [63] Takumi Tamura, Hiroyoshi Ito, Satoshi Oyama, and Atsuyuki Morishima. Influence of ai’s uncertainty in the dawid-skene aggregation for human-ai crowdsourcing. In *International Conference on Information*, pages 232–247. Springer, 2024. 3
- [64] Damien Teney, Yong Lin, Seong Joon Oh, and Ehsan Abbasnejad. Id and ood performance are sometimes inversely correlated on real-world datasets. *Advances in Neural Information Processing Systems*, 36:71703–71722, 2023. 2
- [65] Grant Van Horn, Steve Branson, Scott Loarie, Serge Belongie, and Pietro Perona. Lean multiclass crowdsourcing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2723, 2018. 3
- [66] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999. 6
- [67] 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. *arXiv preprint arXiv:1804.07461*, 2018. 6
- [68] Peter Welinder, Steve Branson, Pietro Perona, and Serge Belongie. The multidimensional wisdom of crowds. *Advances in neural information processing systems*, 23, 2010. 3
- [69] Jianfei Yang, Hanjie Qian, Yuecong Xu, Kai Wang, and Lihua Xie. Can we evaluate domain adaptation models without target-domain labels? *arXiv preprint arXiv:2305.18712*, 2023. 2
- [70] Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 7124–7133. PMLR, 2019. 1, 2, 13

Supplemental Material

9. Additional results

9.1. Alternate metrics

Here we report tabular results for several alternate metrics, providing additional points of comparison to supplement our main results of cumulative regret at step 100 (Tab. 1).

9.1.1. Variance between seeds

Task	Random Sampling	Uncertainty	Active Testing	VMA	Model Selector	CODA (Ours)
DomainNet126						
real→sketch	147.1±38.5	197.7±13.0	269.8±119.8	119.2±51.9	88.8±9.8	101.2±0.0
real→painting	143.6±37.1	167.9±6.8	118.9±68.8	139.9±53.3	92.1±1.6	87.2±2.8
real→clipart	237.7±88.7	217.6±23.7	152.1±40.0	206.9±100.7	153.2±9.5	231.7±0.0
sketch→real	280.4±164.4	252.7±11.8	236.4±160.5	273.4±132.7	268.4±7.5	11.9±0.0
sketch→painting	156.6±72.6	181.9±7.3	237.7±151.7	157.1±60.9	173.9±7.1	13.0±0.0
sketch→clipart	228.2±74.5	224.6±43.2	162.0±83.6	227.9±85.6	38.1±9.2	432.4±1.9
painting→real	364.8±164.4	224.0±7.2	215.6±94.4	358.9±193.6	293.4±20.3	2.4±0.0
painting→sketch	179.3±53.5	440.7±32.9	202.5±79.7	211.5±73.3	209.8±5.2	72.3±0.2
painting→clipart	222.7±144.5	296.6±6.1	251.4±111.8	271.8±116.9	73.2±3.2	43.1±0.1
clipart→real	322.1±127.8	177.1±34.6	159.1±55.6	306.4±181.7	72.7±15.3	25.3±0.0
clipart→sketch	247.2±143.0	924.8±22.7	282.6±186.5	291.3±163.9	532.7±76.1	51.3±0.0
clipart→painting	147.0±46.2	162.9±13.3	222.6±100.5	167.9±100.2	131.7±2.5	122.2±0.1
WILDS						
iwildcam	287.0±65.1	380.4±7.1	392.1±194.1	440.6±103.6	459.0±56.0	201.7±0.0
camelyon	175.2±81.7	311.6±11.9	206.1±121.1	160.1±76.8	198.3±90.6	288.7±0.0
fmow	189.7±66.8	191.9±8.4	153.0±38.7	189.2±62.6	211.7±28.1	70.0±0.2
civilcomments	140.9±75.5	13.3±2.6	76.7±119.0	50.1±29.3	125.3±59.6	318.6±0.0
MSV						
cifar10-low	410.6±231.1	629.7±5.9	399.9±207.5	476.5±114.4	567.2±23.9	58.7±0.0
cifar10-high	346.2±147.4	281.7±21.3	154.9±68.9	383.7±141.9	90.9±12.3	74.1±0.0
pacs	216.6±86.8	6.9±5.4	101.8±55.9	116.5±57.5	97.9±6.0	57.9±0.0
GLUE						
cola	368.1±80.6	169.5±27.1	239.8±162.4	317.8±191.8	207.8±61.1	2226.7±0.0
mnli	237.4±132.2	80.8±27.9	234.2±105.2	312.8±111.5	148.5±114.4	23.5±0.0
qnli	222.7±68.5	231.6±76.2	246.6±112.4	283.0±78.8	185.7±89.6	120.4±0.0
qqp	136.7±20.9	388.9±492.8	127.8±16.5	169.2±114.1	186.8±137.5	4.8±0.0
rte	375.7±184.6	390.3±68.4	424.4±244.4	674.7±186.7	243.6±73.2	283.8±0.0
sst2	219.9±108.1	89.6±39.6	174.9±54.3	373.6±128.6	202.0±131.4	51.7±0.0
mrpc	318.2±100.3	301.1±64.4	235.2±36.0	332.3±156.0	173.1±47.7	49.0±0.0

Table 2. Main results with variance between seeds: Cumulative regret at step 100. Same as Tab. 1 but \pm one standard deviation between runs with different random seeds (5 random seeds used). Standard deviation of 0.0 (as in many CODA runs) indicates method is not stochastic for that task, *e.g.* because of asymmetric priors resulting in deterministic selection.

9.1.2. Instantaneous regret

We report the average instantaneous regret of each method at steps 50 and 100 in Tab. 3 and Tab. 4, respectively. These results are more influenced by stochasticity than cumulative regret, *i.e.* instantaneous regret may fluctuate widely between steps (see Fig. 8).

9.1.3. Success rate

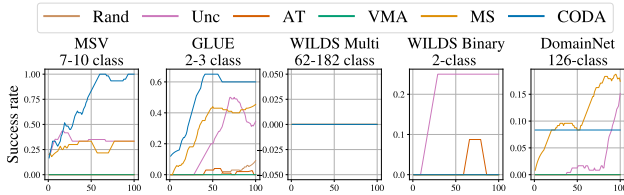


Figure 7. “Success rate” of each model in selecting the *absolute best model at each time step*. Mean over 5 random seeds. In all datasets, several methods have not yet selected the absolute best model by time step 100.

Task	Random Sampling	Uncertainty	Active Testing	VMA	Model Selector	CODA (Ours)
DomainNet126						
real→sketch	1.3±0.6	2.0±0.2	0.8±0.6	1.3±1.2	0.6±0.0	1.0±0.0
real→painting	1.2±0.4	1.2±0.5	1.4±1.4	1.5±1.0	0.9±0.3	1.2±0.0
real→clipart	1.6±1.8	0.5±0.6	1.0±0.9	0.8±0.8	0.2±0.1	2.3±0.0
sketch→real	2.0±3.0	0.4±0.2	3.0±3.0	2.6±3.1	5.4±0.0	0.1±0.0
sketch→painting	1.1±1.3	0.9±1.1	2.2±2.3	0.9±0.6	5.5±0.0	0.1±0.0
sketch→clipart	1.8±1.4	0.2±0.2	1.6±1.6	2.1±0.8	0.0±0.0	4.8±0.0
painting→real	4.1±2.9	2.1±0.7	1.3±0.9	2.3±1.1	4.6±4.6	0.0±0.0
painting→sketch	1.5±1.8	4.3±0.0	2.3±1.5	1.9±1.5	1.8±0.7	0.9±0.0
painting→clipart	1.9±2.3	2.8±0.0	3.2±1.7	2.8±1.5	0.7±0.0	0.4±0.0
clipart→real	2.6±1.7	0.4±0.0	0.2±0.4	2.8±1.8	0.1±0.0	0.3±0.0
clipart→sketch	3.0±2.3	8.7±2.9	0.8±0.7	2.6±2.0	5.4±0.0	0.4±0.0
clipart→painting	1.6±0.9	1.1±0.0	2.6±1.9	1.3±1.5	1.0±0.1	1.2±0.0
WILDS						
iwildcam	2.1±0.9	3.0±0.0	4.5±2.3	6.1±1.9	4.6±1.3	0.9±0.0
camelyon	0.5±0.2	3.5±0.0	2.1±1.7	0.9±1.0	1.5±0.8	0.6±0.0
fmow	1.4±1.2	0.6±0.0	1.2±0.7	1.6±1.0	1.0±0.0	0.8±0.0
civilcomments	0.9±1.1	0.0±0.0	0.8±1.5	0.3±0.3	1.6±2.7	3.3±0.0
MSV						
cifar10-low	5.5±4.1	7.7±0.0	5.3±3.4	4.7±2.6	7.5±0.4	0.0±0.0
cifar10-high	3.1±2.3	3.2±0.0	0.2±0.2	2.6±1.7	0.0±0.0	0.0±0.0
pacs	2.2±1.9	0.0±0.0	0.6±0.8	0.8±0.7	1.4±0.0	0.0±0.0
GLUE						
cola	3.6±1.3	2.0±0.9	2.3±1.9	2.5±1.9	1.4±1.0	0.4±0.0
mnli	2.6±3.0	0.1±0.0	1.0±1.7	2.6±1.6	0.1±0.1	0.1±0.0
qnli	1.3±1.3	0.9±1.2	1.0±1.4	1.7±1.5	0.1±0.2	0.0±0.0
qqp	1.4±0.2	1.3±2.8	1.1±0.1	1.0±0.6	0.3±0.4	0.0±0.0
rte	3.2±4.4	0.0±0.0	3.7±3.6	5.9±3.9	0.0±0.0	0.0±0.0
sst2	1.9±2.5	0.2±0.0	0.1±0.1	3.1±3.5	0.0±0.0	0.0±0.0
mrpc	2.0±2.4	1.6±1.9	1.6±1.2	2.8±2.4	0.8±0.5	0.5±0.0

Table 3. Average instantaneous regret at step 50. Mean and standard deviation reported over 5 random seeds.

Task	Random Sampling	Uncertainty	Active Testing	VMA	Model Selector	CODA (Ours)
DomainNet126						
real→sketch	1.2±1.1	0.0±0.0	2.2±2.2	1.0±0.6	0.8±0.5	1.0±0.0
real→painting	1.1±0.6	0.6±0.0	0.8±0.2	0.9±0.6	1.3±0.3	0.2±0.0
real→clipart	0.6±0.4	1.0±0.0	0.6±0.3	1.5±2.0	0.2±0.0	2.3±0.0
sketch→real	1.7±2.8	0.5±0.0	0.7±1.1	0.2±0.1	0.1±0.0	0.1±0.0
sketch→painting	0.7±0.7	0.0±0.0	0.8±0.8	0.8±0.9	0.1±0.1	0.1±0.0
sketch→clipart	0.9±0.8	0.2±0.3	0.9±0.5	2.1±2.0	0.0±0.0	0.5±0.0
painting→real	3.3±2.4	1.2±0.0	1.1±0.9	1.4±0.8	1.2±0.0	0.0±0.0
painting→sketch	0.6±0.5	0.7±0.0	0.7±0.7	1.3±1.9	1.7±0.0	0.5±0.0
painting→clipart	2.0±1.7	0.5±0.2	2.3±1.7	1.7±1.6	0.3±0.0	0.4±0.0
clipart→real	1.4±1.6	0.6±0.0	0.7±0.8	0.7±0.6	0.1±0.0	0.3±0.0
clipart→sketch	1.6±2.0	5.3±2.1	1.0±0.7	0.6±0.4	0.6±0.0	0.6±0.0
clipart→painting	0.6±0.5	1.1±0.0	1.4±0.7	1.1±0.3	1.0±0.1	1.1±0.0
WILDS						
iwildcam	1.4±1.7	3.0±0.0	3.2±3.0	2.8±2.5	4.1±1.6	0.9±0.0
camelyon	0.5±0.9	3.3±0.7	0.9±1.5	1.4±1.5	1.2±0.9	0.6±0.0
fmow	1.4±0.9	1.0±0.8	0.7±0.3	1.2±0.5	1.5±1.4	0.4±0.0
civilcomments	0.7±1.2	0.0±0.0	0.3±0.4	0.2±0.4	0.3±0.2	2.8±0.0
MSV						
cifar10-low	1.2±1.3	5.7±0.0	2.8±2.9	3.2±1.2	6.3±1.9	0.0±0.0
cifar10-high	1.0±1.4	3.2±0.0	0.3±0.2	1.9±1.6	0.0±0.0	0.0±0.0
pacs	1.1±0.7	0.0±0.0	0.4±0.6	0.6±0.5	0.8±0.5	0.0±0.0
GLUE						
cola	2.9±1.8	0.2±0.5	1.4±1.2	2.8±2.7	0.4±0.0	56.0±0.0
mnli	0.3±0.3	0.1±0.0	0.4±0.2	0.7±0.8	0.2±0.1	0.2±0.0
qnli	0.7±1.1	0.0±0.0	1.6±2.1	1.5±1.9	0.1±0.2	0.4±0.0
qqp	1.0±0.6	0.2±0.3	1.0±1.1	0.9±0.7	0.7±0.9	0.0±0.0
rte	1.2±2.6	0.0±0.0	0.8±1.8	4.3±4.2	0.0±0.0	0.0±0.0
sst2	0.8±0.7	0.1±0.0	1.0±0.9	2.0±2.9	0.0±0.0	0.0±0.0
mrpc	0.2±0.2	0.2±0.1	0.4±0.5	2.6±2.9	0.0±0.0	0.5±0.0

Table 4. Average instantaneous regret at step 100. Mean and standard deviation over 5 random seeds.

9.2. Unsupervised model selection results

Our method defines a prior over model performance that creates a strong starting point for active model selection. This can be used in isolation, without active label collection, to perform unsupervised model selection. We compare against five existing methods for model selection in unsupervised domain adaptation:

Source validation Model selection is performed using validation accuracy on “source” data, *i.e.* using a validation set

	Task	Source val	DEV	Entropy	BNM	EnsV	CODA
DomainNet126	real → sketch	0.5	4.6	3.1	3.1	0.4	1.0
	real → clip	4.5	49.3	0.3	6.5	2.8	2.3
	real → paint	1.3	34.7	2.3	2.0	0.2	1.2
	sketch → real	4.7	6.3	9.1	8.5	4.5	0.1
	sketch → clip	2.4	13.4	5.9	6.4	3.0	4.8
	sketch → paint	3.8	3.5	3.2	3.2	1.7	0.1
	clip → real	1.5	6.3	5.6	5.6	0.5	0.3
	clip → sketch	2.8	4.9	5.1	4.9	2.2	0.4
	clip → paint	3.1	7.8	4.2	4.2	4.2	1.2
	paint → real	0.0	36.6	3.3	3.3	0.1	0.1
	paint → sketch	0.7	26.1	4.3	4.3	3.3	0.9
	paint → clip	2.9	16.2	6.3	6.3	1.1	0.4
WILDS	iwildcam	9.8	-	-	-	0.9	6.1
	camelyon	4.3	-	-	-	13.1	11.3
	fmow	0.8	-	-	-	0.9	0.8
	civilcomments	-	-	-	-	3.8	4.7
MSV	cifar10-low	-	-	-	-	2.3	2.3
	cifar10-high	-	-	-	-	4.9	4.9
	pacs	-	-	-	-	0.4	0.4
GLUE	cola	-	-	-	-	5.3	5.0
	mnli	-	-	-	-	3.0	3.0
	qnli	-	-	-	-	3.3	3.3
	qqp	-	-	-	-	1.1	1.1
	rte	-	-	-	-	14.8	14.8
	sst2	-	-	-	-	3.6	3.6
	mrpc	-	-	-	-	1.0	1.0

Table 5. **Unsupervised model selection results.** We report regret at step 0 (lower is better) for all methods on all tasks. Best method for each task is in bold. CODA matches or exceeds state-of-the-art performance on 20 out of 26 tasks. Note that because models/predictions for MSV and GLUE are black-box/one-hot (as in ModelSelector [45]), the only comparison we are able to make is to EnsV.

from the same distribution as the training set.

Target entropy [43] Shannon entropy is computed on prediction scores on the test set. Model selection is performed by selecting the model with the lowest entropy, *i.e.* the model that is most confident in its test predictions.

Deep embedded validation [70] Computes a classification loss for each source validation sample, and weights each loss based on a computed probability that the sample “belongs to” the test domain. This probability comes from a separate domain classifier trained on source and test data.

Batch nuclear norm [11] Performs singular value decomposition on the prediction matrix. Model selection is performed by selecting the model with the minimum nuclear norm (*i.e.* minimum sum of singular values).

EnsV [20] All models in the hypothesis set are ensembled. To perform model selection, accuracy is estimated for each model with respect to the ensemble’s predictions.

9.3. Unaggregated results on all tasks

In Fig. 8 we visualize regret and cumulative regret at every step for all baselines on all tasks.

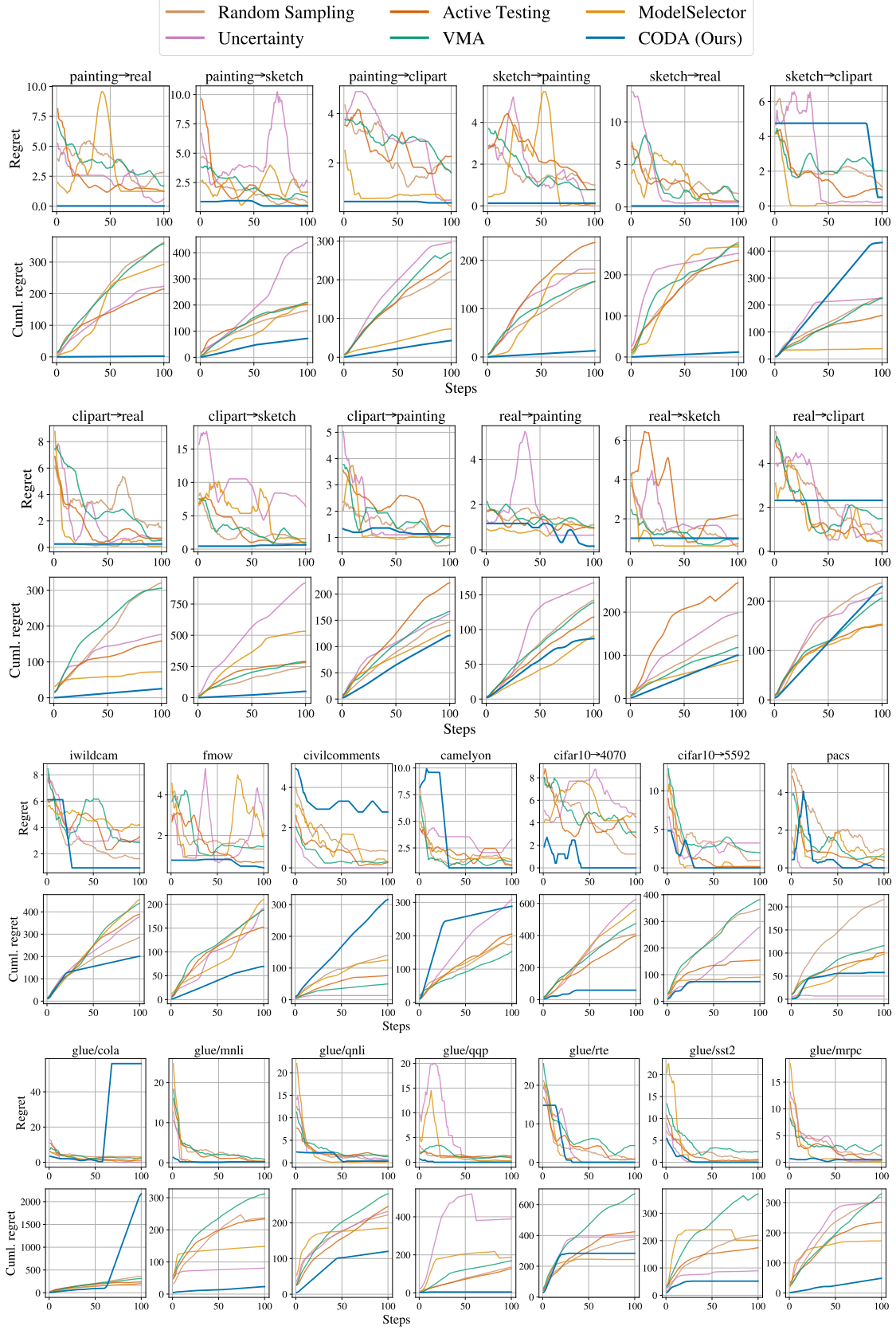


Figure 8. Results on all benchmarks.

10. Implementation details

10.1. Dawid-Skene data generating process

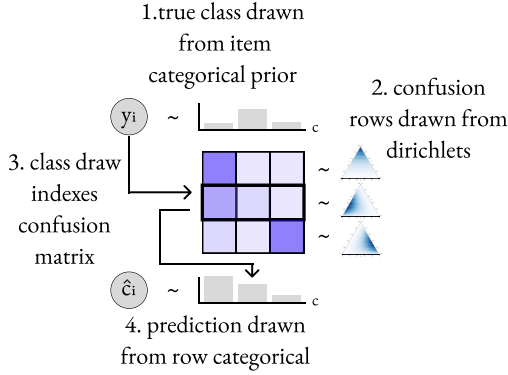


Figure 9. A visual depiction of the Dawid-Skene [14] data generating process that we adapt to active model selection. See Sec. 4.1 of the main paper for more details.

We visualize the data generating process of the Bayesian implementation of the Dawid-Skene model in Fig. 9, as described in Sec. 4.1. We repeat the text here for easy reference.

The data generating process proceeds as follows:

1. Each data point’s true class label y_i is drawn randomly from per-data-point prior distributions over which class that data point could be, $y_i \sim \text{Cat}(\pi(x_i))$.
2. Each row of the classifier’s confusion matrix is drawn randomly from per-row distributions, $M_{k,c,\cdot} \sim \theta_{k,c}$, where $\theta_{k,c}$ is the prior distribution over what the row of the confusion matrix could be. To accommodate Bayesian updates, we initialize each $\theta_{k,c}$ to be a Dirichlet prior.
3. The sampled true class indexes into the corresponding row of the classifier’s confusion matrix, M_{k,y_i} .
4. The classifier’s prediction for that data point is sampled from the distribution over that row’s cells, $\hat{c}_{k,i} \sim \text{Cat}(M_{k,y_i})$.

10.2. Computing P_{Best}

We illustrate visually the computation of P_{Best} from Sec. 4.3 in Fig. 10 in the simplified case of two models. To compute the probability that Model 1 is best, we integrate over all possible accuracy values that Model 1 *could* have. For every possible accuracy, we compute the probability that Model 1 has that accuracy (defined by its PDF f), multiplied that the probability Model 2 has accuracy *less* than that value (defined by its CDF F).

11. Data and model details

We provide more details about the datasets and models in our benchmarking suite in Tab. 6, Tab. 7, and Tab. 8.

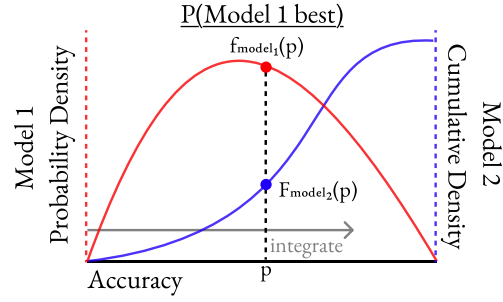


Figure 10. A visual depiction of the integration technique used to construct our distribution over which model is best, P_{Best} . See Sec. 10.2 of the supplemental and Sec. 4.3 of the main paper for more details.

DomainNet126			
Task	Num. Classes	Num. Checkpoints	Test set size
sketch_painting	126	10 algs 20 epochs = 200	3086
sketch_real	126	10 algs 20 epochs = 200	20939
sketch_clipart	126	10 algs 20 epochs = 200	5611
clipart_painting	126	10 algs 20 epochs = 200	3086
clipart_sketch	126	10 algs 20 epochs = 200	7313
clipart_real	126	10 algs 20 epochs = 200	20939
painting_sketch	126	10 algs 20 epochs = 200	7313
painting_real	126	10 algs 20 epochs = 200	20939
painting_clipart	126	10 algs 20 epochs = 200	5611
real_painting	126	10 algs 20 epochs = 200	3086
real_sketch	126	10 algs 20 epochs = 200	7313
real_clipart	126	10 algs 20 epochs = 200	5611

Table 6. **Dataset details: DomainNet126.** For each transfer task, we first train a “source-only” model on the source domain. We then train 10 UDA models for each transfer task (source domain \rightarrow target domain) using the Powerful Benchmark codebase [43].

WILDS				
Task	Num. Classes	Num. Checkpoints	Test set size	Regret w/ val.
RxRx1	1139	4 algs 18 epochs = 72	34,432	0.1
Amazon	5	4 algs 3 epochs = 12	100,050	0.1
CivilComments	2	4 algs 5 epochs = 20	133,782	N/A
fMoW	62	4 algs 60 epochs = 240	22,108	0.8
iWildCam	182	4 algs 12 epochs = 48	42,791	9.8
Camelyon17	2	4 algs 10 epochs = 40	85,054	4.3

Table 7. **Dataset details: WILDS.** We show metrics for all classification datasets in WILDS where we could perform model selection. In our main experiments, we only use the benchmarks where near-perfect model selection can be performed trivially using the default in-distribution validation set. We train all models ourselves using the public code from Koh et al. [30].

MSV			
Dataset	Num. Classes	Num. Checkpoints	Test set size
CIFAR10-High	10	80	10000
CIFAR10-Low	10	80	10000
PACS	7	30	9991

GLUE			
Dataset	Num. Classes	Num. Checkpoints	Test set size
CoLA	2	109	1043
MNLI	3	82	9815
QNLI	2	90	5463
QQP	2	101	40430
RTE	2	87	277
SST2	2	97	872
MRPC	2	95	408

Table 8. **Dataset details: MSV and GLUE.** All model checkpoints sourced directly from Okanovic et al. [45].