Bayesian Online Learning for Consensus Prediction

Sam Showalter[*,1] **Alex Boyd**[*,2] **Padhraic Smyth**[1,2] **Mark Steyvers**[1,3]
[1]Department of Computer Science [2]Department of Statistics [3]Department of Cognitive Science
University of California, Irvine

Abstract

Given a pre-trained classifier and multiple human experts, we investigate the task of online classification where model predictions are provided for free but querying humans incurs a cost. In this practical but under-explored setting, oracle ground truth is not available. Instead, the prediction target is defined as the consensus vote of all experts. Given that querying full consensus can be costly, we propose a general framework for online Bayesian consensus estimation, leveraging properties of the multivariate hypergeometric distribution. Based on this framework, we propose a family of methods that dynamically estimate expert consensus from partial feedback by producing a posterior over expert and model beliefs. Analyzing this posterior induces an interpretable trade-off between querying cost and classification performance. We demonstrate the efficacy of our framework against a variety of baselines on CIFAR-10H and ImageNet-16H, two large-scale crowdsourced datasets.

1 INTRODUCTION

As machine learning classifiers have increasingly become part of society over the past decade, there is a growing interest in integrating the decisions of humans and machines, particularly for high-stakes applications such as medicine and autonomous vehicles. For example, instead of using a fully automated (or manual) system to classify medical images, a preferred approach in some contexts may be to leverage the strengths of both models and human experts, rather than solely relying on one or the other. A number of different problems have been investigated in this context, including frameworks for learning when to defer to a model (Madras et al., 2018; Mozannar and Sontag, 2020), learning when to switch between human and AI agents or to delegate tasks (Lubars and Tan, 2019; Meresht et al., 2022; Pinski et al., 2023), and learning how to combine predictions from both humans and models (Steyvers et al., 2022).

Distinct from prior work, a key aspect of our scenario of interest is that the target variable y is defined as the consensus (plurality) vote of N human experts, rather than assuming that y is available via an oracle (e.g. from a single, infallible expert or via a direct measurement of y). We use pre-trained model predictions, along with a subset of expert votes, to predict expert consensus as if all experts had been queried. Like many active and online learning settings, our goal is to maximize our classification accuracy (of consensus) given an annotation budget.

In particular, assume we are required to make online class label predictions for examples x_t (e.g., medical or astronomical images) over time $t = 1, 2, \ldots, T$, where the examples x_t and labels $y_t \in \{1, \ldots, K\}$ are being drawn IID from some unknown distribution $\mathbb{P}(x, y)$. To make a prediction for each x_t, we have access to K class probabilities produced by a pre-trained model f at no cost. In addition, we have access to label prediction "votes" from N human experts at some cost per expert; we can query no experts, one expert, two experts, and etc., up to N total experts. We assume that the accuracy of the pre-trained model and the identity of individual experts are unknown. Instead, we utilize model predictions and partially observed collections of expert votes. The key problem of interest is how to optimize predictive performance while trading-off the model's predictions (at zero cost) with the human expert votes (at non-zero cost) on a per-sample basis. For example, if N is odd we can identify consensus by querying until one class possesses $(N + 1)/2$ expert votes. However, in practice we may be able to query

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

*Authors contributed equally; correspondence to showalte@uci.edu

fewer experts and still accurately estimate consensus, depending on the quality of the model's predictions.

This type of problem is relatively common in practice, where the best we can do to determine ground truth is rely on the consensus of human experts, e.g., in citizen science (Wright et al., 2017; Beck et al., 2018) or in medicine (Bien et al., 2018; Rajpurkar et al., 2020; Stutz et al., 2023). For example, a hospital may have access to a pre-trained model f as well as its own set of expert radiologists. The hospital views the consensus of the expert radiologists as the ideal predictor, but would like to have an algorithmic workflow where the expert radiologists are consulted only to the extent that they are needed, depending on model predictions.

We adopt a Bayesian approach to this problem, using a generative model for model predictions and expert votes. Our Bayesian framework allows us to update our beliefs sequentially as examples x_t, model predictions f_t, and expert votes are observed. We show how the model's predictions and observed expert votes can be used as a prior to drive beliefs about unobserved expert votes, as well as how we can learn the parameters of the prior in an online manner. This framework allows us to account for uncertainty in (i) deciding whether or not to query experts given an example x_t, as well as in (ii) making final classification predictions for y_t.

In summary, our primary contributions are as follows:

- We develop a general Bayesian framework for online learning of human expert consensus, based on a multivariate hypergeometric likelihood model.
- We identify a computationally simple limiting case of the multivariate hypergeometric approach for the situation when the number of experts becomes infinitely large.
- We propose a well-justified heuristic tied to the model's predictive beliefs on expert consensus to decide when to query and when to predict.
- We systematically evaluate our methods on two large, human-annotated datasets and demonstrate that our proposed approach is significantly more efficient than competing baselines for this problem.

Code implementations of methods and baselines, and experiment scripts are available at https://github.com/SamShowalter/bocp.

2 RELATED WORK

Our work is broadly related to concepts in crowdsourcing, human-AI collaboration, and active online learning and model selection. As we discuss below, what distinguishes our approach from prior work is our focus on (i) predicting the consensus of a set of experts (rather than an oracle ground truth), and (ii) the performance-cost trade-offs that result from being able to query a variable number of experts per example conditioned on model predictions.

The general idea of using multiple human annotators to assign class labels, rather than a single annotator or oracle, has a long history in machine learning in the context of crowd-sourcing and citizen science (Beck et al., 2018; Sheng and Zhang, 2019; Uma et al., 2021; Sayin et al., 2021). Work on analyzing human consensus goes back at least to Cohen's proposal of the Kappa agreement statistic (Cohen, 1960), evolving in modern times to analysis of human labeling in applications such as natural language processing (Plank, 2022), and medical diagnosis (Stutz et al., 2023). Techniques such as active online learning have also been explored in this context, where the focus is on learning a model f with minimal human supervision. In this general setting, the most relevant body of prior work involves the use of model predictions to assist the human labeling process (Wright et al., 2019). A commonly used technique in this context is the use of confusion matrices to infer patterns of labeling errors for individual humans and machines (Branson et al., 2017; Van Horn et al., 2018; Tanno et al., 2019; Kerrigan et al., 2021). However, inherent in this work is the assumption that annotators are noisy estimators of a stand-alone oracle ground truth y. By contrast, we specifically assume the consensus of human experts solely defines the label y that we wish to predict.

Human-AI Collaboration A closely-related and recent body of work in machine learning develops algorithmic approaches for various workflows in human-AI collaboration (e.g., see Jarrett et al. (2022)), such as algorithms that allow models to "learn to defer" to human experts (Madras et al., 2018; Mozannar and Sontag, 2020; Verma and Nalisnick, 2022). Of particular relevance is work that focuses on optimally combining model and expert predictions, in the situation where both are assumed to be available (e.g., Steyvers et al. (2022); Choudhary et al. (2023)). This differs from our work in that prior literature assumes oracle ground truth exists (separate from the human experts) and also assumes that the cost of querying experts is negligible.

Online Active Model Selection Another topic of relevance is online active model selection (OAMS), where pre-trained model predictions from several classifiers are provided at no cost and an algorithm must learn to make sequential predictions for a stream of data (Karimi et al., 2021). Given a fixed budget, these methods can query ground truth at a cost, with the ultimate goal of identifying an optimal policy for routing samples among the provided models. Our work is

distinct from OAMS in that we only get predictions from a single classifier and re-define ground truth in terms of human expert consensus instead of an oracle. In our experimental results we adapt the approach of Karimi et al. (2021) to create a baseline for comparison with our proposed methods.

3 METHODOLOGY

Problem Setting Consider a stream of inputs $x_t \in \mathcal{X} \subset \mathbb{R}^d$ with dimension d for time $t = 1, \ldots, T$. We are interested in predicting an associated class $y_t \in \mathcal{Y} := \{1, \ldots, K\}$ at each time t. Furthermore, we would like to do so in an online fashion by predicting $y_t \,|\, x_t$ using the information seen in the previous $t-1$ timesteps. We assume unlimited access to a fixed, pre-trained classifier $f(x)$ that produces a conditional probability vector over classes; however, the expected performance and calibration of f is not known beforehand. For brevity, we will refer to specific predictions of $f(x_t)$ as f_t.

A key distinction from other settings is how the ground truth class y_t is determined. We consider the scenario where, given a fixed pool of N human experts, each member can vote on the corresponding class of each sample x_t. These votes are denoted as $h_t^i \in \{1, \ldots, K\}$ for $t = 1, \ldots, T$ and $i = 1, \ldots, N$. For convenience, we will denote $H_t^i := \sum_{j=1}^{i} \texttt{one-hot}(h_t^j) \subset \{0, 1, \ldots, N\}^K$ as the histogram of the first i votes queried at time t. For each timestep, the ground truth is determined as the *consensus* or plurality of human votes, $y_t := \arg\max_k (H_t)_k$ where $H_t := H_t^N$ and $(z)_k$ is the k^{th} element of the vector z. In the case of ties, consensus is determined randomly.

Throughout we assume that the stream of examples (x_t, y_t), $t = 1, \ldots$, are drawn IID from some unknown distribution $\mathbb{P}(x, y)$. The pretrained model f need not have been trained on $\mathbb{P}(x, y)$, e.g., might have been trained on some other distribution $\mathbb{P}'(x, y)$ subject (potentially) to covariate shift, label shift, etc, relative to $\mathbb{P}(x, y)$.

Throughout the prediction process, we may either make an immediate prediction based on current beliefs or sample one or more remaining human votes for that timestep. Sampling is conducted one-at-a-time to allow current feedback to further inform the decision making process. We will denote the number of votes drawn so far at time t for input x_t as $N_t \in \{0, 1, \ldots, N\}$ and the summary (histogram) of all votes seen as $H_t^{N_t}$. The total running cost is $C_t := \sum_{t'=1}^{t} N_{t'}$. An ideal method will produce the lowest achievable error rate when predicting $y_1, \ldots, y_T$ for any fixed average total cost C_T.

Probabilistic Model We take a probabilistic approach to modeling the generative process of the data to account for uncertainty while querying experts and making predictions. This also confers a convenient means of integrating existing predictions via f.

For a given timestep $t \in \{1, \ldots, T\}$, we assume there exists (conceptually) a population (effectively infinite) of human opinions concerning the class of x_t. By assuming individual votes to be non-fractional and conditionally independent, we have the following generative distributions:

$$\pi_t \sim \text{Dirichlet}(\alpha_t) \tag{1}$$

$$H_t \sim \text{Multinomial}(N, \pi_t) \tag{2}$$

where π_t is the distribution of beliefs about the class of x_t over the population of human experts and α_t represent prior beliefs over the concentration of π_t. H_t can be thought of as the finite sample of experts available to query at inference time and determines ground truth via $y_t := \arg\max_k (H_t)_k$. When deciding whether to predict or query additional votes, we have access to N_t votes currently drawn as a sub-sample from H_t. Since these are drawn without replacement, this sub-sample is distributed as

$$H_t^{N_t} \sim \text{HyperGeo}_K(N, N_t, H_t) \tag{3}$$

where HyperGeo_K is a K-dimensional hypergeometric distribution of N items, N_t separate draws, and H_t contents to subsample (Johnson et al., 1997). Due to conjugacy, Eqs. (1) and (2) exhibit closed form posterior distributions in the presence of the first i votes H_t^i:

$$\pi_t \,|\, H_t^i \sim \text{Dirichlet}(\alpha_t + H_t^i) \tag{4}$$

$$\text{and } H_t^j \,|\, H_t^i \sim H_t^i + \text{DirMult}(j - i, \alpha_t + H_t^i) \tag{5}$$

where $j = i + 1, \ldots, N$ and DirMult is a compound Dirichlet-Multinomial distribution. Knowing the posterior H_t^j for when $j < N$ is useful for analyzing the potential next $j - i$ queried votes.

A Useful Approximation Our primary object of interest is H_t, which determines the consensus class y. We note that the consensus class does not change if we use the normalized proportion of votes instead of total vote counts:

$$y_t := \arg\max_{k \in \mathcal{Y}} (H_t)_k \equiv \arg\max_{k \in \mathcal{Y}} \left(\frac{H_t}{N} \right)_k \tag{6}$$

As such, we note the following limiting case as the finite number of experts N grows:

$$\begin{aligned} \frac{H_t^N}{N} \,|\, H_t^i &\sim \frac{H_t^i}{N} + \frac{1}{N} \text{DirMult}(N - i, \alpha_t + H_t^i) \\ &\xrightarrow{d} \pi_t \,|\, H_t^i \text{ as } N \to \infty \end{aligned} \tag{7}$$

for some fixed $i \in \mathbb{N}$. (See the Appendix for a proof). Thus, in the limiting case, $y_t := \arg\max_k (\pi_t)_k$ which matches what intuition would tell us. This limiting case can be used as an approximation when N is large, and potentially simplify computations. For brevity, we will refer to the exact, finite expert setting described previously as FinExp and this approximate, infinite expert setting as InfExp.

Incorporating Prior Predictions via f Determining α_t in 1, which controls our prior beliefs over the population-level distribution of classes π_t, allows incorporation of the predictions of the model f_t with the human votes. We do so in the following manner with (regularized) positive coefficients:

$$\begin{aligned} \theta &\sim \text{Gamma}(a_\theta, b_\theta) \\ \phi &\sim \text{Gamma}(a_\phi, b_\phi) \\ (\tau)_k &\sim \text{Gamma}(a_\tau, b_\tau) \text{ for } k = 1, \ldots, K \\ \alpha_t &:= \theta \cdot \text{softmax}(\tau \odot \log f_t) + \phi \end{aligned} \tag{8}$$

where $\odot$ is the element-wise product, and a and b values are hyperparameters.[1] We choose this simple transformation to allow for clear interpretation of parameter values. ϕ can be thought of as representing the base level of uncertainty concerning all classes as it determines the lower bound on α_t. θ directly quantifies how many expert votes a pretrained model's beliefs are worth. Finally, $(\tau)_k$ enables calibration of f for predictions concerning class k. High values of $(\tau)_k$ indicate trustworthy and well-calibrated predictions, and vice versa for low $(\tau)_k$ values. The full set of learnable parameters will be denoted with $\Theta := (\theta, \phi, \tau)$. Fig. 1 shows the corresponding graphical model.

Learning Objective and Optimization Let $\mathcal{D}_t := \{(f_i, H_i^{N_i})\}_{i=1}^t$ be the collection of model predictions and accumulated votes seen up to time t. To perform inference for $t+1$, we must first find the posterior for the learnable parameters: $\Theta \,|\, \mathcal{D}_t$. This allows for a more informed α_{t+1} which will influence further inference.

While this posterior could be found using Markov-chain Monte-Carlo techniques or approximated with variational inference, for computational convenience we instead work with the *maximum a posteriori* (MAP)

[1]While we restrict attention in this paper to only having access to a single pretrained model f, this setup allows for easily extending to multiple models f^i for $i = 1, \ldots, M$, each with individual model calibration, e.g., with $(\tau_i)_k \sim \text{Gamma}(a_\tau, b_\tau)$ and $\alpha_t := \theta \cdot \text{softmax}\left(\sum_i \tau_i \odot \log f_t^i\right) + \phi$.

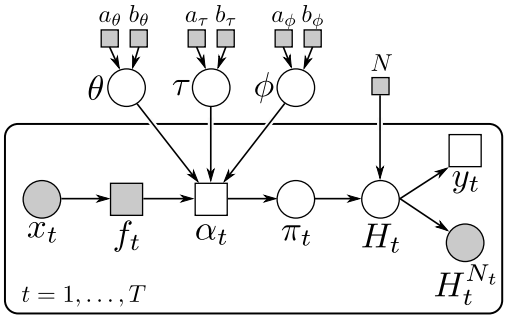

Figure 1: Graphical model of the generative process. Grey shapes are observed and assumed known, white are random and unobserved. Circles are random variables and rectangles are deterministic values / transformations. We are interested in estimating H_t for all x_t. Note that while in reality H_t is informed solely by x_t, for our purposes we instead choose to capture this relationship indirectly through π_t to facilitate how human judgement correlates with model predictions.

of Θ. Specifically, we compute

$$\Theta_t^* := \arg\min_\Theta \mathcal{L}(\Theta; \mathcal{D}_t) \tag{9}$$

$$\mathcal{L}(\Theta; \mathcal{D}_t) := \sum_{i=1}^t \log p(H_i^{N_i} \,|\, f_i, \Theta) + \log p(\Theta)$$

where $\log p(\Theta) = \log p(\theta) + \log p(\phi) + \sum_i \log(\tau)_i$. The likelihood of seeing N_i votes changes depending on whether we are using InfExp or FinExp. For the former, the likelihood is:

$$\begin{aligned} &p(H_i^{N_i} \,|\, f_i, \Theta) \\ &:= \text{DirMult}(H_i^{N_i}; N_i, \alpha_i) \\ &:= \frac{\Gamma(\overline{\alpha}_i)\Gamma(N_i + 1)}{\Gamma(\overline{\alpha}_i + N_i)} \prod_{k=1}^K \frac{\Gamma((\alpha_i + H_i^{N_i})_k)}{\Gamma((\alpha_i)_k)\Gamma((H_i^{N_i})_k + 1)} \end{aligned} \tag{10}$$

where α_i is defined in Eq. (8) and $\overline{\alpha}_i := \sum_{k=1}^K (\alpha_i)_k$.

Unfortunately, for FinExp the analytic form of $p(H_i^{N_i} \,|\, f_i, \Theta)$ involves a sum with $\binom{N-N_i+K-1}{K-1}$ terms, which can be prohibitively large to compute. To avoid this, we represent this likelihood as an expected value to facilitate Monte-Carlo estimation:

$$p(H_i^{N_i} \,|\, f_i, \Theta) := \mathbb{E}_{H_i \,|\, \alpha_i}\left[p(H_i^{N_i} \,|\, H_i)\right] \tag{11}$$

$$p(H_i^{N_i} \,|\, H_i) := \binom{N}{N_i}^{-1} \prod_{k=1}^K \binom{(H_i)_k}{(H_i^{N_i})_k}. \tag{12}$$

The resulting expectation is with respect to a discrete distribution and thus cannot be differentiably sampled from, which is required for gradient-based optimization.

To address this, we apply importance sampling with a proposal distribution q that does not rely on the parameters of interest. Leveraging currently seen votes $H_i^{N_i}$ may result in lower estimator variances by biasing samples towards similar proportions of values. However, all possible values of H_i (that do not result in $p(H_i^{N_i} \,|\, H_i) = 0$) must be present in the support of q to ensure valid importance sampling. This leads to the following proposal distribution:

$$H_i \sim_q H_i^{N_i} + \text{Multinomial}\left(N - N_i, \frac{H_i^{N_i} + 1}{N_i + K}\right) \quad (13)$$

where we ensure all classes have support by ensuring the multinomial probability vector has all non-zero values adding 1 to $H_i^{N_i}$ prior to normalizing. Applying importance sampling with this proposal distribution to Eq. (11) yields the following form:

$$p(H_i^{N_i} \,|\, f_i, \Theta) = \mathbb{E}^q_{H_i}\left[\frac{p(H_i \,|\, \alpha_i)}{q(H_i \,|\, H_i^{N_i})} p(H_i^{N_i} \,|\, H_i)\right], \quad (14)$$

where $p(H_i \,|\, \alpha_i) := \text{DirMult}(H_i; N, \alpha_i)$ and $\mathbb{E}^q$ is the expectation with respect to the proposal distribution q.

Thus, with M Monte-Carlo samples, the likelihood can be approximated via

$$p(H_i^{N_i} \,|\, f_i, \Theta) \approx \frac{1}{M} \sum_{j=1}^{M} \frac{p(H_i^{(j)} \,|\, \alpha_i)}{q(H_i^{(j)} \,|\, H_i^{N_i})} p(H_i^{N_i} \,|\, H_i^{(j)})$$

where $H_i^{(j)} \sim q(H_i \,|\, H_i^{N_i})$ for $j = 1, \ldots, M$. Computing this for $i = 1, \ldots, t$ allows for the computation of $\mathcal{L}(\Theta; \mathcal{D}_t)$ in a differentiable manner.

Decision-Making Assume that the MAP estimate for $\Theta \,|\, \mathcal{D}_{t-1}$ has been found, resulting in $\alpha_t^* = \theta^* \cdot \text{softmax}(\tau^* \odot \log f_t) + \phi^*$. For generality, also assume we have already seen $N_t > 0$ votes for the current time step t, and are deciding whether to continue querying new votes or to make a prediction for y_t.

The model has beliefs over the true consensus class y_t for x_t. In FinExp, it follows that:

$$\begin{aligned} p(y_t = k \,|\, H_t^{N_t}, f_t, \mathcal{D}_{t-1}) &\approx p(y_t = k \,|\, H_t^{N_t}, \alpha_t^*) \\ &= p(\arg\max_{k'}(H_t)_{k'} = k \,|\, H_t^{N_t}, \alpha_t^*) \\ &= \mathbb{E}_{H_t \,|\, H_t^{N_t}, \alpha_t^*}\left[\mathbb{1}(\arg\max_{k'}(H_t)_{k'} = k)\right] \quad (15) \end{aligned}$$

where $\mathbb{1}(\cdot)$ is the indicator function, returning 1 if the argument is true and 0 if false.[2] This expected value can be approximated using a Monte-Carlo estimate. For InfExp, the formula is the same, aside from swapping H_t for π_t. This is justified using the same reasoning as to the existence of the limiting case in the first place, as demonstrated in Eq. (7).

Given current information, choosing the $\hat{y}_t$ with the highest likelihood, i.e., $\hat{y}_t := \arg\max_k p(y_t = k \,|\, H_t^{N_t}, \alpha_t^*)$ represents the optimal decision under the posterior. Additionally, assuming our model is well-calibrated, $p(y_t = \hat{y}_t \,|\, H_t^{N_t}, \alpha_t^*)$ can be interpreted as the expected accuracy of our prediction. This probability will be denoted as $\text{Acc}_t(H_t^{N_t})$. One obvious choice of heuristic for determining when to stop querying is to simply predict once the accuracy has cleared some set threshold: $\text{Acc}_t(H_t^{N_t}) > \rho$. We utilize this threshold decision on estimated accuracy as our heuristic for experiments detailed in Section 4.

4 EXPERIMENTS AND RESULTS

We evaluate our online consensus prediction methods on two large-scale datasets that include multiple per-sample human predictions. For each, samples x_t are drawn randomly from the test set without replacement to create a sequence. At a given timestep t, each method is given predicted model confidences f_t for sample x_t and the opportunity to query expert votes h_t or make a final prediction. Expert votes can be queried sequentially, allowing methods to re-evaluate under new information. Once the method ceases querying, it then generates a final prediction and proceeds to the next timestep. Under such scenarios, we evaluate the error rate of a particular method, evaluated relative to the true human consensus, across a range of cost budgets and with several different pre-trained neural network models f of varying accuracy relative to human performance. Overall, the experiments systematically demonstrate improved predictive performance for the proposed methods compared to baselines, as well as robustness to distribution shift.

4.1 Datasets

For our experiments we use CIFAR-10H (Peterson et al., 2019), an extended test dataset from (Krizhevsky and Hinton, 2009) that contains 10,000 natural images categorized into 10 classes. Each image in CIFAR-10H also includes 50 human predictions, allowing us to explore multiple expert pool sizes by first randomly selecting $N \leq 50$ experts and then generating randomly ordered sequences of length 10,000 from the test set. We present results for $N = 3$ and $N = 50$, with additional results in the Appendix. Consensus is determined from this sampled expert pool, with ties broken randomly. We also train several ResNet18 models (He et al., 2016)

[2] In the case of a simulated H_t resulting in a tie, we randomly choose one of the tied classes to be the arg max, in the same manner as ground truth is determined.

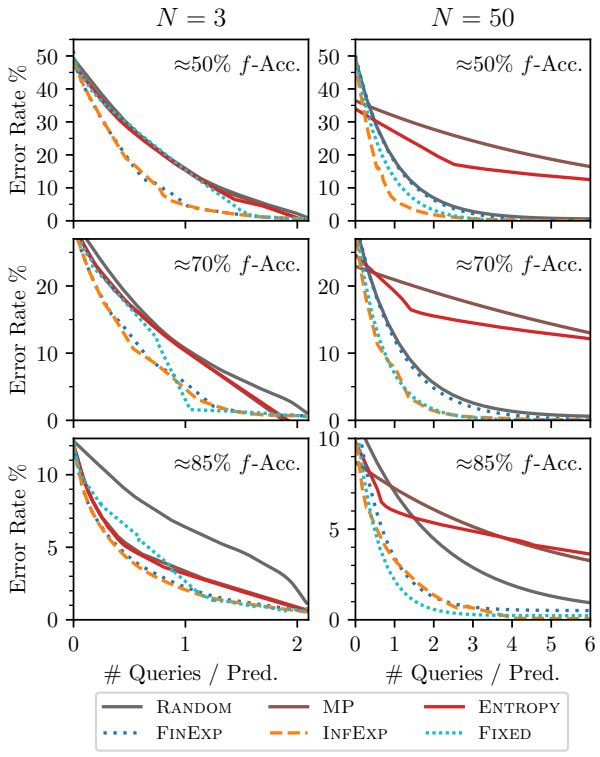

Figure 2: Error-cost curves on CIFAR-10H with $N = 3, 50$ and model accuracies of 50%, 70%, and 85%. Solid and dotted curves are baselines and proposed methods.

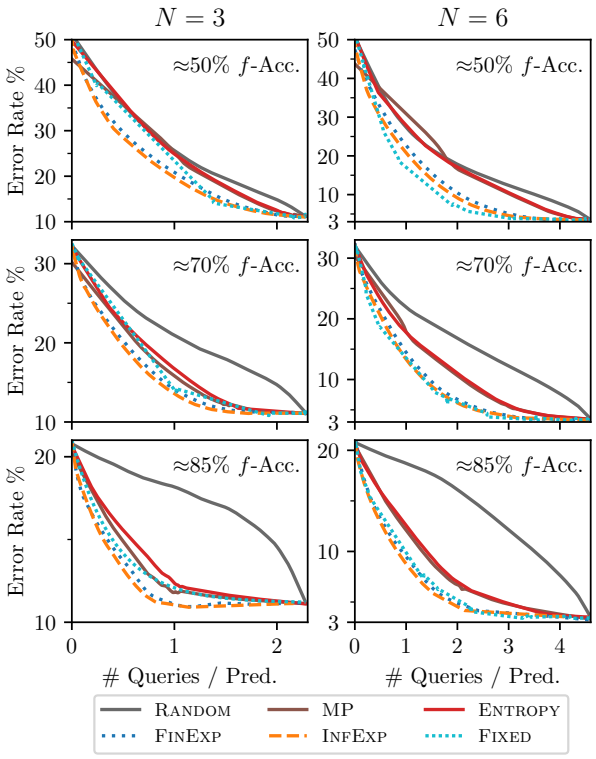

Figure 3: Error-cost curves similar to Fig. 2 except on ImageNet-16H with $N = 3, 6$.

to utilize as f, using subsets of the CIFAR-10 training dataset with varying levels of class imbalance, yielding models with significantly different error rates.

As a second dataset in our experiments we use the ImageNet-16H dataset (Steyvers et al., 2022), which contains 4800 natural images grouped into 16 classes with 4 different levels of phase noise (none, low, medium, and high noise, 1200 images each). In the original work by Steyvers et al. (2022) 5 different classifiers pre-trained on the ImageNet-16H training data with varying error rates. Four variants of each of these classifiers were then trained for 0, 0.5, 1, or 10 epochs on noised training data, increasingly improving model performance. In this dataset, each image is annotated by 6 experts. Similar to CIFAR-10H, we present results for $N = 3$ and $N = 6$, and we define y similarly as the consensus of available experts.

4.2 Baselines and Methods

We compare against several baselines motivated by the active learning and online prediction literature and adapt these methods to accommodate online consensus prediction. As a simple baseline, we utilize a random (RANDOM) predictor. This method is parametrized by a single global Bernoulli parameter β and ignores model predictions f_t when querying. For each example x_t, we draw $\text{Binom}(N, \beta)$ number of expert votes and predict the arg max of them. Should no experts be queried, then $\arg\max_k (f_t)_k$ is used as a prediction. The other two baselines both use model predictions to influence β on a per-example basis. The ENTROPY baseline uses the prediction entropy $H(f_t)$ of the model's predictions, scaled and restricted to the range $[0, 1]$, to define β_t, which is then used in the same manner as the random baseline. The third baseline is based on the Model Picker method from Karimi et al. (2021) (MP), which adapts the Exponential Weights algorithm Littlestone and Warmuth (1994) to dynamically maintain a loss-estimate over several models. Since we utilize a single model in our evaluation, we tailor MP to track per-class loss estimates. The method uses approximate loss estimates over classes plus the model's label distribution for sample x_t to aggregate into a single Bernoulli parameter β_t. Queries are then determined in the same manner as the ENTROPY and RANDOM baselines. Additional details can be found in the Appendix.

In the context of our proposed framework, we apply FINEXP and INFEXP in the following manner. We learn a MAP estimate of Θ and leverage it to create α_t^* via Eq. (8). For each sample x_t, querying is de-

cided by thresholding $\max p(y_t = k \mid H_t^{N_t}, \alpha_t^*)$ with hyperparameter ρ. If an expert is queried, we then update our observed expert votes $H_t^{N_t}$ and repeat the process. Otherwise, $\hat{y} := \arg\max_k p(y_t = k \mid H_t^{N_t}, \alpha_t^*)$ is submitted as the final prediction.

In addition to FinExp and InfExp, we seek to isolate the impact of our learned MAP estimates from the inference procedure. In turn, we fix the prior parameters $\theta = \phi = \tau = 1$ and do not update them during inference. We will refer to this method as Fixed. Here we only present results for Fixed under the FinExp inference procedure and refer the reader to the Appendix for the InfExp equivalent. Finally, we note several computational speedups that yield a negligible performance difference with the naive implementation. We utilize the latter in all experiments and note a 20x speedup. All experiments were conducted on NVIDIA GeForce 2080ti GPUs over roughly 4 days. Please see the Appendix for further details.

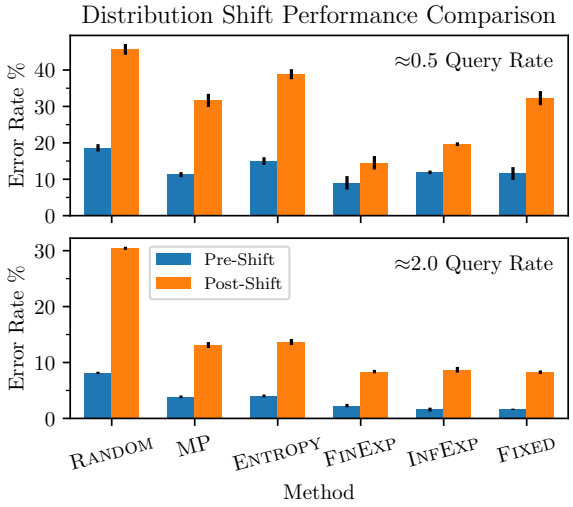

Figure 4: Visualization of aggregate distribution shift error rate on ImageNet-16H with a DenseNet model f for budgets of 0.5 and 2 queries per sample.

4.3 Results: Error-Cost Curves

Using the methodology described above, we generate error-cost curves for each of our proposed methods and baselines, for each dataset, for models of different accuracies. Given a particular randomly-ordered sequence from the test set, each method sequentially handles querying and prediction for each example x_t and the overall per-sample querying cost and error rate are then plotted as a single point, using a default cost of 1 for querying a single human expert. By sweeping over hyper-parameter settings and 3 trials, we generate an error-cost curve that reflects expected error rates for a range of budgets. Please see the Appendix for additional implementation details.

We plot error rate as a function of querying budget, defined as the average number of queries taken per sample. For each plot, the lower-limit of the y-axis represents an empirical lower bound on error rate due to ties in consensus. Figures 2 and 3 show the cost-error results for CIFAR-10H and ImageNet-16H, respectively. We evaluate all methods on different sizes of N using three different models with approximately 50%, 70%, and 85% accuracy relative to consensus ground truth. For ImageNet-16H we see that the FinExp and InfExp methods consistently outperform all baselines across all budget settings with all models. In some cases, Fixed also outperforms the baselines, even achieving the outright best performance with $N = 6$ and base model accuracy of 50%. However, the performance of Fixed is generally inconsistent across settings.

These trends continue when analyzing results from CIFAR-10H in Fig. 2 with dramatic improvements observed for InfExp over all settings. FinExp also outperforms all baselines, though by a very small margin over the Random baseline for certain settings where $N = 50$. In general we found InfExp to be more consistent than FinExp across different settings. We conjecture these results as a whole to be attributable to both the online calibration of the model predictions through parameters τ for FinExp and InfExp, and also to tying our decision to query (based on the model's beliefs) directly to expert consensus through $\hat{y}_t := \arg\max_k (H_t)_k$ (or $(\pi_t)_k$ for InfExp).

4.4 Results: Out-of-Distribution Performance

To further explore the robustness of our methods under distribution shift, we refined the ImageNet-16H dataset such that each test sequence consists of 1200 non-noisy images (in random order) followed by a random sequence of 1200 high-noise images. We conduct online consensus prediction as before for our cost-error models, using the five pre-trained models that never observed noisy data. This in effect induces a distribution shift in the test sequences after 1200 samples, causing models to suddenly encounter much noisier images than those on which it was trained.

To compare our results, we run all methods and baselines across a hyperparameter sweep and multiple trials. Then, we group all runs that fall within 10% of 0.5, 1, 2, and 3 average querying costs to produce aggregate results. Shown in Fig. 4, our proposed methods possess lower error rates and demonstrate significantly lower performance degradation after the distribution shift occurs, even when pre-shift error rates are comparable.

Fig. 5 more granularly explores the distribution-shift

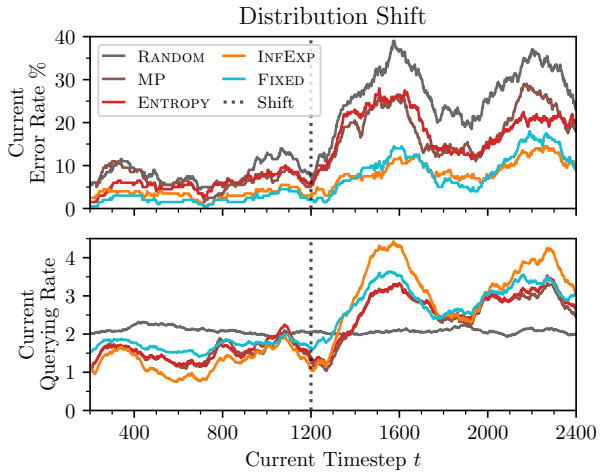

Figure 5: Visualization of distribution shift error rate and querying cost on ImageNet-16H with a DenseNet model f and an average querying cost of 2 experts per sample. Distribution shift occurs at $t = 1200$, increasing uncertainty in model predictions and expert consensus.

behavior of all methods by focusing only on a subset of runs that all possess an average cost of 2 queries per sample. Both before and after distribution shift, InfExp and Fixed consistently outperform the baselines. Interestingly, though InfExp and Fixed maintain nearly identical error rates, InfExp has the *lowest* pre-shift and the *highest* post-shift querying costs, demonstrating the flexibility conferred by dynamically learning a prior under distribution shift. This is also highlighted by visualizing the τ parameters against model performance, as shown in Fig. 6. To avoid clutter, we only plot the results of InfExp and Fixed, with further results provided in the Appendix. Due to stochasticity in the sequence ordering as well as the effect of the moving average, we note fluctuations performance over time for all methods. We explore these fluctuations for additional properties like distribution shift convergence in the Appendix by averaging over many sequence orderings for the same budget.

4.5 Results: Calibration and Query-Free Prediction

In a practical setting, there may be situations where human experts will be unavailable, such as nights, weekends, holidays, or periods where everyone is busy. We explore these scenarios with a suite of two-phase experiments where we first query and predict as normal but then enter a second phase where no human experts are available to query and we must solely rely on the learned parameters and pretrained model predictions. We use both InfExp and FinExp as described for both querying and predicting. We also consider the scenario of using the Fixed protocol for querying while learning and predicting with InfExp and FinExp. We refer to these options as Fixed-InfExp and Fixed-FinExp respectively. Results can be seen in Table 1. We find a small but consistent average performance increase of up to 3% over the base model accuracy in many settings, but particularly in cases where the querying budget is large and base model performance is low. This demonstrates the ability for the MAP parameter estimates to learn per-class model performance through noisy expert votes and calibrate the pretrained model's predictions.

Table 1: Query-free accuracy improvement over base model f under different model accuracy and querying budget settings for CIFAR-10H.

Queries / Sample:		1	2	3
Acc.	Method	Acc. Improvement Over f		
50%	InfExp	0.99±0.08	1.01±0.09	1.09±0.14
	Fix-InfExp	1.21±0.11	1.10±0.06	0.98±0.15
	FinExp	1.45±0.17	1.33±0.13	1.98±0.22
	Fix-FinExp	**1.59±0.34**	**2.79±0.08**	**3.29±0.23**
70%	InfExp	0.77±0.08	0.63±0.04	0.59±0.11
	Fix-InfExp	0.81±0.06	0.69±0.03	0.5±0.03
	FinExp	**1.05±0.08**	0.83±0.08	1.08±0.06
	Fix-FinExp	0.90±0.11	**1.07±0.03**	**1.24±0.07**
85%	InfExp	0.13±0.02	0.07±0.01	0.07±0.02
	Fix-InfExp	0.11±0.01	0.09±0.00	0.13±0.02
	FinExp	0.15±0.04	**0.16±0.06**	**0.32±0.03**
	Fix-FinExp	**0.15±0.02**	0.15±0.01	0.24±0.04

To visualize this capability, we plot the performance of a pretrained model with extremely varied per-class accuracy on CIFAR-10H against the learned τ values of FinExp and InfExp, shown in Fig. 6. In addition to correlating strongly with the true consensus predictions, we observe that FinExp more heavily weights its beliefs with new evidence, while InfExp, which assumes a pool of experts, does not. This can be seen by noting the increased values in the blue curves from solid to dotted/dashed (especially for classes 7, 8, and 9), while the different orange curves remain all roughly the same. Please see the Appendix for a more thorough analysis.

5 DISCUSSION AND CONCLUSION

Limitations Our experimental results rely only on two crowdsourced datasets from a single data modality (images). Nonetheless, the strength of our empirical findings offer a robust starting point for future work in online consensus prediction.

Future Directions The work in this paper leaves multiple future directions to explore. For example, we did not consider adversarial data streams, non-stationary streams, or open-set tasks where new classes

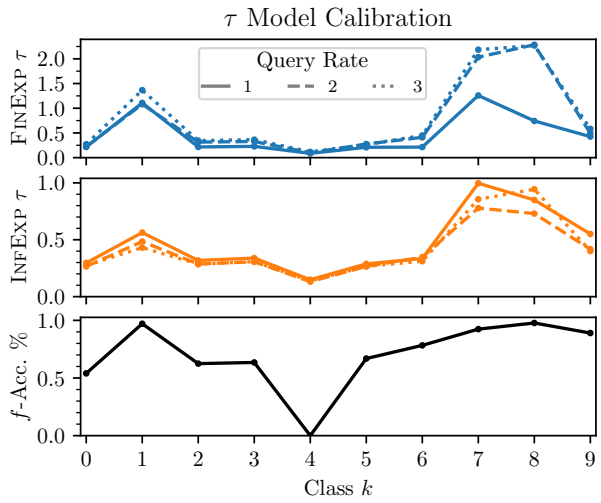

Figure 6: Visualization of per-class performance of a CIFAR-10 model with 70% accuracy under learned τ parameters from FinExp and InfExp with different querying budget rates.

may be progressively observed. Exploring these scenarios is a promising opportunity for future work. Further, allowing experts to be identifiable, ranked (e.g. by seniority), and variable in cost and/or availability offers new dimensions to consensus prediction.

Conclusion In this paper we proposed the novel and practical problem of online human consensus estimation, where the target y is defined as the consensus of a pool of human experts. We approached this problem in a Bayesian fashion, offering a principled means of reasoning about human expert consensus given a subset of votes and a pre-trained model f with unknown performance. We derived FinExp, based on the multivariate hypergeometric likelihood, and also introduced the InfExp method, a computationally simple limiting case of FinExp when $N \to \infty$. Empirically, we evaluated FinExp and InfExp against several baselines, including Fixed, a variant of FinExp with a fixed prior. Our results demonstrate the promise of FinExp and InfExp both in standard experimental settings as well as with distribution shift.

ACKNOWLEDGEMENTS

We thank the reviewers for their suggestions on improving the paper. This work was supported by National Science Foundation Graduate Research Fellowship grant DGE-1839285 (SS and AB), by the National Science Foundation under award number 1900644 (PS and MS), by the National Institute of Health under awards R01-AG065330-02S1 and R01-LM013344 (PS), by the HPI Research Center in Machine Learning and Data Science at UC Irvine (SS and PS), and by Qualcomm Faculty awards (PS).

References

Beck, M. R., Scarlata, C., Fortson, L. F., Lintott, C. J., Simmons, B., Galloway, M. A., Willett, K. W., Dickinson, H., Masters, K. L., Marshall, P. J., et al. (2018). Integrating human and machine intelligence in galaxy morphology classification tasks. *Monthly Notices of the Royal Astronomical Society*, 476(4):5516–5534.

Bien, N., Rajpurkar, P., Ball, R. L., Irvin, J., Park, A., Jones, E., Bereket, M., Patel, B. N., Yeom, K. W., Shpanskaya, K., Halabi, S., Zucker, E., Fanton, G., Amanatullah, D. F., Beaulieu, C. F., Riley, G. M., Stewart, R. J., Blankenberg, F. G., Larson, D. B., Jones, R. H., Langlotz, C. P., Ng, A. Y., and Lungren, M. P. (2018). Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of mrnet. *PLOS Medicine*, 15(11):1–19.

Branson, S., Van Horn, G., and Perona, P. (2017). Lean crowdsourcing: Combining humans and machines in an online system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6109–6118.

Choudhary, V., Marchetti, A., Shrestha, Y. R., and Puranam, P. (2023). Human-AI ensembles: When can they work? *Journal of Management*, page 01492063231194968.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Jarrett, D., Hüyük, A., and van der Schaar, M. (2022). Online decision mediation. *Advances in Neural Information Processing Systems*, 35:1790–1805.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1997). *Discrete Multivariate Distributions*, volume 165. Wiley New York.

Karimi, M. R., Gürel, N. M., Karlaš, B., Rausch, J., Zhang, C., and Krause, A. (2021). Online active model selection for pre-trained classifiers. In *International Conference on Artificial Intelligence and Statistics*, pages 307–315. PMLR.

Kerrigan, G., Smyth, P., and Steyvers, M. (2021). Combining human predictions with model probabilities via confusion matrices and calibration. *Advances in Neural Information Processing Systems*, 34:4421–4434.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, 108(2):212–261.

Lubars, B. and Tan, C. (2019). Ask not what AI can do, but what AI should do: Towards a framework of task delegability. volume 32, pages 57–67.

Madras, D., Pitassi, T., and Zemel, R. (2018). Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31, pages 6150–6160.

Meresht, V. B., De, A., Singla, A., and Rodriguez, M. G. (2022). Learning to switch among agents in a team. *Transactions on Machine Learning Research*, 7:1–30.

Mozannar, H. and Sontag, D. (2020). Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 7076–7087. PMLR.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. (2019). Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9617–9626.

Pinski, M., Adam, M., and Benlian, A. (2023). AI knowledge: Improving AI delegation through human enablement. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Plank, B. (2022). The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682.

Rajpurkar, P., O'Connell, C., Schechter, A., Asnani, N., Li, J., Kiani, A., Ball, R. L., Mendelson, M., Maartens, G., van Hoving, D. J., et al. (2020). CheXaid: deep learning assistance for physician diagnosis of tuberculosis using chest X-rays in patients with HIV. *NPJ Digital Medicine*, 3(1):1–8.

Sayin, B., Krivosheev, E., Yang, J., Passerini, A., and Casati, F. (2021). A review and experimental analysis of active learning over crowdsourced data. *Artificial Intelligence Review*, 54:5283–5305.

Sheng, V. S. and Zhang, J. (2019). Machine learning with crowdsourcing: A brief summary of the past research and future directions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9837–9843.

Steyvers, M., Tejeda, H., Kerrigan, G., and Smyth, P. (2022). Bayesian modeling of human–AI complementarity. *Proceedings of the National Academy of Sciences*, 119(11):e2111547119.

Stutz, D., Cemgil, A. T., Roy, A. G., Matejovicova, T., Barsbey, M., Strachan, P., Schaekermann, M., Freyberg, J., Rikhye, R., Freeman, B., et al. (2023). Evaluating AI systems under uncertain ground truth: a case study in dermatology. *arXiv preprint arXiv:2307.02191*.

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. (2019). Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11244–11253.

Uma, A. N., Fornaciari, T., Hovy, D., Paun, S., Plank, B., and Poesio, M. (2021). Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Van Horn, G., Branson, S., Loarie, S., Belongie, S., and Perona, P. (2018). Lean multiclass crowdsourcing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2723.

Verma, R. and Nalisnick, E. (2022). Calibrated learning to defer with one-vs-all classifiers. In *International Conference on Machine Learning*, pages 22184–22202. PMLR.

Wright, D. E., Fortson, L., Lintott, C., Laraia, M., and Walmsley, M. (2019). Help me to help you: machine augmented citizen science. *ACM Transactions on Social Computing*, 2(3):1–20.

Wright, D. E., Lintott, C. J., Smartt, S. J., Smith, K. W., Fortson, L., Trouille, L., Allen, C. R., Beck, M., Bouslog, M. C., Boyer, A., et al. (2017). A transient search using combined human and machine classifications. *Monthly Notices of the Royal Astronomical Society*, 472(2):1315–1323.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable]
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable]
   (b) The license information of the assets, if applicable. [Yes/No/Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable]
   (d) Information about consent from data providers/curators. [Yes/No/Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable]

Appendix: Bayesian Online Learning for Consensus Prediction

A Proof for InfExp being a Limiting Case of Section 3 in paper

Below, we outline and prove that the limiting case of FinExp is InfExp as $N \to \infty$.

For $\theta \sim \text{Dirichlet}(\alpha)$ with K dimensions and $h_i \mid \pi \sim \text{Multinomial}(1, \pi)$ for $i = 1, 2, \ldots,$ let $H^n := \sum_{i=1}^{n} \texttt{one-hot}(h_i)$ for $n > 0$ and $H^0 = \vec{0}$. It follows that $H^n \mid \pi \sim \text{Multinomial}(n, \pi)$, and that $\theta \,|\, H^n \sim \text{Dirichlet}(\alpha + H^n)$. H^n also is known to have an unconditional distribution of $H^n \sim \text{DirMult}(n, \alpha)$ and posterior $H^N \mid H^n \sim H^n + \text{DirMult}(N - n, \alpha)$ for $N > n$. Note that this is shorthand for denoting $H^N - H^n \,|\, H^n \sim \text{DirMult}(N - n, \alpha)$.

The posterior distribution of the normalized counts, H^N/N, converges in distribution to the posterior of π. Put formally:

$$\begin{aligned} \frac{H^N}{N} \,|\, H^n &\sim \frac{H^n}{N} + \frac{1}{N}\text{DirMult}(N - n, \alpha + H^n) \\ &\xrightarrow{d} \pi \,|\, H^n \text{ as } N \to \infty \end{aligned} \tag{16}$$

for some fixed $n \in \mathbb{N}$

Proof: First, we note that we are interested in the limiting case of the random variable $\frac{H^N}{N} - \frac{H^n}{N} \,|\, H^n$. Since we condition on H^n, $\frac{H^n}{N}$ can be treated as a constant term with respect to N, which converges to 0 as $N \to \infty$. Thus, analyzing the limiting case of $\frac{H^N}{N} \,|\, H^n$ is sufficient.

Next, we denote the joint distribution of $H^N, \pi \mid H^n$ as $p(H^n, \pi \mid H^n) = p(\pi \mid H^n)p(H^N \mid \pi) := \text{Dir}(\pi; \alpha + H^n)\text{Mult}(H^N; \pi)$. The distribution of $\pi \mid H^n$ can be treated as a constant with respect to N. Lastly, $\frac{H^N}{N} \,|\, \pi = \frac{1}{N}\sum_{i=1}^{N} h_i \,|\, \pi \xrightarrow{d} \pi \,|\, \pi$ as $N \to \infty$ due to the law of large numbers. Thus we can conclude that $\frac{H^N}{N}, \pi \,|\, H^n \xrightarrow{d} \pi \,|\, H^n$ as $N \to \infty$ which implies that $\frac{H^N}{N} \,|\, H^n \xrightarrow{d} \pi \,|\, H^n$.

B Experimental Details (Section 4 in paper)

B.1 Datasets

CIFAR-10H The CIFAR-10H dataset Peterson et al. (2019) consists of 10,000 samples corresponding to the test-set of CIFAR-10 Krizhevsky and Hinton (2009), with 50 human annotations per label. On average, each individual human annotator only labeled a few hundred of the test samples: as a result, 50 "full" annotators were synthetically generated by combining annotations from multiple human annotators.

Model predictions utilized for CIFAR-10H experiments were generated by training ResNet-18 models on random sub-slices of the training dataset with stochastic gradient descent and a learning rate of 0.01. To get a variety of models, checkpoints were stored every 10 epochs and leveraged to provide models with different levels of test-set accuracy. Predictions for these models are included in the code repository noted in the main paper.

ImageNet-16H The ImageNet-16H dataset used in our experiments includes model predictions provided with the original ImageNet-16H paper Steyvers et al. (2022). These predictions were generated by selecting 5 pre-trained models (AlexNet, DenseNet, VGG-16, ResNet, and GoogLeNet) on the original ImageNet dataset and renomalizing the prediction probabilities over 16 classes. 1200 images belonging to these 16 classes (300 each) were stored and then perturbed with 3 different levels of phase noise, also preserving the original. In total, this created 4800 images. For each image, 6 human annotations are available. Each of the 5 models is then fine-tuned for 0, 0.5, 1, and 10 epochs on noised equivalents of the training dataset, totaling 20 models. Each model produced a prediction for each image. The dataset provided in the repository noted in the main paper includes model predictions to allow for easy experiment replication.

B.2 Baselines

Random The RANDOM baseline determines how many experts to query per example as a function of the hyperparameter β. Given a fixed pool of N experts, the random baseline queries $Q \sim \text{Binom}(N, \beta)$ experts one-at-a-time, stopping early if consensus is reached.

Entropy The ENTROPY baseline first computes the prediction entropy from a pre-trained model f as $\mathcal{H}(f_t) := -\frac{1}{K}\sum_{i=1}^{K}(f_t)_i \log(f_t)_i$. A tuning hyperparameter $v \in \mathbb{R}_+$ is then applied and the per-sample query parameter β_t is determined as $\beta_t := \max(\min(H(f_t), 1), 0)$, i.e., the entropy is clipped to the $[0, 1]$ range. The number of queries is then determined in the same manner as for the RANDOM baseline.

Model Picker As noted in Karimi et al. (2021), Model Picker (MP) is a context-free online-active model selection algorithm. We adapt MP to the case of a single model by determining variance calculations over the classes and updating per-class belief accordingly. All other features of MP are left unchanged, except that instead of sampling a single ground truth oracle, MP now samples the number of queried experts from a binomial in the same fashion as the RANDOM and ENTROPY baselines.

Learning MAP Parameter Estimates In our experiments we utilize Gamma distributions $\Gamma(a, b)$ as (hyper-)priors over our learnable parameters $\Theta := \{\theta, \phi, \tau\}$. All learnable parameters utilize the same (relatively flat) prior distribution and are initialized to the mode of the prior before optimization begins. We compute MAP estimates by use of stochastic gradient descent and the Adam optimizer with a learning rate of 0.1. For a given timestep t, we define the dataset on which we learn our prior as $\mathcal{D} := \{f_{t'}, H_{t'}^{N_{t'}} | t' < t, N_{t'} > 0\}$. In words, we consider all observed data from previous samples, but filter out samples for which we did not query a single expert, as these samples do not contribute to learning MAP parameters. Furthermore, for computational feasibility over long sequences, we further reduce $\mathcal{D}$ to only the last w observed samples, creating a sliding window on which we learn MAP estimates. For our experiments, w is set to 500. In addition, rather than learn MAP estimates for each new sample, we learn them at a fixed interval of 20 iterations for all experiments, noting a negligible difference in converged values. Training is conducted for 1000 iterations or until the maximum difference in the parameter values of Θ is less than a tolerance of 0.01 between updates across 10 iterations. In general, we note that, with the exception of distribution shift experiments, after a few hundred iterations the values of Θ appear to converge and experiments accelerate.

For FINEXP experiments, a finite number of experts is assumed in the likelihood, regularizing the method more so that the INFEXP counterpart, which assumes an expert pool of infinite size. To reflect this, we utilize priors of $\Gamma(1.1, 1)$ and $\Gamma(3, 2)$, for FINEXP and INFEXP, respectively. This is to more heavily constrain the INFEXP prior parameters under the potential for infinite feedback. These settings were applied to all experiments.

B.3 Experiment: Error-Cost Curves

In addition to the discussion of error-cost curves provided in the main paper, we include the following additional details to better inform experiment replication. First, a dataset is loaded and a subset of experts is selected from the population pool. A sequence of samples is then drawn from the dataset without replacement and placed in random order. Ground truth is subsequently defined with regard to all experts in the subset, with ties broken randomly. We adjust the origin of the y-axis of the plots in both the main paper and the appendix to reflect the empirical lower bound on error rate due to ties. This is particularly noticeable in ImageNet-16H. We utilize random seeds 3,4, and 5 to seed all randomness in our experiments. In addition, we evaluate a range of

hyperparameter settings for each method to generate full power-cost curves.

Once ground truth is computed, we begin experiments by sweeping across hyperparameters for each method. The base ranges for hyperparameters is listed below:

- FinExp, InfExp, Fixed- $[0, 1]$
- RANDOM - $[0, 1]$
- ENTROPY - $[0, 1000]$
- MP - $[0, 1000]$

All of these hyperparameter values are multiplied by the original β coefficients for binomial sampling and then clamped to be within the 0-1 range to create a final β parameter for sampling. Both the ENTROPY and MP baselines may possess very small values β_t. To ensure the final β parameter fully spans the 0-1 range (and therefore all querying budgets), we sweep across the larger range of 0 to 1000 and clamp values to 1 in cases where the scaled value exceeds this range.

For each error-cost curve in the main paper in Fig. 2 and Fig. 3, we fit a Lowess smoother with a weight fraction of 0.20 to the resulting scatter plots to visualize a smoothed error-cost curve for all querying budgets. For each plot, we set the domain of the error-cost curves such that methods achieve the empirical lower bound on performance. That bound is 10.03% for ImageNet-16H and $N = 3$ experts, 2.92% for the same dataset and $N = 6$ experts. The error lower bound for CIFAR-10H, a much less noisy dataset, is less than 0.50% for all settings of N.

B.4 Experiment: Distribution Shift

Distribution shift experiments combine the 1200 clean (un-noised) samples from ImageNet-16H with most-noised samples. All methods are then run on this sequence, which randomly orders the first and second 1200 samples but keeps all of the clean samples in the first half of the experiment to make the distribution shift abrupt. We plot all results, shown in Fig. 5 in the main paper and additional results in Figs. 9 and 10, by utilizing a simple moving average to produce running estimates of error rate and querying cost. Aggregate data explores these trends further in Figure 4, grouping runs that achieve within 0.1 queries/sample of a budget of 0.5 and 2 experts per sample. Evaluation of additional budgets can be found in the additional experiments below.

B.5 Experiment: Two-Phase Evaluation

In two-phase evaluation experiments, we take the first 1000 data points from a sequence generated from either the CIFAR-10H or ImageNet-16H dataset. Given our method of choice, we query experts as needed for these first 1000 samples. At this point, access to expert feedback stops and methods must predict using just the base model predictions and the MAP parameters of Θ that they learned on the previously observed data. In this case, we do not utilize a sliding window and instead learn a prior over all of the observed data. We then group experiments that are within 0.1 queries/sample of querying budgets of 1, 2, and 3 experts per sample, producing the mean and standard error or the runs. Our results demonstrate that our proposed methods, FinExp, InfExp, and the Fixed method combined with either the FinExp or InfExp inference method (known as Fix-FinExp and Fix-InfExp), all consistently (but slightly) outperform the base model accuracy. These metrics are all relative to the base model accuracy.

B.6 Experiment: Model Calibration

Exploring the results of the two-phase evaluation further, we select a model from CIFAR-10H with highly varied per-class performance. Several different experiments were conducted using this model, and reported in the main paper is a non-cherry-picked set of results for τ under budgets of 1, 2, and 3 experts per sample. These are plotted against the base model accuracy. In the additional experiments below, we include the aggregate correlations between the learned MAP parameters of FinExp and InfExp and display these results in the following section.

C Additional Experimental Results (Sections 4.3 - 4.5 in paper)

Below, we produce and examine experimental results not reported in the main paper. While the general findings are consistent, we note additional details and insights that extend on our initial results.

Results: Error-Cost Curves (Section 4.3 in paper) In addition to creating error-cost curves for experiments on CIFAR-10H with $N = 3, 50$, we also include findings for $N = 10$. Seen in Figure 7, we witness similar trends to $N = 50$. However, we observe no degradation in performance in FinExp across runs, whereas with $N = 50$ FinExp approaches the random baseline for model accuracies of 50% and 70%.

At the same time, we include the results for the Fix-InfExp baselines, shown in all experimental settings but omitted from the main paper due to its comparable performance to Fixed, which we refer to here as Fix-FinExp here for clarity. These represent baselines where no prior parameters are learned but either the InfExp or FinExp procedure is still utilized. In general, our findings remain consistently superior to the baselines, with InfExp offering the most consistent performance overall.

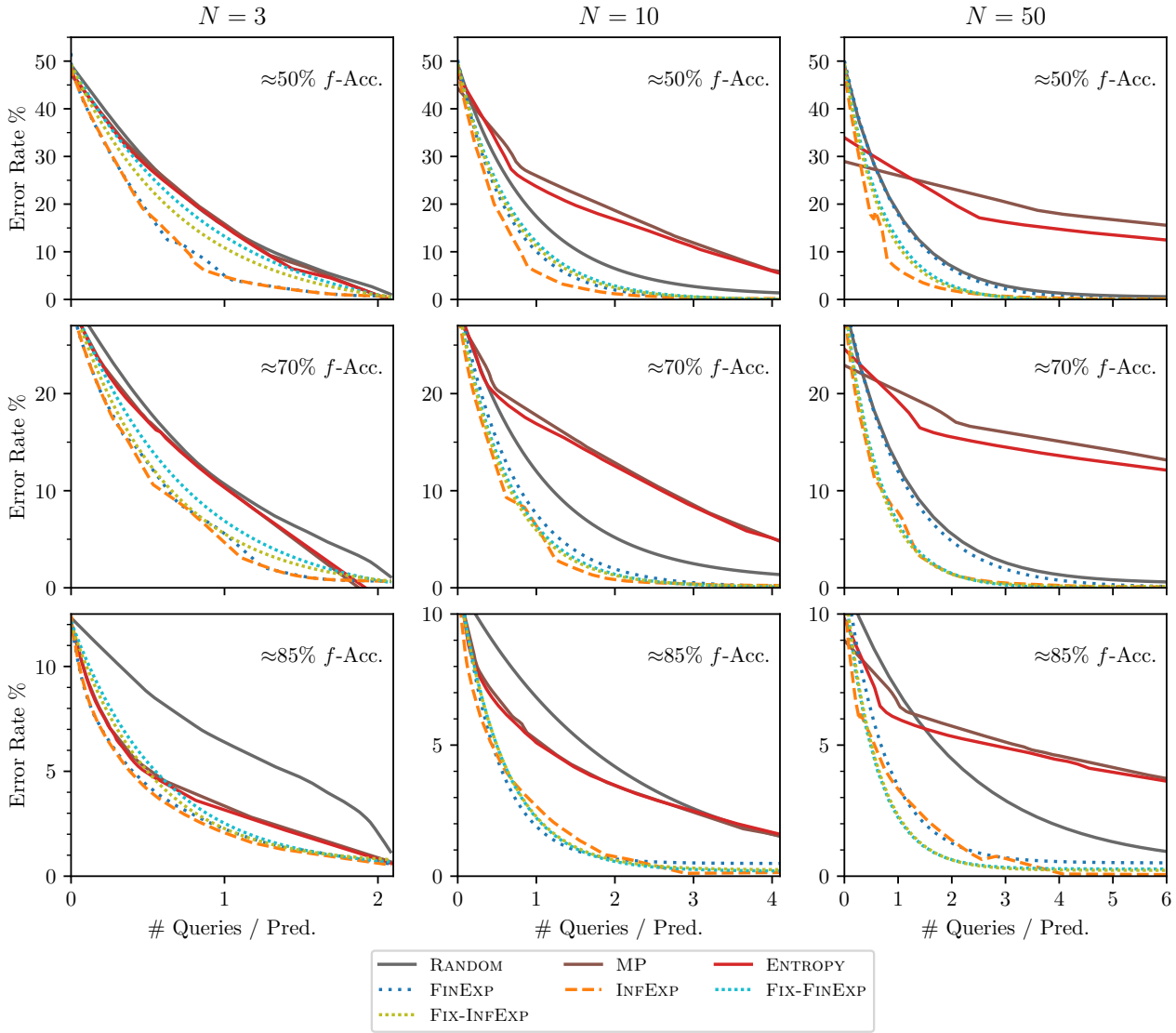

Figure 7: Error-cost plots for CIFAR-10H data with $N = 3, 10, 50$. Fixed baselines Fix-InfExp and Fix-FinExp are also included and show strong but inconsistent results. By contrast, InfExp and FinExp maintain consistent results across a variety of budget and hyperparameter settings.

Transitioning to additional findings for ImageNet-16H, we include the Fix-InfExp baseline and observe its high but variable performance. Though superior in many cases, we witness the results of both Fix-InfExp and Fix-FinExp varying significantly across querying budgets as well as experimental settings. The process of learning a prior with either FinExp or InfExp demonstrably stabilizes empirical performance across hyperparameter settings. In all settings we notice a tendency for FinExp to underperform relative to InfExp and posit these

effects to be a result of approximating the likelihood via sampling, whereas the likelihood of InfExp is available in closed form and does not require sampling.

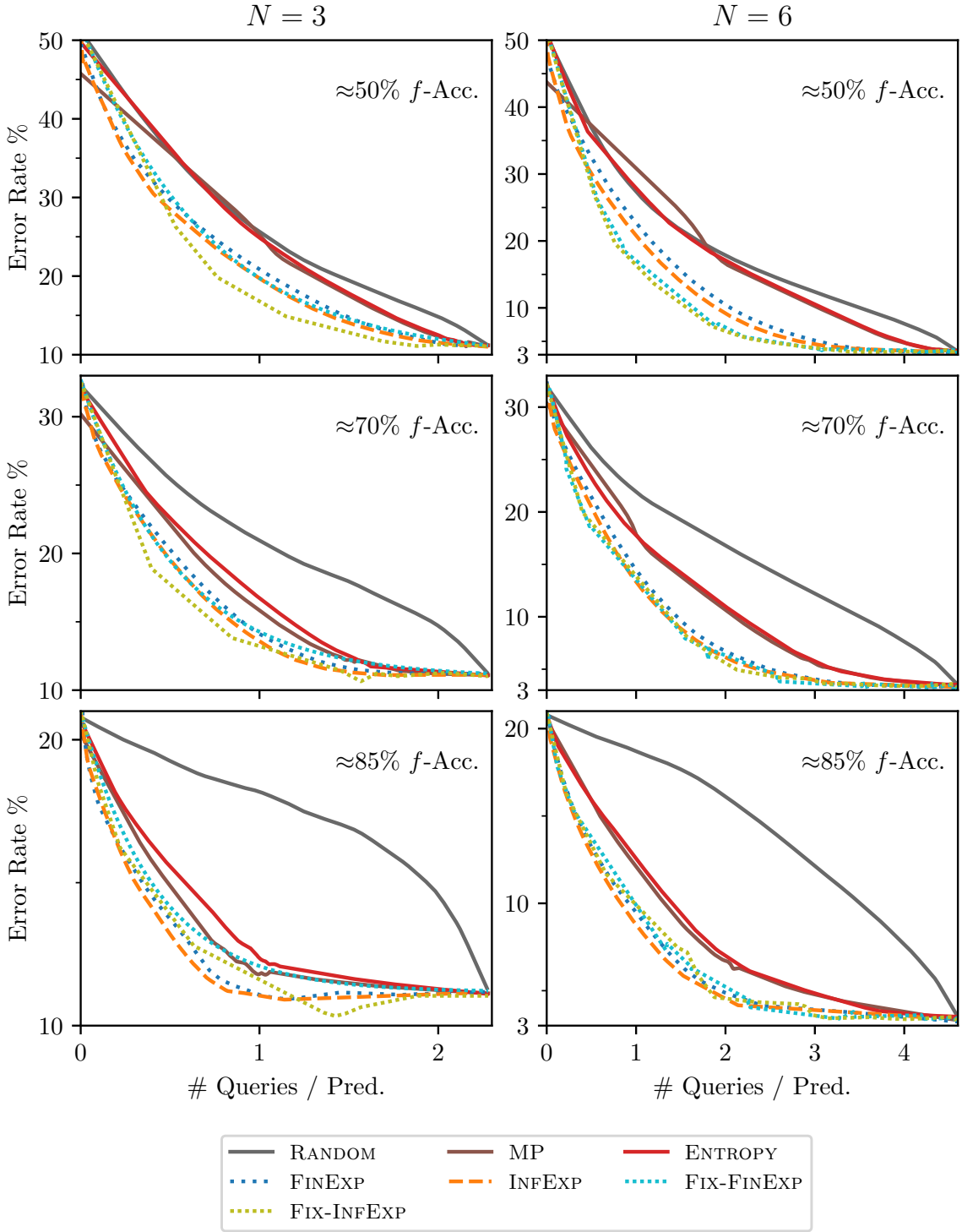

Figure 8: Error-cost plots for ImageNet-16H data with $N = 3, 6$. Fixed baselines Fix-InfExp and Fix-FinExp are also included and depict strong but variable results. By contrast, InfExp and FinExp maintain consistent results across budget and hyperparameter settings.

Results: Distribution Shift (Section 4.4 in paper) To ablate our distribution shift findings in Fig. 9, we report a selection of results for all methods in different budget settings. We see consistent results across these experiments. For the same querying budget, our proposed methods consistently offer improved performance under distribution shift as well as increased sensitivity to the shift as measured by adjusted querying budgets. That is, our methods tend to under-query baselines pre-shift, but over-query post-shift. This behavior demonstrates the promise of these methods to be robust under distribution shifts in real-world situations. At the highest budgets (i.e. 3 queries / sample), we naturally see the methods converge in budget and performance; such a high budget on ImageNet-16H with $N = 6$ often yields absolute ground truth. Even so, in these settings we still consistently (slightly) outperform the baselines, as shown in Table 2.

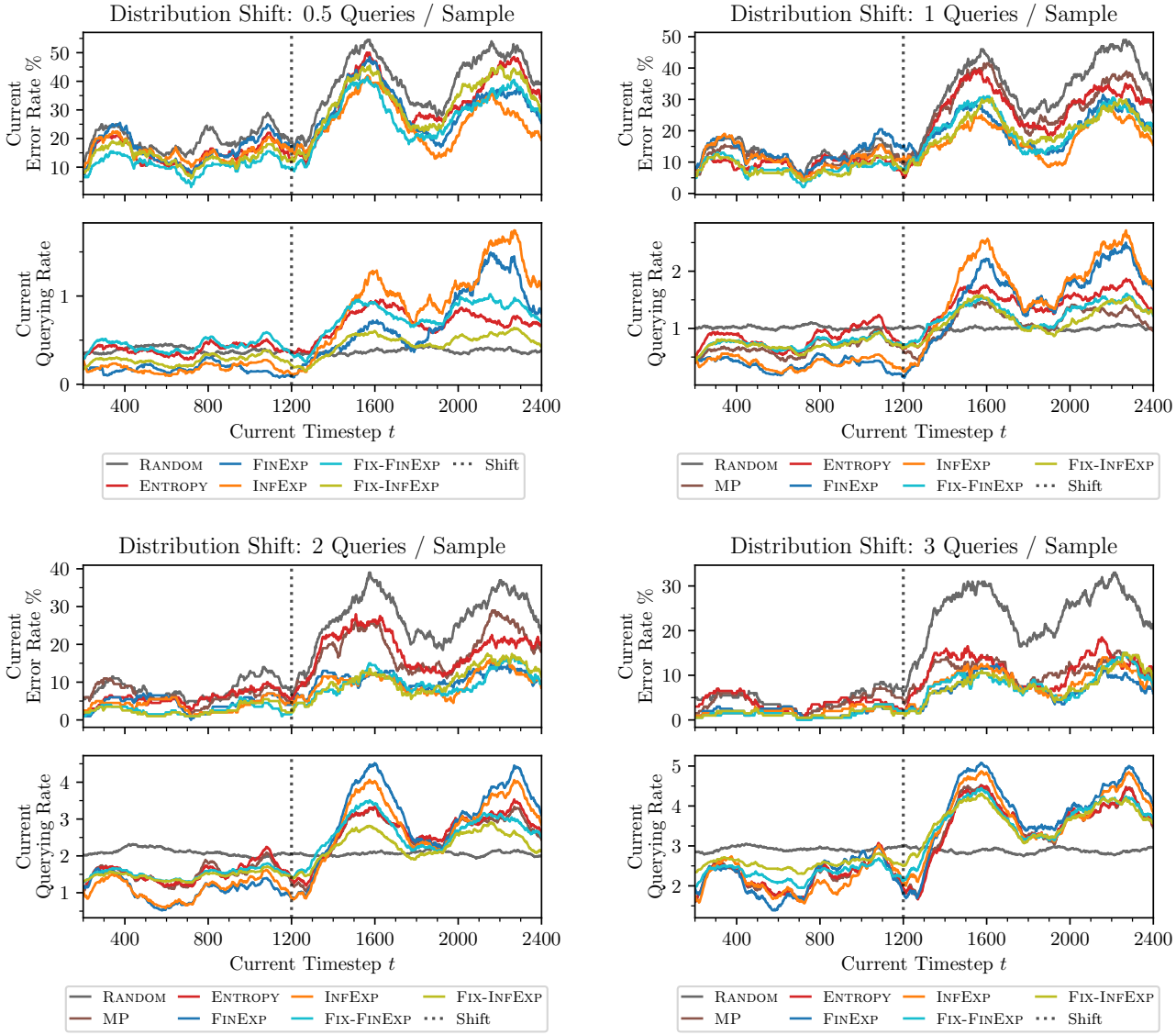

Figure 9: Distribution shift analysis for all methods and baselines across different querying budgets on ImageNet-16H, with performance converging at high budgets when consensus is reached on most samples.

To validate our findings remain consistent across different model architectures, we select a budget of 2 queries / sample and evaluate the performance of methods across architectures, visualized below in Figure 10. Findings remain consistent with Figure 9 and the results in the main paper. However, we note that fixed prior methods Fix-FinExp and Fix-InfExp do not demonstrate the same budget sensitivity to distribution shifts and possess similar querying budget behavior to the baselines.

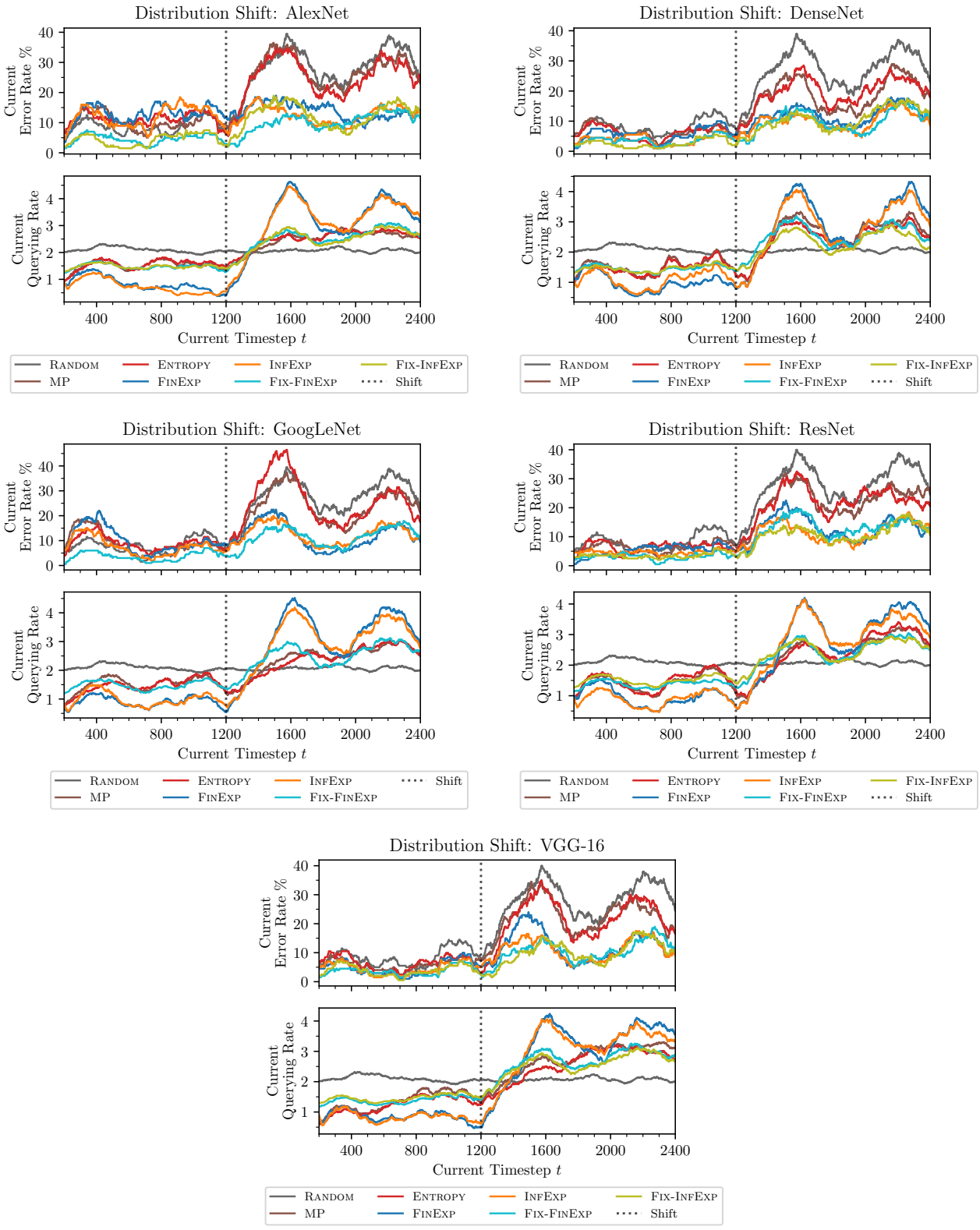

Figure 10: Distribution shift architecture ablation where runs with an average querying budget of 2 queries / sample are compared across five architectures on ImageNet-16H.

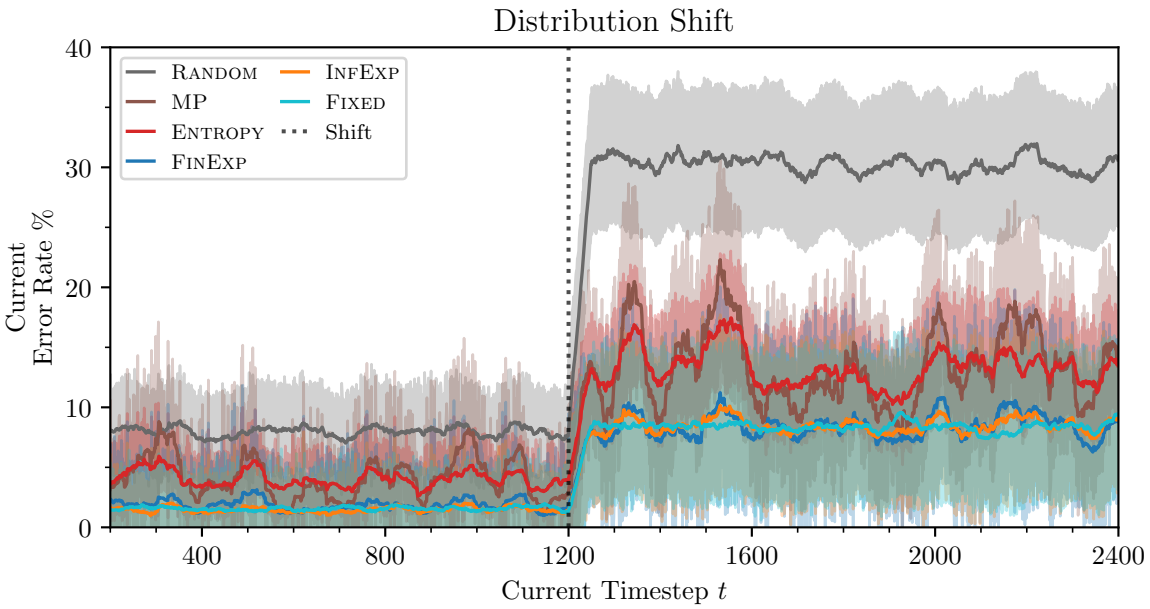

Figure 11: Time-averaged error-rate over 100 sequences of proposed methods and baselines for distribution shift plots of ImageNet-16H. All methods are set to consume a querying budget of roughly 2 queries / sample. Hue represents roughly a 95% confidence interval as measured by the standard error.

Across all runs, we note in Table 2 a significant and consistent improvement in error rate for all budgets with our proposed methods. These findings hold even in cases where the pre-shift error rates of all methods are comparable, such as with a querying budget of 0.5 queries / sample.

Table 2: Error rate (%) of all methods before and after distribution shift on ImageNet-16H, averaged over model architectures and grouped by approximate querying budget per sample (0.5, 1, 2, 3).

Queries / Sample	0.5		1		2		3	
Method	Pre-shift	Post-shift	Pre-shift	Post-shift	Pre-shift	Post-shift	Pre-shift	Post-shift
RANDOM	20.23±1.26	47.55±2.34	13.21±0.29	40.03±0.48	7.9±0.11	29.28±0.24	4.95±0.05	25.5±0.03
ENTROPY	15.02±0.52	38.85±0.69	9.38±0.41	28.58±0.56	4.02±0.12	13.66±0.27	2.52±0.08	9.10±0.09
MP	**11.26±0.35**	31.65±0.9	9.06±0.33	28.0±0.64	3.85±0.12	13.13±0.26	2.44±0.06	9.03±0.07
Fixed	11.56±0.88	32.27±0.97	5.63±0.55	19.66±1.16	1.63±0.06	**8.25±0.16**	1.25±0.07	7.83±0.14
FinExp	13.07±1.15	21.14±1.63	5.84±0.31	11.98±0.19	2.18±0.11	**8.26±0.13**	1.31±0.05	**7.24±0.13**
InfExp	11.55±0.23	**19.65±0.25**	**5.47±0.19**	**11.52±0.17**	**1.54±0.09**	**8.26±0.20**	**1.19±0.07**	7.68±0.16

Results: Time-averaged Distribution Shift Across several of our distribution shift ablation experiments we noted several fluctuations in the querying and error rate, particularly post-shift. We investigated this phenomena and discovered that this was a product of the specific sequence orderings we were averaging over and the moving average smoother accentuating these fluctuations.

Therefore, we conducted an additional analysis by running distribution shift experiments over 100 different random sequence orderings and averaging the error rate performance. This offers us two insights. First, it assists in validating the performance of our proposed methods over time is superior two the baselines. In addition, we can investigate a time-averaged plot to attempt to observe any distribution shift convergence behavior. Below in Fig. 11, we note that our proposed methods consistently outperform the baselines and possess smaller fluctuations in error rate. We attribute the larger error hue not to the performance of our methods but rather to the variance in difficulty in classifying specific samples. For this reason, and likely due to the relatively small number of samples we evaluate on, we are not able to identify any convergence behavior, where the model adjusts to the distribution shift and reaches a "steady-state" error rate. This aligns with our intuition on two-phase evaluation results in Table 3 which demonstrate that our learned prior parameters can only recalibrate the existing model f enough to yield an error rate reduction of a few percentage points.

Results: Model Calibration Analysis (Section 4.5 in paper) To further analyze the ability for our framework to learn model performance and adjust predictions accordingly, we extend the visual findings of Fig. 6 in the main paper by aggregating many runs and grouping them across budgets. In Table 3 below, we present the average Pearson correlations between the learned parameters τ and the per-class model performance relative to consensus. Consistently, we witness strong positive correlations in these values across all budgets and model performance levels. Increased querying budget generally tends to also increase this correlation, though this finding does not always persist across settings.

Table 3: Pearson correlation between learned MAP parameters τ and per class performance of f.

CIFAR-10H		Avg. Queries per Sample			
f-Acc.	Method	0.5	1	2	3
50%	Fix-InfExp	0.550±0.024	-	0.614±0.004	0.712±0.039
	Fix-FinExp	0.625±0.060	-	0.477±0.002	0.471±0.008
	InfExp	0.576±0.007	0.638±0.012	0.649±0.009	0.562±0.034
	FinExp	0.622±0.014	0.623±0.020	0.560±0.012	0.590±0.038
70%	Fix-InfExp	-	0.600±0.016	0.662±0.002	0.670±0.020
	Fix-FinExp	-	0.573±0.013	0.577±0.002	0.506±0.005
	InfExp	0.576±0.01	0.640±0.012	0.712±0.006	0.705±0.011
	FinExp	0.600±0.008	0.601±0.015	0.662±0.013	0.640±0.019
90%	Fix-InfExp	-	0.669±0.021	0.702±0.002	-
	Fix-FinExp	-	0.687±0.014	0.590±0.002	-
	InfExp	0.521±0.011	0.587±0.006	0.712±0.013	0.767±0.024
	FinExp	0.589±0.005	0.681±0.018	0.659±0.041	0.672±0.022