DUET: OPTIMIZING TRAINING DATA MIXTURES VIA FEEDBACK FROM UNSEEN EVALUATION TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The performance of a machine learning (ML) model depends heavily on the relevance of its training data to the domain of the downstream evaluation task. However, in practice, the data involved in an unseen evaluation task is often not known to us (e.g., conversations between an LLM and a user are end-to-end encrypted). So, it is not obvious what data would be relevant for training/fine-tuning the ML model to maximize its task performance. Instead, one can only deploy the ML model in the unseen evaluation task to gather multiple rounds of coarse feedback on how well the model has performed. This paper presents a novel global-to-local algorithm called DUET that can exploit the feedback loop by interleaving a *data selection* method with *Bayesian optimization*. As a result, DUET can efficiently refine the training data mixture from a pool of data domains to maximize the model's performance on the unseen evaluation task and its convergence to the optimal data mixture can be theoretically guaranteed by analyzing its *cumulative regret*. Empirical evaluation on image and LLM evaluation tasks shows that DUET finds better training data mixtures than conventional baselines.

1 INTRODUCTION

The performance of an ML model depends heavily on the composition of training data domains Chen et al. (2024a); Xie et al. (2023) and the downstream evaluation task Hoffmann et al. (2022); Long et al. (2017). For instance, if users of an LLM are interested in asking layman science questions, then fine-tuning the LLM with more Wikipedia data allows it to converse better with the users. Hence, knowing the evaluation task is important as it informs us on the relevant training data to be selected from an existing pool of data domains to produce a better-performing ML model.

However, in practice, the data (e.g., its domain, distribution, or labels) involved in a downstream *unseen evaluation task* is often not known to us. So, it is not obvious what data would be relevant for training or fine-tuning the ML model. Instead, one can only deploy the ML model a few times in the unseen evaluation task to gather multiple rounds of feedback on how well our ML model has performed, thereby creating a feedback loop. Furthermore, each round of feedback incurs significant time or monetary costs. Hence, the key challenge lies in how to achieve efficiency in the number of feedback rounds to refine the training data and improve the task performance of the ML model. This problem setting has become increasingly important recently: Any LLM owner would be interested in fine-tuning its LLM to converse better with the users but due to privacy concerns Li et al. (2024), conversations between their deployed LLM and users are end-to-end encrypted (`openai.com/enterprise-privacy`). So, the LLM owner does not know the conversation domain or data seen by the deployed LLM. Rather, the LLM owner can only receive coarse feedback on how well its LLM has performed in the conversation (e.g., ratings from human users indicating their satisfaction with the LLM) and gather multiple rounds of feedback from the users.

This paper presents a novel algorithm called **DUET** (Fig. 1) that can exploit the feedback loop to optimize the training **D**ata mixture for the **U**nseen **E**valuation **T**ask. DUET is a *global-to-local* algorithm that interleaves *influence function* (IF) Koh & Liang (2017) as a *data selection* method Albalak et al. (2024); Ting & Brochu (2017) with *Bayesian optimization* (BO) Snoek et al. (2012); Srinivas et al. (2010) to optimize the training data mixture for the unseen evaluation task. At the global level, BO in DUET uses feedback from the unseen evaluation task to automatically reconfigure the mixing ratio of data domains in the training data mixture iteratively. At the local level, DUET uses IF to retrieve high-quality data points from each data domain until the proposed mixing ratio is reached, improving the quality of data mixture every iteration. By doing so, DUET efficiently refines the training data mixture and improve the ML model's performance without needing to know the data involved in the unseen evaluation task.

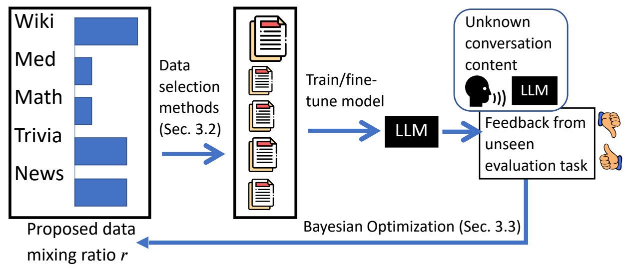


Figure 1: Overview of DUET algorithm that exploits a feedback loop to optimize the data mixture for the unseen evaluation task.

In our problem setting, (a) there is no direct access to the data (e.g., its domain, distribution, or labels) involved in the unseen evaluation task but (b) multiple rounds of feedback (details covered in Sec. 2.2) can be gathered from the task using a trained ML model. App. A.1 provides other practical examples of such a setting. This setting is distinctively different from those considered in conventional domain adaptation (DA) and domain generalization (DG) works. In particular, prior DA work assumes knowledge of fine-grained data (e.g., a pool of labeled/unlabeled data Zhang et al. (2022) or data distribution Ganin & Lempitsky (2015); Zhang et al. (2021)) from the evaluation task for selecting relevant training data that match the evaluation data. On the other hand, DG considers a rigid setting with no knowledge (not even feedback) of the evaluation task Muandet et al. (2013); Shin et al. (2024); Wang et al. (2022). Recently, works such as DoReMi Xie et al. (2023) have also used distributionally robust optimization (DRO) Chen et al. (2024a); Fan et al. (2024) to reweigh training data domains so that a trained LLM performs well for *any* distribution of downstream language tasks. However, they do not exploit feedback from the actual downstream evaluation task to improve the training data mixture. Hence, they do not work well in our setting, as shown in Sec. 5. Lastly, some works Ruder & Plank (2017) have used feedback to select training data for transfer learning but rely heavily on hand-crafted data features and still require knowledge of the data from the downstream evaluation task.

Other straightforward approaches do not work well in our problem setting. A naive approach is to train an ML model on the union of data taken from every data domain. However, our work here (Sec. 5.2) and some others Xia et al. (2024) show that the trained ML model does not perform as well as a model trained using strategically selected data relevant to the evaluation task. Another brute-force approach is to iterate through all possible data mixtures (of different mixing ratios) and select one that yields the best evaluation task performance, which is not feasible due to the need to evaluate an excessive number of ML models. Lastly, App. A.2 discusses related work on data selection which, in isolation, do not exploit feedback from the evaluation task.

To the best of our knowledge, DUET is the first work to exploit coarse feedback from an unseen evaluation task and interleaves data selection with BO to reweigh the data domains *adaptively*. After several iterations, DUET automatically assigns a higher proportion of data mixture to more relevant training data domains, consequently producing a better data mixture. The specific contributions of our work here are as follows:

- We introduce a novel and realistic problem setting where the data involved in an unseen evaluation task is not known to us but our ML model can be deployed to gather multiple rounds of feedback from the task. Then, we introduce a novel algorithm called **DUET** that can exploit the feedback loop to optimize the training **D**ata mixture for the **U**nseen **E**valuation **T**ask. To achieve this, DUET interleaves influence function as a data selection method (Sec. 3.2) with Bayesian optimization (Sec. 3.3).
- We provide a theoretical analysis of DUET's convergence to the optimal unseen evaluation task performance by analyzing DUET's *attained cumulative regret* Chen et al. (2024b); Chowdhury & Gopalan (2017) under the BO framework (Sec. 4).

- We demonstrate the effectiveness of DUET on a variety of image classification and LLM language tasks comprising both in-domain and out-of-domain unseen evaluation tasks. Compared to conventional approaches (e.g., DoReMi or uniform weights), DUET finds better data mixtures for training ML models that perform better on the downstream unseen evaluation task (Sec. 5.2).

2 Preliminaries

2.1 Bayesian optimization

We first provide an outline of how BO can be used to optimize a generic black-box objective function. We will provide details later on how BO is used in DUET (Sec. 3.3). We consider a black-box objective function $f : \mathbb{R}^n \mapsto \mathbb{R}$ over the space of inputs $r \in \mathbb{R}^n$. The goal is to find $r^* \triangleq \arg\min_r f(r)$ which minimizes the objective function. BO is an *active algorithm* that strategically selects input points to query the black-box objective function, conditioned on previous function observations. At each iteration $t = 1, 2, \ldots, T$ of BO, we query the black-box function with a selected input r_t to obtain a *noisy* observation $\tilde{y}_t \triangleq f(r_t) + \epsilon_t$ with a sub-Gaussian noise ϵ_t (e.g., Gaussian or bounded noise) to form the sample $(r_t, \tilde{y}_t)$. Consistent with the work of Chowdhury & Gopalan (2017), we model the unknown function f as a realization of a *Gaussian process* (GP) Williams & Rasmussen (2006) that is fully specified by its *prior* mean $\mu(r)$ and covariance $\kappa(r, r')$ for all $r, r' \in \mathbb{R}^n$ where κ is a *kernel* function chosen to characterize the correlation of the observations between any two inputs r and r'; a common choice is the *squared exponential* (SE) kernel $\kappa(r, r') \triangleq \exp(-\|r - r'\|_2^2/(2m^2))$ with a *length-scale* hyperparameter m that can be learned via maximum likelihood estimation from observations. Given a column vector $\boldsymbol{y}_t \triangleq [\tilde{y}_\tau]^\top_{\tau=1,\ldots,t}$ of noisy observations at previous inputs $r_1, \ldots, r_t$, the posterior belief of f at any new input r' is a Gaussian distribution with the following *posterior* mean and variance:

$$
\begin{aligned}
\mu_t(r') &\triangleq \kappa_t^\top(r')(K_t + \zeta I)^{-1}\boldsymbol{y}_t \\
\sigma_t(r') &\triangleq \kappa(r', r') - \kappa_t^\top(r')(K_t + \zeta I)^{-1}\kappa_t(r')
\end{aligned} \tag{1}
$$

where $\kappa_t(r') \triangleq [\kappa(r', r_\tau)]^\top_{\tau=1,\ldots,t}$ is a column vector, $K_t \triangleq [\kappa(r_\tau, r_{\tau'})]_{\tau,\tau' \in 1,\ldots,t}$ is a $t \times t$ covariance matrix, and $\zeta > 0$ is viewed as a free hyperparameter that depends on the problem setting Chowdhury & Gopalan (2017). Using equation 1, the BO algorithm selects the next input query r_{t+1} by optimizing an *acquisition function*, such as minimizing the *lower confidence bound* (LCB) acquisition function Srinivas et al. (2010): $r_{t+1} = \arg\min_r \mu_t(r) - \beta_{t+1}\sigma_t(r)$ with an exploration parameter β_{t+1}. In addition, BO can also handle constraints on inputs r Gardner et al. (2014). The cumulative regret (for T BO iterations w.r.t. a minimization problem) $R_T \triangleq \sum_{t=1}^{T}[f(r_t) - f(r^*)]$ is used to assess the performance of a BO algorithm Chowdhury & Gopalan (2017); Tay et al. (2023) where $f(r^*)$ is the true function minimum. A lower cumulative regret indicates a faster convergence rate of the BO algorithm. We provide a theoretical analysis of DUET's cumulative regret in Sec. 4.

2.2 Problem setting: Optimizing data mixtures

In this subsection, we formally describe our problem setting. Suppose that we have N training datasets $\mathcal{D} \triangleq \{D_1, D_2, \ldots, D_N\}$ from N different domains (e.g., Wikipedia, ArXiv for language tasks). Hence, $\mathcal{D}$ is the union of training datasets from each domain. Let $\mathcal{L}_{\text{eval}}(\theta)$ be the unseen evaluation task loss w.r.t. an ML model parameterized by θ. This loss can only be observed as a coarse feedback from the unseen evaluation task and does not have a closed, mathematical form. Our goal is to find an optimal data mixture $\mathcal{X}^* \in \mathcal{D}$ (a set of training data points) and learn model parameters $\theta_{\mathcal{X}^*}$ such that the unseen evaluation task loss $\mathcal{L}_{\text{eval}}$ is minimized:

$$
\begin{aligned}
\min_{\mathcal{X} \in \mathcal{D}} \quad & \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}}) \\
\text{s.t.} \quad & |\mathcal{X}| = M,
\end{aligned} \tag{2}
$$

where $\theta_{\mathcal{X}} \triangleq \arg\min_\theta \mathcal{L}_{\text{train}}(\mathcal{X}, \theta)$ is the model parameters learnt in a standard supervised learning manner (e.g., gradient descent) from a chosen data mixture $\mathcal{X}$ and $\mathcal{L}_{\text{train}}$ is a standard model training loss (e.g., cross-entropy loss for LLM prediction). M is a practical constraint that can be decided beforehand Mirzasoleiman et al. (2020) and is used to ensure the selected data mixture is not too large. In practice, evaluation task loss $\mathcal{L}_{\text{eval}}$ can also be interchanged with other measures to be maximized (e.g., accuracy, user ratings).

3 OPTIMIZING TRAINING DATA MIXTURES USING DUET

Unfortunately, solving problem (2) is challenging because the unseen evaluation task loss $\mathcal{L}_{\text{eval}}$ does not have a closed, mathematical form and finding the optimal data mixture $\mathcal{X}^*$ directly is a high-dimensional discrete optimization problem if the size of each dataset in $\mathcal{D}$ large. To alleviate this, DUET adopts a global-to-local approach to optimize the training data mixture. At a global level, DUET exploits feedback $\mathcal{L}_{\text{eval}}$ from the unseen evaluation task to iteratively refine the mixing ratio of training data domains in $\mathcal{D}$. At a local level, DUET uses IF as a data selection method to remove low-quality data points from the data mixture at each iteration.

3.1 REPARAMETERIZATION OF THE OPTIMIZATION PROBLEM

To perform DUET effectively, we first reparameterize the objective function of problem (2) into a bilevel optimization problem that, at the outer level, depends on the mixing ratio $r \in \mathbb{R}^N$ of training data domains (entries in r sum to 1). This reparameterized problem has a unique structure that can be solved by interleaving data selection methods with BO, which we cover in Sec. 3.2 & 3.3.

Theorem 3.1. *$\mathcal{X}^*$, the optimal set of data points from $\mathcal{D}$, is the solution of the original problem (2) iff $r^* = ratio(\mathcal{X}^*)$ is the optimal mixing ratio solution of the reparameterized problem:*

$$\min_{r\in\mathbb{R}^N} \min_{\mathcal{X}\in S_r} \quad \mathcal{L}_{eval}(\theta_{\mathcal{X}}), \tag{3}$$

where $S_r \triangleq \{\mathcal{X} : \mathcal{X} \in \mathcal{D}, ratio(\mathcal{X}) = r, |\mathcal{X}| = M\}$ and $ratio(\mathcal{X}) = r$ means that the data points in $\mathcal{X}$ satisfies the given ratio $r \in \mathbb{R}^N$ from N data domains and $\|r\|_2 = 1$.

The proof can be found in App. B.1, where we show that $\mathcal{X}^*$, the solution data mixture of original problem (2), satisfies a mixing ratio r^* that is also the solution of the reparameterized problem (3). Notice that this reparameterized problem consists of an outer and inner optimization problem, and the outer problem requires us to find the optimal mixing ratio r^*. DUET aims to solve problem (3) in an iterative manner. At the outer optimization level (global), DUET uses BO to exploit feedback from the evaluation task to propose a promising mixing ratio r_t at each iteration t. At the inner optimization level (local), we introduce a sampling strategy that uses the IF values of each data point w.r.t. its local domain to retrieve a high-quality subset of data points that satisfies the proposed mixing ratio r_t and approximately solves the inner problem. By repeating the process iteratively, our approach theoretically converges (theorem. 4.1) to the optimal data mixture and outperforms other baselines in our empirical experiments (Sec. 5.2).

To illustrate DUET qualitatively, consider an unseen evaluation task consisting of an LLM being deployed to converse with users that frequently ask layman scientific questions. At first, an LLM fine-tuned on data from different domains with uniform ratio cannot perform optimally on the evaluation task, since most of the fine-tuning data are irrelevant. In the outer optimization problem, BO in DUET uses the feedback from the task to automatically place more weight w.r.t. mixing ratio r on the Wikipedia domain (better for layman scientific questions). In the inner optimization problem, DUET uses IF to remove low-quality data points (e.g., stub articles) from Wikipedia data Shen et al. (2017) and allows us to estimate the solution of the inner problem more accurately (Sec. 3.2). In the next few sections, we provide details and theoretical insights involved in solving both the inner (using IF as a data selection method in Sec. 3.2) and outer problem (using BO in Sec. 3.3).

3.2 USING DATA SELECTION METHODS FOR INNER PROBLEM

The inner optimization problem seeks to find the best-performing data mixture that satisfies the given mixing ratio r from the outer level. In this section, we propose an IF-driven estimator that relies on sampling to approximately solve the inner problem given a data ratio r:

$$\mathcal{X}_r^* \triangleq \arg\min_{\mathcal{X}\in S_r} \quad \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}}), \tag{4}$$

where $S_r \triangleq \{\mathcal{X} : \text{ratio}(\mathcal{X}) = r, |\mathcal{X}| = M\}$. To solve the inner problem, we need to find a subset of data $\mathcal{X}_r^*$ that yields the lowest evaluation task loss $y_r^* = \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_r^*})$ while still constrained to the proposed mixing ratio r. A simple approach, based on prior works on estimating distribution extrema de Haan (1981); Lee & Miller (2022), is to *randomly* sample k different data mixtures from S_r. This yields k samples of training data mixtures $\{\mathcal{X}_1, \ldots, \mathcal{X}_k\}$ (each satisfying the mixing ratio r), in which a **uniform random estimator** for y_r^* can be obtained by checking the evaluation task loss of the ML model trained on each data mixture sample and taking the minimum: $\widetilde{y_r^*} = \min_{\mathcal{X}_i}\{\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_1}), \ldots, \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_k})\}$ and $\widetilde{\mathcal{X}}_r^* = \arg\min_{\mathcal{X}_i}\{\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_1}), \ldots, \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_k})\}$ as the solution estimate of inner problem (4). The estimator $\tilde{y}_r^*$ is the 1st-order statistic Arnold et al. (2008) and a random variable. While consistent (i.e., as we increase the sampling size k, we can estimate the solution of Eq. 4 more accurately), the uniform random estimator $\tilde{y}_r^*$ has high variance (we provide empirical evidence in Fig. 6) because from k uniformly random data mixture samples, it is unlikely we can select the optimal data mixture.

We aim to improve the quality of estimator $\widetilde{y_r^*}$ by using data selection methods Sim et al. (2022); Wang et al. (2024a) in our sampling process to improve the chance of selecting a data mixture that results in a smaller evaluation task loss. Specifically, we want to reduce the estimator's variance or bias (w.r.t. a fixed sampling size k) by increasing the chance of sampling high-quality data points (conversely, reduce the chance of sampling low-quality data points) from each data domain, before using it to train an ML model. To do so, we incorporate Influence function Koh & Liang (2017) (IF), a popular data selection method that identifies high-quality data points Saunshi et al. (2023) into our estimator $\widetilde{y_r^*}$, and show empirically that doing so improves our estimation of the inner problem solution by reducing our estimator's bias and variance. In App. A.4, we also explore and discuss the use of other data selection methods, such as coresets Mirzasoleiman et al. (2020) and diversity-driven subset selection Wang et al. (2024b). In general, we found the use of IF the most practical due to its ease of implementation and efficiency.

IF-driven estimator. We construct the IF-driven estimator in the following manner: first, for each dataset $D_i \in \mathcal{D}$ from the training domains, we train or fine-tune a local model on that dataset (e.g., train a model from Wikipedia data, a model from ArXiv etc.). This produces N different ML models. Second, we derive the IF value of every training data point w.r.t. the trained ML model for its respective domain (this can be computed and stored beforehand; more details in App. A.3). Lastly, given a mixing ratio r proposed at each iteration, we perform weighted sampling from each domain based on each data point's IF value within the domain dataset (instead of uniform sampling as mentioned previously) until we satisfy the mixing ratio r. From hereon, we refer to this sampling process as *IF-weighted sampling*. Hence, for each data domain, there is a higher chance to sample a data point with a higher IF value. This yields a single sample of data mixture $\mathcal{X}^{IF}$. By performing IF-weighted sampling k times, we obtain k samples of IF-weighted data mixtures $\{\mathcal{X}_1^{IF}, \ldots, \mathcal{X}_k^{IF}\}$, in which we obtain a new **IF-driven estimator**:

$$\widetilde{y_r^*} = \min_{\mathcal{X}_i}\{\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_1^{IF}}), \ldots, \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_k^{IF}})\}, \tag{5}$$

which we use to estimate the solution of inner optimization problem (4). The key difference between the IF-driven estimator and the uniform random estimator is that the IF-driven estimator places higher emphasis on selecting data with high IF values, and prior works Saunshi et al. (2023) have regarded data points with higher IF values as of higher quality. Next, we provide empirical evidence into why the IF-driven estimator performs better than the uniform random estimator in finding better data mixtures.

In Fig. 2, we have a simple setting of mixing data from two domains to train an ML model to maximize an evaluation task accuracy (while Eq. 4 & 5 consider the minimization case, we can use $\max$ instead of $\min$ for the maximization case). Here, we use a fixed mixing ratio r of 1:1. The optimal data mixture satisfying this ratio attains the evaluation task accuracy indicated by the red line and is also the solution of the inner optimization problem (in this example, we obtain this by iterating through all possible data mixtures in a brute-force manner). Ideally, we want our estimator to be as close to the red line as possible. Next, we plot the empirical distribution of the **uniform random estimator** and **IF-driven estimator**. Empirically, the IF-driven estimator (green histogram) has a lower variance and bias than the uniform random estimator (gray histogram), producing a closer estimate to the true solution (red line). Therefore, the IF-driven estimator $\widetilde{y_r^*}$ estimates the solution of Eq. 4 more accurately with lower bias and variance.

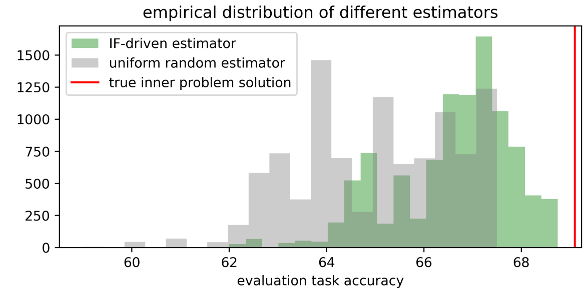


Figure 2: Empirical distribution of the uniform random and IF-driven estimator $\widetilde{y_r^*}$. Red line is the true inner problem solution that we are estimating.

Next, we would like to characterize how close the evaluation task loss of data mixture obtained from our IF-driven estimator $\widetilde{y_r^*}$ is to the optimal evaluation task loss y_r^* w.r.t. a given data ratio r. To do so, we theoretically analyze the estimator's empirical distribution. From our experiments, the sampling distribution of the evaluation task loss of each data mixture sample $\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}^{IF}})$ is similar to a truncated exponential distribution (we provide more evidence in App. A.5). Based on this, the following theorem characterizes how well the IF-driven estimator $\widetilde{y_r^*}$ estimates y_r^*.

Theorem 3.2. *Let $\{\mathcal{X}_1^{IF}, \ldots, \mathcal{X}_k^{IF}\}$ be k samples of data mixtures drawn from S_r using IF-weighted sampling. Furthermore, assume each independent sample $\mathcal{L}_{eval}(\theta_{\mathcal{X}_i^{IF}})$ follows the shifted truncated exponential distribution $y_r^* + \exp_t(\lambda, c)$, for $i = 1, 2, \ldots, k$ where $\exp_t(\lambda, c)$ is a truncated exponential distribution governed by rate parameter λ and truncated at $c > 0$. Then, the IF-driven estimator $\widetilde{y_r^*}$ defined in Eq. 5 is a random variable: $y_r^* + \epsilon$, where y_r^* is the true inner problem solution of Eq. 4 and ϵ is a random noise variable with probability density function:*

$$pdf_\epsilon(u) = \frac{\lambda k e^{-\lambda u}}{1 - e^{-\lambda c}} \left(\frac{e^{-\lambda u} - e^{-\lambda c}}{1 - e^{-\lambda c}} \right)^{k-1} \quad on\ u \in [0, c]\ .$$

The proof is shown in App. B.2 and computes the probability distribution of the 1st order statistic (in which our estimator uses) of a truncated exponential distribution. In App. B.4, we also provide details to help readers extend our analysis to other empirical sampling distributions as long as they are sub-Gaussian Chowdhury & Gopalan (2017). This theorem indicates that the support of our IF-driven estimator's distribution is on $[y_r^*, y_r^* + c]$ and so this estimator is positively biased. Furthermore, the pdf indicates that the IF-driven estimator is *consistent*, since the estimation error ϵ reduces asymptotically to 0 as the sampling size k increases. Surprisingly, our experiments (Sec. 5) show that using $k = 1$ is enough to select good data mixtures, underscoring the effectiveness of the IF-driven estimator in finding high-quality data mixtures. Theorem 3.2 will be used in our theoretical analysis of DUET's convergence in Sec. 4.

3.3 Using Bayesian optimization for outer problem

With the IF-driven estimator introduced to estimate the inner optimization problem solution, we shift our focus to solving the outer optimization problem of problem (3), which aims to find the optimal data mixing ratio r^* for the unseen evaluation task. Since the solution of the inner problem $y_r^* = \min_{\mathcal{X} \in S_r} \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}})$ depends only on the mixing ratio r, we can succinctly define a function $f(r) \triangleq y_r^* = \min_{\mathcal{X} \in S_r} \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}})$, where for a given mixing ratio r, we use the IF-driven estimator to estimate a solution for the inner problem, producing $f(r)$. As such, the outer optimization problem of problem (3) can be rewritten into:

$$\min_r f(r). \tag{6}$$

where $r \in \mathbb{R}^N$ is the mixing ratio over the N training domains and the sum of entries in r is constrained to 1 (since it is a ratio). DUET uses BO with constraints of $\|r\|_2 = 1$ (Sec. 2.1) to find the optimal data mixture ratio r^* to solve outer problem (6). BO is suitable for solving this problem for a few reasons. First, evaluating f requires us to use the IF-driven estimator to estimate the inner optimization problem solution and thus f is a black-box function with no closed, mathematical form; BO is a principled and popular framework to optimize such black-box functions Garnett (2023); Pyzer-Knapp (2018). Second, we can only estimate the inner problem solution (Theorem 3.2) using our IF-driven estimator introduced in the previous section. Hence, this implies we can only obtain *noisy observations* $f(r) + \epsilon$, where ϵ is a random noise variable with the same distribution as that in theorem 3.2; fortunately, BO handles noisy function observations gracefully Srinivas et al. (2010); Chowdhury & Gopalan (2017) during the optimization process, allowing us to find the optimal mixing ratio eventually (theoretical results shown in Sec. 4).

3.4 Interleaving the IF-driven estimator and BO

DUET uses BO at the outer level and IF-driven estimator at the inner level to iteratively optimize the data mixture, solving problem (3). We describe DUET in Algorithm. 1.

Algorithm 1 DUET: Optimizing **D**ata Mixtures for **U**nseen **E**valuation **T**ask

1: **Input:** N training datasets from N domains $\{D_1, \dots, D_N\}$. Computed IF values of each data point (App. A.3) w.r.t. its domain dataset and locally trained model. Initial observation of data mixture ratio and evaluation task performance: $\mathcal{D}_0 \triangleq \{(r_0, \tilde{y}_0)\}$, SE kernel κ, sampling size k, parameter β_t for acquisition step and total number of BO iterations T.
2: **for** $t = 1, \dots, T$ **do**
3: $\quad r_t = \arg\min_r \mu_t(r) - \beta_t \sigma_t(r)$ (BO acquisition step)
4: $\quad$ IF-weighted sampling to obtain k samples of data mixtures $\{\mathcal{X}_1^{IF}, \dots, \mathcal{X}_k^{IF}\}$ (Sec. 3.2).
5: $\quad$ **IF-driven estimator** at iteration t:
$\quad \widetilde{y_t^*} = \min_{\mathcal{X}_i}\{\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_1^{IF}}), \dots, \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_k^{IF}})\}$.
6: $\quad$ Keep track of best performing data mixture $\mathcal{X}_t^* = \arg\min_{\mathcal{X}_i}\{\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_1^{IF}}), \dots, \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_k^{IF}})\}$.
7: $\quad \mathcal{D}_t = \mathcal{D}_{t-1} \cup \left\{ \left(r_t, \widetilde{y_t^*} \right) \right\}$
8: $\quad$ Update the GP posterior and κ with updated observations $\mathcal{D}_{t+1}$ (Sec. 2.1).
9: **end for**
10: $\mathcal{X}^* = \arg\min_{\mathcal{X}_i^* \in \{\mathcal{X}_1^*, \dots, \mathcal{X}_T^*\}} \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_i^*})$

At iteration t, DUET uses the LCB acquisition function Srinivas et al. (2010) on the GP posterior to propose a candidate mixing ratio r_t for our data domains (Line 3). Using the proposed mixing ratio r_t, we use IF values of each data point to compute the IF-driven estimator $\widetilde{y_r^*}$ and keep track of the best performing data mixture $\mathcal{X}_t^*$ at current iteration t (Line 4, 5 and 6). Note that the data mixture $\mathcal{X}_t^*$ at each iteration t satisfies the proposed mixing ratio r_t. Next, we include $(r_{t+1}, \widetilde{y_t^*})$ into our historical observations $\mathcal{D}_{t+1}$ (Line 7) and update our GP posterior (Line 8). After which, we repeat the entire process, until the budget of T BO iterations is exhausted. In the end, we recover the best performing data mixture $\mathcal{X}^*$ (Line 10).

DUET can be implemented easily by LLM practitioners. Once a data mixture is sampled using the IF-driven estimator to fine-tune the LLM at each BO iteration, the trained LLM can be deployed for a small period of time (e.g., one day on a small subset of users) to gather feedback (e.g., user rating) from conversations with human users. Then, DUET proposes a new data mixing ratio to refine the training data mixture. As seen from our experiments (Sec. 5), the model performance on the unseen evaluation task improves as DUET progressively optimizes the data mixture to be more relevant to the task.

4 Theoretical Analysis

4.1 Convergence analysis using cumulative regret

We analyze the convergence rate of DUET using the growth of *attained cumulative regret* Chen et al. (2024b) $\tilde{R}_T = \sum_{t=1}^T |\widetilde{y^*_{r_t}} - f(r_t)| = \sum_{t=1}^T |f(r^*) + \epsilon_t - f(r_t)|$ for T BO iterations. The attained cumulative regret consists of two terms, where $|f(r^*) - f(r_{r_t})|$ indicates the quality of mixing ratio r_t proposed at each iteration while ϵ_t indicates how well we can estimate the inner problem solution at every iteration. By analyzing the attained *average* regret $\tilde{R}_T/T$ with $T \to \infty$, the following theorem helps us understand how close our algorithm converges Berkenkamp et al. (2019) to the optimal evaluation task loss with increasing number of BO iterations T.

Theorem 4.1. *Let f be the outer problem objective defined in Eq. 6 with bounded RKHS norm: $||f||_\kappa = \sqrt{\langle f, f\rangle_\kappa}$. Also, let our IF-driven estimator for the inner problem solution be governed by the error distribution introduced in Theorem 3.2 with constant c and $\lambda = 1$. Let $A_{c,k} = \frac{c^2(1-e^{-c}-\frac{c}{2})^{k-1}}{(1-e^{-c})^k}$, where k is a fixed predecided sampling size. Then, running DUET over f using the LCB acquisition function found in (Chowdhury & Gopalan, 2017) at each BO iteration $t = 1, \ldots, T$ yields the following **attained average regret** Chen et al. (2024b) upper bound with probability at least $1 - \delta$:*

$$\lim_{T\to\infty} \frac{\tilde{R}_T}{T} \leq \frac{6(\sqrt[4]{\delta} + \sqrt{k})}{\sqrt[4]{\delta}k} + 2A_{c,k} + \frac{\sqrt{2A_{c,k}}}{\sqrt[4]{\delta}}.$$

The proof is provided in App. B.3 and bounds $|f(r^*) - f(r_{r_t})|$ and ϵ_t independently using BO regret analysis Chen et al. (2024b); Chowdhury & Gopalan (2017) and the error distribution defined in Theorem. 3.2. Our theorem's average regret indicates how close our algorithm converges to the optimal evaluation task loss with increasing BO iteration T and different choices of sampling size k. Notice that because c characterizes the error of our estimator in Theorem. 3.2, a larger c would decrease $A_{c,k}$ and our average regret. In addition, a larger sampling size k reduces the estimation error of the inner problem (Theorem. 3.2), decreasing $A_{c,k}$ and also reduces our regret bound, allowing us to achieve a better-performing data mixture.

In practice, using a large k is computationally expensive because we need to use our IF-driven estimator to sample data mixtures and train our ML models k times at each iteration (selecting one that attains the smallest $\mathcal{L}_{\text{eval}}$). Fortunately, our experiments (Sec. 5.2) show that setting $k = 1$ is sufficient to achieve better results than other baselines. If computational resource is not an issue, we can also consider setting an adaptive sampling size Chen et al. (2024b) that increases w.r.t. each iteration t.

5 Experiments and Discussion

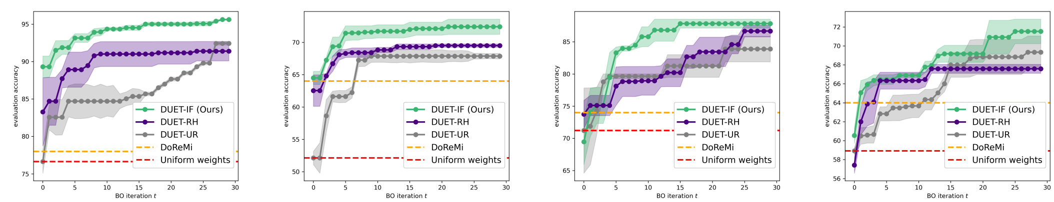
In this section, we conduct extensive experiments to showcase the effectiveness of DUET compared to other baselines. Our experimental evaluation pipeline is constructed as follows: *first*, we select data mixtures from different data domains with DUET or other baselines. *Second*, we train or fine-tune an ML model according to the selected data mixture. *Third*, we deploy the ML model on the unseen evaluation task to evaluate how well the model has performed. In the next subsection, we provide more details next on how our experiment is setup with varying training and evaluation data domains to showcase DUET's effectiveness even in traditionally difficult out-of-domain scenarios. **Our code is in the supplementary material folder**.

5.1 Experimental setup

Our experiments are carried out on two broad classes of evaluation tasks. The first consists of image classification tasks by a VGG-16 model Simonyan & Zisserman (2015) over different object domains Russakovsky et al. (2015); Xiao et al. (2017). The second consists of LLM evaluation tasks by a Llama-8b-Instruct model Touvron et al. (2023) across different knowledge domains. The image training data consist of binary classification of 4 different clothing types (**Shirt**, **Boots**, **Sandals**, **Bags**) from the FashionMNIST dataset Xiao et al. (2017) and cat/dog classification from the **Dog** & **Cat** dataset Elson et al. (2007) (abbreviated as **Dog** in our plots). The training data domains for LLM evaluation consists of 9 topics: **Wikitext** Merity et al. (2016), **gsm8k** Cobbe et al. (2021), **PubmedQA** Jin et al. (2019), **HeadQA** Vilares & Gómez-Rodríguez (2019) , **SciQ** Welbl et al. (2017), **TriviaQA** Joshi et al. (2017), **TruthfulQA** Lin et al. (2022), **Hellaswag** Zellers et al. (2019), and **CommonsenseQA** Talmor et al. (2019). These domains are chosen specifically for their diversity to mimic topics seen by user-facing LLMs. In our experiments, we vary the difficulty of the unseen evaluation task by adjusting the training and evaluation data domains (see captions of Fig. 3 & 4 for more information).

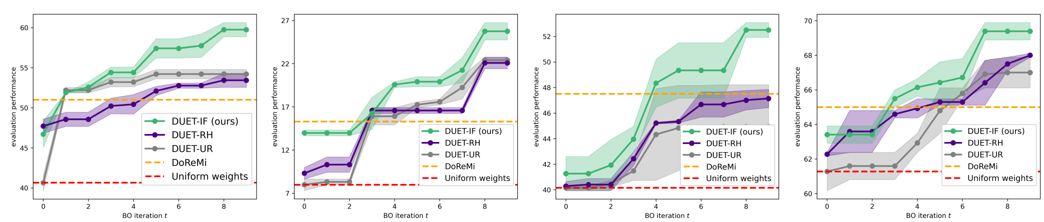
We compare our algorithm with several other baselines: **DoReMi** is a DRO-driven approach which optimizes the data mixture so that the trained ML model performs well for any evaluation task domain distributions (the original algorithm is used for pre-training, but in our LLM setting we fine-tune our LLM instead). The **Uniform weights** baseline samples from the training data domains uniformly to produce a data mixture of uniform ratio across different domains. We use DUET with a few different data selection methods: **DUET-IF** is our main method that uses our IF-driven estimator (Eq. 5) to select data mixtures at each BO iteration; **DUET-UR**, introduced in Sec. 3.2, uses the uniform random estimator and randomly selects data mixtures that satisfy the proposed mixing ratio; **DUET-RH** (**R**emove **H**armful) removes the 20% of data points with the lowest IF values from each data domain, before random sampling from the leftover data points. Other data selection baselines are discussed and shown in App. C.3. We use a sampling size of $k = 1$ and BO iterations $T = 30$ for image classification and $T = 10$ for language tasks. We also constrained the total number of selected data points to $M = 10000$.

5.2 Main result



(a) **Bag & Boots** (b) **Sneaker & Shirt** (c) **<u>Bag</u>** (d) **<u>Shirt, Dog</u>**

Figure 3: Comparison of **DUET**'s convergence with other baselines for unseen image classification task domains (higher is better) over 30 iterations. The subcaptions denote the evaluation task domains. **<u>Underlined evaluation tasks</u>** are more difficult because the evaluation task domains are removed from the training data (i.e., they are out-of-distribution).



(a) **TruthfulQA** (b) **<u>gsm8k</u>** (c) **<u>PubMedQA, HeadQA</u>** (d) **<u>Commonsense, Trivia</u>**

Figure 4: Results on unseen LLM evaluation task domains over 10 iterations based on the same setting as that in Fig. 3 (higher is better). The subcaptions denote the evaluation task domains. **<u>Underlined evaluation tasks</u>** are more difficult because the evaluation task domains are removed from the training data (i.e., they are out-of-distribution). All results are done in a 0-shot setting with no special prompts.

DUET finds better data mixtures. Our result (Fig. 3 & 4) shows that in different evaluation tasks, DUET finds data mixtures that produce better-performing ML models within a few iterations of feedback loops. The first column in Fig. 3 and 4 (for both image classification and LLM) consists of a relatively easier task where the evaluation task domain is found in the training task domains. In this case, DUET (green plot) uses feedback from the evaluation task to find the optimal data mixture with more emphasis on the relevant training data domain. On the other hand, DoReMi (orange dotted line) cannot adapt to the evaluation task and hence produces worse data mixtures. In the 2nd, 3rd and 4th columns, we increased the difficulty of our evaluation task by removing the evaluation task domain from our training domains (so, the task is out-of-domain). Surprisingly, even for these cases, DUET can still use the unseen evaluation task feedback to automatically improve the quality of the data mixture, achieving better model performance. This is because data from certain training domains could still be useful for the out-of-domain evaluation task (e.g., **Wikitext** data can still be helpful for mathematical questions in **gsm8k**). Hence, DUET uses feedback from the unseen evaluation task to place higher weights on more relevant training data domains. In App. C.2, we provide more experimental results for different combinations of evaluation tasks to showcase the effectiveness of DUET.

IF is an effective data selection method. Our result also shows that DUET-IF, which uses the IF-driven estimator (Eq. 5) to place more sampling emphasis on data points with high IF values, performs better than DUET-UR and DUET-RH. This showcases the effectiveness of using IF values for DUET to work effectively, as compared to other data selection methods.

5.3 Ablation study on different components of DUET

Next, we perform ablation studies to qualitatively analyze the influence of BO and different data selection methods on DUET's convergence to the optimal evaluation task performance. For clarity purpose, we only use the results for one evaluation task for our analysis. Fig. 5 shows that by naively

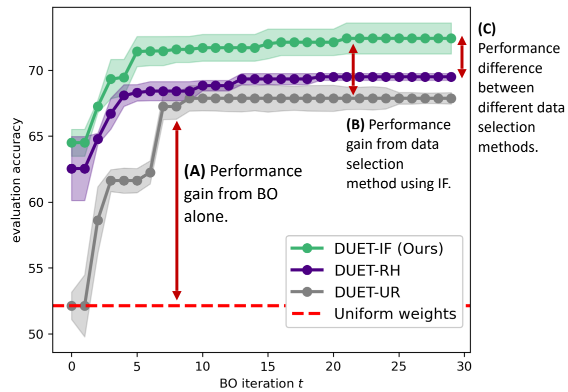


Figure 5: Performance gains attained by different components of our algorithm. BO and data selection methods both increase the performance of the ML model on an unseen evaluation task.

using a uniform data mixture and training an ML model, we can only achieve an evaluation task performance given by the red dotted line. With only BO, DUET automatically reconfigures the mixing ratio and attains performance gain (**A**) over the uniform training data mixture. Next, by incorporating data selection methods, such as using IF values in DUET-IF, we attain even more performance gains (**B**) indicated by the green plot. This is because using IF values helps to retrieve higher-quality data points at each iteration and reduces the estimation error of our inner problem (Sec. 3.2), yielding higher-quality data mixtures. Lastly, different data selection methods have varying effectiveness and yield different performance gains for DUET (**C**). Here, we see that the IF-driven estimator attains the best performing data mixture in DUET-IF as compared to other data selection methods (e.g. DUET-RH). We also show more ablation studies w.r.t. the use of larger sampling size k and other diversity-driven data selection methods in App. C.3. In general, our results show that increasing k improves the convergence of DUET. We also found that diversity-driven data selection methods Wang et al. (2024b) are too computationally expensive to be practical in our setting even with a greedy implementation Chen et al. (2018).

6 Conclusion

Our paper proposes DUET, a novel algorithm that exploits multiple rounds of feedback from a downstream unseen evaluation task to automatically optimize training data mixture. We provide theoretical guarantees of DUET and show that it finds better data mixtures in a variety of image and LLM evaluation tasks as compared to other conventional baselines. In light of the growing importance of our problem setting where we do not know the data in an unseen evaluation task is not known, we hope our work inspires future research to use coarse feedback from the evaluation task to refine the training data mixture for ML models.

REFERENCES

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *arXiv:2402.16827*, 2024.

Julyan Arbel, Olivier Marchal, and Hien D. Nguyen. On strict sub-gaussianity, optimal proxy variance and symmetry for bounded random variables, 2019.

Barry C Arnold, Narayanaswamy Balakrishnan, and Haikady Navada Nagaraja. *A first course in order statistics*. SIAM, 2008.

Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. No-regret bayesian optimization with unknown hyperparameters. In *Proc. ICML*, 2019.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proc. NeurIPS*, 2018.

Mayee F. Chen, Michael Y. Hu, Nicholas Lourie, Kyunghyun Cho, and Christopher Ré. Aioli: A unified optimization framework for language model data mixing. *arXiv:2411.05735*, 2024a.

Zhiliang Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Towards AutoAI: Optimizing a machine learning system with black-box and differentiable components. In *Proc. ICML*, 2024b.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proc. ICML*, pp. 844–853, 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

Laurens de Haan. Estimation of the minimum of a function using order statistics. *Journal of the American Statistical Association*, 76(374):467–469, 1981.

Jeremy Elson, John R Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. *CCS*, 7:366–374, 2007.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation. *arXiv:2310.15393*, 2024.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. ICML*, 2015.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL `https://zenodo.org/records/12608602`.

Jacob Gardner, Matt Kusner, Xu Zhixiang, Kilian Weinberger, and John Cunningham. Bayesian optimization with inequality constraints. In *Proc. ICML*, 2014.

Roman Garnett. *Bayesian Optimization*. Cambridge Univ. Press, 2023.

Stewart Greenhill, Santu Rana, Sunil Gupta, Pratibha Vellanki, and Svetha Venkatesh. Bayesian optimization for adaptive experimental design: A review. *IEEE Access*, 8:13937–13948, 2020. doi: 10.1109/ACCESS.2020.2966228.

Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. *arXiv:2310.05773*, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Proc. NeurIPS*, 2022.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *arXiv:1909.06146*, 2019.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv:1705.03551*, 2017.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proc. ICML*, 2017.

Anthony Lee and Steven J Miller. Generalizing the german tank problem. *arXiv:2210.15339*, 2022.

Tianshi Li, Sauvik Das, Hao-Ping Lee, Dakuo Wang, Bingsheng Yao, and Zhiping Zhang. Human-centered privacy research in the age of large language models. *arXiv:2402.01994*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv:2109.07958*, 2022.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proc. ICML*, 2017.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv:1609.07843*, 2016.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. *arXiv:1906.01827*, 2020.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. *arXiv:1301.2115*, 2013.

Edward O Pyzer-Knapp. Bayesian optimization for accelerated drug discovery. *IBM Journal of Research and Development*, 62(6):2–1, 2018.

Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with bayesian optimization. *arXiv:1707.05246*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *arXiv:1409.0575*, 2015.

Nikunj Saunshi, Arushi Gupta, Mark Braverman, and Sanjeev Arora. Understanding influence functions and datamodels via harmonic analysis. In *Proc. ICLR*, 2023.

Aili Shen, Jianzhong Qi, and Timothy Baldwin. A hybrid model for quality assessment of Wikipedia articles. In *Proc. Australasian Language Technology Association Workshop*, 2017.

Seungjae Shin, HeeSun Bae, Byeonghu Na, Yoon-Yeong Kim, and Il chul Moon. Unknown domain inconsistency minimization for domain generalization. In *Proc. ICLR*, 2024.

Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning: ”ingredients”, strategies, and open challenges. In *Proc. IJCAI*, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2015.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Proc. NeurIPS*, 2012.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, 2010.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv:1811.00937*, 2019.

Sebastian Shenghong Tay, Chuan-Sheng Foo, Daisuke Urano, Richalynn Leong, and Bryan Kian Hsiang Low. Bayesian optimization with cost-varying variable subsets. In *Proc. NeurIPS*, 2023.

Daniel Ting and Eric Brochu. Optimal sub-sampling with influence functions. *arXiv:1709.01716*, 2017.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.

David Vilares and Carlos Gómez-Rodríguez. HEAD-QA: A healthcare dataset for complex reasoning. In *Proc. ACL*, 2019.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *arXiv:2103.03097*, 2022.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Helpful or harmful data? fine-tuning-free shapley attribution for explaining language model predictions. In *Proc. ICML*, 2024a.

Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. Diversity measurement and subset selection for instruction tuning datasets. *arXiv:2402.02318*, 2024b.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Workshop on Noisy User-generated Text*, 2017.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv:2402.04333*, 2024.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Proc. NeurIPS*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv:1905.07830*, 2019.

Bo Zhang, Xiaoming Zhang, Yun Liu, Lei Cheng, and Zhoujun Li. Matching distributions between model and data: Cross-domain knowledge distillation for unsupervised domain adaptation. In *Proc. ACL*, 2021.

Wenyu Zhang, Li Shen, Wanyue Zhang, and Chuan-Sheng Foo. Few-shot adaptation of pre-trained networks for domain shift. *arXiv:2205.15234*, 2022.

Xiaoyu Zhang, Juan Zhai, Shiqing Ma, Chao Shen, Tianlin Li, Weipeng Jiang, and Yang Liu. Speculative coreset selection for task-specific fine-tuning. *arXiv:2410.01296*, 2024.

A Additional Discussions

A.1 Real-world examples of our problem setting

In our problem setting, (a) there is no direct access to the data (e.g., its domain, distribution, or labels) involved in the unseen evaluation task but (b) multiple rounds of coarse feedback (details covered in Sec. 2.2) can be gathered from the task using a trained ML model. Here, we provide several real-world examples in which such a setting occurs.

End-to-end encrypted conversations between LLM and users. This setting is specific to the conversational setting between a trained LLM and human users. LLM owners are interested in fine-tuning an LLM to converse well with some human-user demographics but due to real-world privacy concerns Li et al. (2024), conversations between a deployed LLM and users are end-to-end encrypted during test-time (`openai.com/enterprise-privacy`). So, an LLM owner does not have any knowledge of the conversation domain or the (unlabeled or labeled) data seen during test-time. Instead, they only receive a feedback on how well the LLM has performed in the conversation (e.g., ratings from the human user, how long each user stays on the applicaton). The LLM owner can collect multiple rounds of feedback over a period of time. Hence, they can exploit this feedback to iteratively refine the training data for the ML model. Many chat-driven applications (e.g., whatsapp, telegram) nowadays use end-to-end encrypted chats, so our problem setting is relevant here.

Model marketplace. In addition, there are other scenarios in which a model owner needs to improve an ML model without having access to the data involved in the unseen evaluation task. For instance, an ML model owner might rent or sell an image classification model in a model marketplace (e.g., `https://aws.amazon.com/marketplace/solutions/machine-learning`). However, the consumer might give feedback (e.g., how often the model makes mistakes) to the ML model owner in hope that the ML model owner can improve the model's performance on its own evaluation task. Furthermore, the images used by the consumer in its evaluation task are considered sensitive data, so the ML model owner does not know any data involved in the unseen evaluation task. Hence, the ML owner can only rely on feedback from the consumer to improve the model's performance.

A.2 More related works

Recently, a large class of data selection methods utilizing coresets, diversity or influence functions Zhang et al. (2024); Xia et al. (2024); Koh & Liang (2017) have been introduced to retrieve a smaller subset of data from an existing dataset. These data selection methods have become popular because they reduce training dataset size (which is an attractive feature when traning LLMs) and prior work Xia et al. (2024) showed that training a model with strategically selected data points allows it to perform better. However, these works, when used in isolation, do not work well in our setting because they do not exploit feedback from an unseen evaluation task. For example, even if we can retrieve a high-quality data subset from an original dataset of a training domain, that domain might not even be relevant to the unseen evaluation task. Hence, data selection methods on their own **are not applicable to our setting**. Instead, our paper's algorithm interleaves BO and data selection method together to exploit feedback from the unseen evaluation task to optimize our training data mixture.

A.3 Influence function and its calculations

Influence function (IF) Koh & Liang (2017) has been developed to study the influence of a single data point on an ML model's predictions. In this section we provide a summary of IF and its derivation. The influence of a data point z on the loss of a test data point (or a set of test data points) z_{test} for an ML model parameterized by θ is given by the closed-form expression:

$$\mathrm{IF}_{z,z_{\text{test}}} = -\nabla_\theta L(z_{\text{test}}, \theta)^T H_\theta^{-1} \nabla_\theta L(z, \theta), \tag{7}$$

where L is the loss function of the ML model and H is the hessian of the ML model w.r.t. parameters θ. In short, a data point is deemed more "influential" in reducing the model loss on a test data point if it has a higher IF value. As such, IF values have also become a popular method in selecting data points which are more helpful in training an ML model.

In our work, we segregated a validation dataset from each data domain's dataset, in which we use to derive the IF value of every training data point in that domain w.r.t. the validation dataset (after training a ML model over the training data). Then, we normalize these IF values (for data points in each data domain), allowing us to perform weighted random sampling at every BO iteration of our algorithm, obtaining a data subset of size n for a given data domain. This IF-weighted sampling is repeated for every data domain until we sample a dataset fulfilling the proposed mixing ratio at every BO iteration. Hence, the resulting data mixture contains more proportion of high-quality data points (based on IF values). A summary of the IF-weighted sampling process for one data domain is given in Alg. 2. In our algorithm, we repeat this procedure for every data domain.

Algorithm 2 IF-weighted sampling for **one data domain containing** dataset D

1: **Input:** number of data points n required for the given data domain (taken from the mixing ratio proposed at current iteration). Dataset $D = \{x_1, x_2, ..., x_{|D|}\}$, Influence value of each data point in data domain dataset D: $\mathcal{I} \triangleq [I_1, I_2, \ldots, I_{|D|}]$, small constant ϵ to avoid degenerate-case normalization.
2: Normalize the IF values into probabilities: $\mathcal{I}_{\text{normalized}} \triangleq [\frac{I_1+\min(\mathcal{I})+\epsilon}{\sum \mathcal{I}}, \frac{I_2+\min(\mathcal{I})+\epsilon}{\sum \mathcal{I}}, \ldots, \frac{I_{|D|}+\min(\mathcal{I})+\epsilon}{\sum \mathcal{I}}]$
3: Perform weighed sampling from dataset D according to weights given by $\mathcal{I}_{\text{normalized}}$ n times.

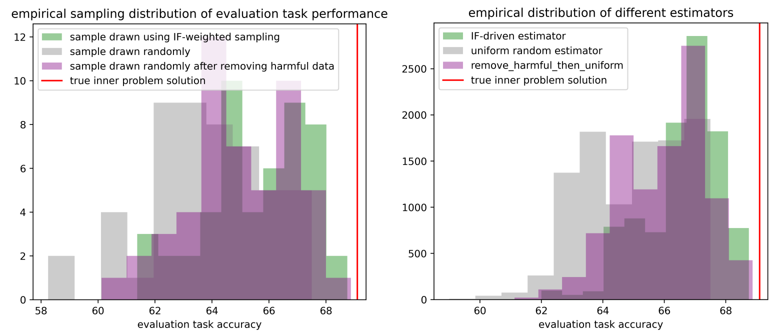
Precomputing IF values. In addition, we just need to precompute the IF values of every data point once before reusing them repeatedly at every BO iteration. This greatly improves our algorithm's efficiency and runtime, as compared to other methods (see next section).

A.4 Using other data selection methods to solve inner optimization problem

Data selection methods Albalak et al. (2024); Guo et al. (2024) have been used to retrieve a representative subset of data from larger datasets. We note that in our work different data selection methods can be interchanged to produce different estimators for the inner problem solution in line 4 and 5 of Algorithm 1. For example, instead of using the IF-driven estimator which performs weighted sampling based on each data point's IF values, one could simply remove data points from each data domain whose IF value falls below a certain threshold because they have a higher chance of being low-quality (Koh & Liang, 2017). However, our experiments (Sec. 5.2) have shown that this (labeled as DUET-RH in Fig. 3 & 4) does not work as well as the IF-driven estimator.

Other data selection methods can be considered as well. For example, coresets Zhang et al. (2024) have been used to distill a larger dataset into a smaller one while retaining some data properties (e.g., training loss gradients, the final performance of the trained ML model). This can also be used as an estimator of the inner problem solution: when a mixing ratio r is proposed by BO, we can derive the number of data points needed for each data domain (e.g., n number of data points for data domain A etc.). Then, we can retrieve a size n coreset from data domain A (and also do this similarly for the other data domains), before combining each coreset into the final dataset used to train the ML model. Despite being conceptually simple, coresets typically require much more computational resources because we need to account for the interaction between every data point. Furthermore, we need to recompute the coreset for every BO iteration because it depends on the mixing ratio, which changes every iteration. In contrast, IF values do not depend on the mixing ratio and can be precomputed and stored beforehand.

Lastly, uncertainty or diversity-driven Wang et al. (2024b) data selection methods can be used to select subsets of data that satisfy the proposed mixing ratio at every BO iteration. However, they also demand large amount of computational resources and require recomputation at every iteration. In App. C.3, we provided additional experimental results using the log-determinant Wang et al. (2024b); Chen et al. (2018), which captures the diversity of a sampled dataset, as a method to select data mixtures when estimating the solution of the inner problem. However, our results show that such methods do not work better than IF in DUET and are computationally expensive, making them unsuitable for our problem setting.



(a) **Empirical sampling distribution of** (b) **Empirical estimator distribution**

Figure 6: (a): Empirical distribution of evaluation task accuracy $\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}})$ from each data mixture sample $\mathcal{X}$ (b): empirical distribution of the estimators introduced in Sec. 3.2. The green histogram is our method of performing IF-weighted sampling to obtain data mixtures. The gray histogram is simply randomly sampling data mixtures with no data selection methods. The purple histogram is the method of removing 20% of the data points with the lowest IF values.

A.5 Empirical distributions of estimators from different data selection methods

We have introduced the IF-driven estimator in Sec. 3.2 as a method for us to estimate the solution of the inner problem. The IF-driven estimator performs IF-weighted sampling on data points from each data domain to produce data mixture samples (Eq. 5) constrained to a data mixing ratio r. Each data mixture sample is then used to train an ML model before obtaining a feedback on how well it has performed on the unseen evaluation task. Hence, this feedback on each data mixture sample is also a sampling distribution that we can empirically observe. Fig. 6a shows the sampling distribution of the evaluation task performance obtained from each data mixture. Empirically, we see that the negative of this distribution is similar to a truncate exponential distribution mentioned in Theorem 3.2 (We consider the negative of this random variable because our paper considers the evaluation task loss, but empirically we consider maximizing the evaluation task accuracy instead). In addition, the truncated exponential distribution is appropriate because it implies the unseen evaluation task loss is upper bounded at $y_r^* + c$ for a non-negative constant c; this is reasonable for many real-world settings (e.g., user rating is bounded).

Next, we plot the empirical distribution of the IF-driven estimator introduced in Eq. 5 in Fig. 6b. The distribution coincides with the estimator's distribution (formally, $y_r^* + \epsilon$) introduced in Theorem 3.2. From the estimator's distribution, we see that the IF-driven estimator (green histogram) has the lower bias and variance as compared to other estimators.

B Proofs

B.1 Proof of Theorem 3.1

Theorem 3.1. *$\mathcal{X}^*$, the optimal set of data points from $\mathcal{D}$, is the solution of the original problem (2) iff $r^* = ratio(\mathcal{X}^*)$ is the optimal mixing ratio solution of the reparameterized problem:*

$$\min_{r \in \mathbb{R}^N} \min_{\mathcal{X} \in S_r} \quad \mathcal{L}_{eval}(\theta_{\mathcal{X}}), \tag{3}$$

where $S_r \triangleq \{\mathcal{X} : \mathcal{X} \in \mathcal{D}, ratio(\mathcal{X}) = r, |\mathcal{X}| = M\}$ and $ratio(\mathcal{X}) = r$ means that the data points in $\mathcal{X}$ satisfies the given ratio $r \in \mathbb{R}^N$ from N data domains and $\|r\|_2 = 1$.

Proof. Theorem 3.1 can be proven in two steps. First, we restate the theoretical results from (Chen et al., 2024b) in Lemma B.1. This Lemma reparameterizes any optimization problem $\min_x f(x)$ (while retaining the solution set *exactly*) under some regular assumptions:

Lemma B.1. *Let $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^n$. Also, consider well-defined functions f over $\mathbb{R}^d \to \mathbb{R}$ and g over $\mathbb{R}^d \to \mathbb{R}^n$. Then x^* is a solution of $\arg\min_x f(x)$ if and only if $y^* = g(x^*)$ is a solution of the* *second optimization problem over domain* $\{y \mid \exists x, g(x) = y\}$:

$$\begin{aligned} \min_y \min_x \quad & f(x) \\ \textit{s.t.} \quad & g(x) = y \end{aligned}$$

The proof of Lemma B.1 can be found in Lemma C.1 of (Chen et al., 2024b). Next, we show that the objective function of problem 3 introduced in our optimization problem satisfies these assumptions, allowing us to apply the Lemma B.1 directly.

In our setting, we set $x \triangleq \mathcal{X}$, $f(x) \triangleq \mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}})$ and $g(x) \triangleq \text{ratio}(\mathcal{X})$. We can see that both functions are well-defined, where for any chosen input $\mathcal{X}$, there certainly exists an observed evaluation task loss $\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}})$ and mixing ratio ratio$(\mathcal{X})$. Lastly, by setting $y \triangleq r$, our optimization problem in problem (3) is of the identical form of the optimization problem shown in Lemma B.1. Therefore, our reparameterization process is valid.

□

B.2 Proof of Theorem 3.2

Theorem 3.2. *Let* $\{\mathcal{X}_1^{IF}, \ldots, \mathcal{X}_k^{IF}\}$ *be* k *samples of data mixtures drawn from* S_r *using IF-weighted sampling. Furthermore, assume each independent sample* $\mathcal{L}_{eval}(\theta_{\mathcal{X}_i^{IF}})$ *follows the shifted truncated exponential distribution* $y_r^* + \exp_t(\lambda, c)$*, for* $i = 1, 2, \ldots, k$ *where* $\exp_t(\lambda, c)$ *is a truncated exponential distribution governed by rate parameter* λ *and truncated at* $c > 0$*. Then, the IF-driven estimator* $\widetilde{y_r^*}$ *defined in Eq. 5 is a random variable:* $y_r^* + \epsilon$*, where* y_r^* *is the true inner problem solution of Eq. 4 and* ϵ *is a random noise variable with probability density function:*

$$\textit{pdf}_\epsilon(u) = \frac{\lambda k e^{-\lambda u}}{1 - e^{-\lambda c}} \left(\frac{e^{-\lambda u} - e^{-\lambda c}}{1 - e^{-\lambda c}} \right)^{k-1} \quad \textit{on } u \in [0, c] \, .$$

Proof. Let $X_1, X_2, \ldots, X_k$ be k samples randomly drawn from a sampling distribution and $X_{\text{min}} = \min\{X_1, X_2, \ldots, X_k\}$. This scenario mirrors the setting in Theorem 3.2. Our goal is to derive the distribution of X_{min} and show that it is exactly the same as the distribution of $\widetilde{y_r^*}$ shown in the Theorem 3.2.

If each random sample $X_i \sim \exp_t(\lambda, c)$, we first use a commonly known result Chen et al. (2024b) that the CDF of any truncated distribution on $[0, c]$ is $\frac{F(u) - F(0)}{F(c) - F(0)}$ where F is the CDF of the original distribution. Also, we note that for the untruncated exponential distribution, $F(u) = 1 - e^{-\lambda u}$. Hence, The CDF of X_{min} is

$$\begin{aligned} \text{cdf}_{(X_{\text{min}})}(u) &= 1 - \mathbb{P}(X_{\text{min}} \geq u) \\ &= 1 - \mathbb{P}(X_1 \geq u, X_2 \geq u, \ldots, X_k \geq u) \\ &= 1 - \left(1 - \frac{1 - e^{-\lambda u}}{1 - e^{-\lambda c}} \right)^k, \quad 0 \leq u \leq c. \end{aligned}$$

and so the PDF of X_{min} can be computed as

$$\begin{aligned} \text{pdf}_{(X_{\text{min}})}(u) &= \frac{\partial}{\partial u} F_{(X_{\text{min}})}(u) \\ &= \frac{\lambda k e^{-\lambda u}}{1 - e^{-\lambda c}} \left(\frac{e^{-\lambda u} - e^{-\lambda c}}{1 - e^{-\lambda c}} \right)^{k-1}, \quad 0 \leq u \leq c. \end{aligned}$$

In the original theorem, each sample X_i follows the shifted truncated exponential distribution $y_r^* + \exp_t(\lambda, c)$ where y_r^* is a constant. Hence, we can see that our estimator has the distribution of $y_r^* + X_{\text{min}}$ where X_{min} has the PDF above. Hence, the Theorem is proven by setting the random variable $\epsilon = X_{\text{min}}$.

□

B.3 Proof of Theorem 4.1

Theorem 4.1. *Let f be the outer problem objective defined in Eq. 6 with bounded RKHS norm: $||f||_\kappa = \sqrt{\langle f, f\rangle_\kappa}$. Also, let our IF-driven estimator for the inner problem solution be governed by the error distribution introduced in Theorem 3.2 with constant c and $\lambda = 1$. Let $A_{c,k} = \frac{c^2(1-e^{-c}-\frac{c}{2})^{k-1}}{(1-e^{-c})^k}$, where k is a fixed predecided sampling size. Then, running DUET over f using the LCB acquisition function found in (Chowdhury & Gopalan, 2017) at each BO iteration $t = 1, \ldots, T$ yields the following **attained average regret** Chen et al. (2024b) upper bound with probability at least $1 - \delta$:*

$$\lim_{T\to\infty} \frac{\tilde{R}_T}{T} \leq \frac{6(\sqrt[4]{\delta} + \sqrt{k})}{\sqrt[4]{\delta}k} + 2A_{c,k} + \frac{\sqrt{2A_{c,k}}}{\sqrt[4]{\delta}}.$$

Proof. We provide the proof of the sub-linear $\tilde{R}_T$ growth of DUET in Theorem 4.1 by establishing upper bounds of $|\mu_t(x) - f(x)|$ and ϵ_t separately at each BO iteration t and use the independence rule to bound their sum. To do so, we introduce the following two Lemmas.

Our first Lemma is taken from from known literature on Kernelized Bandits Chowdhury & Gopalan (2017) and provides the upper bound on difference between $f(x_t)$ and $\mu_t(x)$ at each BO iteration t.

Lemma B.2. *Let $||f||_\kappa = \sqrt{\langle f, f\rangle_\kappa} \leq B$. Also, assume that the observation noise associated with each BO iteration is R-sub-Gaussian with $R > 0$. Then with probability at least $1 - \delta$, the following holds for BO iteration $t \leq T$:*

$$|\mu_t(x) - f(x)| \leq \left(B + R\sqrt{2(\gamma_t + 1 + \ln(1/\delta)}\right)\sigma_t(x) \tag{8}$$

where γ_t is the maximum information gain after t observations and $\mu_t(x), \sigma_t^2(x)$ are mean and variance of posteror distribution of GP defined in Equation 1, with $\lambda = 1 + 2/T$.

Our second Lemma attempts to bound the expectation and variance of ϵ_t, the non-negative observation noise (in our case, it corresponds to the estimation error involved in solving the inner problem) at each BO iteration t. These expectation and variance will be used later to bound our cumulative regret.

Lemma B.3. *Let each observation noise ϵ_t of BO iteration t follow the same probability distribution as ϵ defined in Theorem 3.2 with sampling size k probability density function $f_{\epsilon_t}(u) = \frac{\lambda k e^{-\lambda u}}{1-e^{-\lambda c}}\left(\frac{e^{-\lambda u}-e^{-\lambda c}}{1-e^{-\lambda c}}\right)^{k-1}$ with $0 < c \leq 1$, $\lambda = 1$ and $u \in [0, c]$, then $\mathbb{E}(\epsilon_t) \leq \frac{6}{k} + \frac{2c^2((1-e^{-c})-\frac{c}{2})^{k-1}}{(1-e^{-c})^k}$ and $\mathrm{Var}(\epsilon_t) \leq \mathbb{E}(\epsilon_t)$.*

Proof. For $\lambda = 1$, we have that $f_{\epsilon_t}(u) = \frac{ke^{-u}}{1-e^{-c}}\left(\frac{e^{-u}-e^{-c}}{1-e^{-c}}\right)^{k-1}$ with $0 < c < 1$ and $u \in [0, c]$. Then, the expectation:

$$
\begin{aligned}
\mathbb{E}(\epsilon_t) &= \int_0^c u f_{\epsilon_t}(u)\, du \\
&= \int_0^c \frac{uke^{-u}}{1-e^{-c}} \left(\frac{e^{-u}-e^{-c}}{1-e^{-c}}\right)^{k-1} du \\
&= \frac{k}{(1-e^{-c})^k} \int_0^c ue^{-u} \left(e^{-u}-e^{-c}\right)^{k-1} du \\
&\overset{(1)}{\leq} \frac{k}{(1-e^{-c})^k} \int_0^c u \left(e^{-u}-e^{-c}\right)^{k-1} du \\
&\overset{(2)}{\leq} \frac{k}{(1-e^{-c})^k} \int_0^c u \left(\left(1-\frac{u}{2}\right)-e^{-c}\right)^{k-1} du \\
&\overset{(3)}{\leq} \frac{k}{(1-e^{-c})^k} \left(\frac{(u-2(1-e^{-c}))((1-e^{-c})-\frac{u}{2})^{k-1}(2(1-e^{-c})+(k-1)u+u)}{k(k+1)}\right)\Bigg|_{u=0}^{u=c} \\
&\overset{(4)}{=} \frac{1}{(1-e^{-c})^k} \left(\frac{(c-2(1-e^{-c}))((1-e^{-c})-\frac{c}{2})^{k-1}(2(1-e^{-c})+kc)+4(1-e^{-c})^{k+1}}{k+1}\right) \\
&\overset{(5)}{\leq} \frac{4(1-e^{-c})^{k+1}}{(k+1)(1-e^{-c})^k} + \frac{2kc^2((1-e^{-c})-\frac{c}{2})^{k-1}}{(k+1)(1-e^{-c})^k} + \frac{2((1-e^{-c})-\frac{c}{2})^{k-1}(1-e^{-c})}{(k+1)(1-e^{-c})^k} \\
&\overset{(6)}{\leq} \frac{6}{k} + \frac{2c^2((1-e^{-c})-\frac{c}{2})^{k-1}}{(1-e^{-c})^k}
\end{aligned} \tag{9}
$$

where $\overset{(1)}{\leq}$ makes use of the fact that $e^{-\lambda u} \leq 1$ for $u \in [0,c]$ with $c > 0$, $\overset{(2)}{\leq}$ uses the inequality $e^{-u} \leq 1 - \frac{u}{2}$ for $u \in [0,c]$, and $c \leq 1$, $\overset{(3)}{=}$ uses the fact that $e^{-\lambda c} < 1$, $\overset{(4)}{=}$ is derived by solving the definite integral by parts and substitution and $\overset{(4)}{=}$ simplifies the upper bound with algebraic manipulation.

Next, the upper bound of the variance of ϵ_t can be derived by

$$
\begin{aligned}
\mathrm{Var}(\epsilon_t) &= \int_0^c u^2 f_{\epsilon_t}(u)\, du \\
&\overset{(1)}{\leq} c\int_0^c u f_{\epsilon_t}(u)\, du \\
&\overset{(2)}{\leq} \int_0^c u f_{\epsilon_t}(u)\, du \\
&= \mathbb{E}(\epsilon_t)
\end{aligned} \tag{10}
$$

where $\overset{(1)}{\leq}$ makes use of the fact that ϵ_t lies in $[0,c]$ and $\overset{(2)}{\leq}$ makes use of the fact that $0 < c \leq 1$. This completes the proof on the bounds on $\mathbb{E}(\epsilon_t)$ and $\mathrm{Var}(\epsilon_t)$. □

Next, we observe that x_t at each BO iteration t is chosen via the IGP-LCB acquisition function (i.e., $x_t = \arg\min_x \mu_{t-1}(x) - \beta_t\sigma_{t-1}(x)$ and $\beta_t = B + R\sqrt{2(\gamma_{t-1}+1+\ln(1/\delta_1))}$ where the observation noise associated with each BO iteration is R-sub Gaussian). Thus, we can see that at each iteration $t \geq 1$, we have $-\mu_{t-1}(x_t) + \beta_t\sigma_{t-1}(x_t) \geq -\mu_{t-1}(x^*) + \beta_t\sigma_{t-1}(x^*)$. It then follows that for all $t \geq 1$ and with probability at least $1 - \delta_1$,

$$
\begin{aligned}
|f(x_t) - f(x^*)| &\overset{(1)}{\leq} f(x_t) - \mu_{t-1}(x^*) - \beta_t \sigma_{t-1}(x^*) \\
&\overset{(2)}{\leq} f(x_t) - \mu_{t-1}(x_t) + \beta_t \sigma_{t-1}(x_t) \\
&\leq \beta_t \sigma_{t-1}(x_t) + |\mu_{t-1}(x_t) - f(x_t)| \\
&\leq 2\beta_t \sigma_{t-1}(x_t)
\end{aligned}
\tag{11}
$$

Therefore, by setting $\delta_1 = \delta_2 = \sqrt{\delta}$, it follows that with probability $1 - \delta$ (this follows by rule of independence applied to the upper bound of events $\sum_{t=1}^{T} |f(x_t) - f(x^*)|$ and $\sum_{t=1}^{T} \epsilon_t$) that our **attained cumulative regret** can be bounded as

$$
\begin{aligned}
\tilde{R}_T &= \sum_{t=1}^{T} |\tilde{y}_t - f(x^*)| \\
&= \sum_{t=1}^{T} |f(x_t) - f(x^*) + \epsilon_t| \\
&\overset{(1)}{=} \sum_{t=1}^{T} |f(x_t) - f(x^*)| + \sum_{t=1}^{T} \epsilon_t \\
&\overset{(2)}{\leq} 2\beta_T \sum_{t=1}^{T} \sigma_{t-1}(x_t) + \sum_{t=1}^{T} \epsilon_t \\
&\overset{(3)}{=} 2\left(B + R\sqrt{2(\gamma_T + 1 + \ln(1/\sqrt{\delta}))}\right) \sum_{t=1}^{T} \sigma_{t-1}(x_t) + \sum_{t=1}^{T} \epsilon_t \\
&\overset{(4)}{\leq} 2\left(B + R\sqrt{2(\gamma_T + 1 + \ln(1/\sqrt{\delta}))}\right) \sum_{t=1}^{T} \sigma_{t-1}(x_t) + \sum_{t=1}^{T} \mathbb{E}(\epsilon_t) + \sum_{t=1}^{T} \sqrt{\frac{\mathrm{Var}(\epsilon_t)}{\delta_2}} \\
&\overset{(5)}{=} 2\left(B + R\sqrt{2(\gamma_T + 1 + \ln(1/\sqrt{\delta}))}\right) O(\sqrt{T\gamma_T}) + \sum_{t=1}^{T} \mathbb{E}(\epsilon_t) + \sum_{t=1}^{T} \sqrt{\frac{\mathrm{Var}(\epsilon_t)}{\delta_2}} \\
&= O\left(\sqrt{T}(B\sqrt{\gamma_T} + R\gamma_T)\right) + \sum_{t=1}^{T} \mathbb{E}(\epsilon_t) + \sum_{t=1}^{T} \sqrt{\frac{\mathrm{Var}(\epsilon_t)}{\delta_2}} \\
&\overset{(6)}{=} O\left(\sqrt{T}(B\sqrt{\gamma_T} + \frac{c^2\gamma_T}{4})\right) + \sum_{t=1}^{T} \mathbb{E}(\epsilon_t) + \sum_{t=1}^{T} \sqrt{\frac{\mathrm{Var}(\epsilon_t)}{\delta_2}}
\end{aligned}
\tag{12}
$$

where we have followed the attained cumulative regret proof in Chen et al. (2024b) closely and used the following facts:

- $\overset{(1)}{=}$ uses the fact that ϵ_t is non-negative in our problem setting (Theorem 3.2).
- $\overset{(2)}{\leq}$ is derived from Eq. equation 11.
- $\overset{(3)}{=}$ uses the definition of β_T in IGP-LCB acquisition function Chowdhury & Gopalan (2017) w.r.t. $\delta_1 = \sqrt{\delta}$
- $\overset{(4)}{\leq}$ uses Chebyshev's inequality over ϵ_t with probability at least $1 - \delta_2$.
- $\overset{(5)}{=}$ uses $\sum_{t=1}^{T} \sigma_{t-1}(x_t) \leq O(\sqrt{T\gamma_T})$ as shown in **Lemma 4** by Chowdhury & Gopalan Chowdhury & Gopalan (2017).

- $\overset{(6)}{=}$ uses the fact that ϵ_t is bounded on $[0, c]$ and all bounded random variables are R-sub-Gaussian with $R = \frac{c^2}{4}$ Arbel et al. (2019).

Next, we need to derive the upper bound of $\sum_{t=1}^{T} \mathbb{E}(\epsilon_t) + \sum_{t=1}^{T} \sqrt{\frac{\mathrm{Var}(\epsilon_t)}{\delta_2}}$ w.r.t. T. This can be done by using the upper bound of the expectation and variance of ϵ_t proven in Lemma B.3:

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}(\epsilon_t) + \sum_{t=1}^{T} \sqrt{\frac{\mathrm{Var}(\epsilon_t)}{\delta_2}} &\overset{(1)}{\le} \sum_{t=1}^{T} \left(\frac{6}{k} + \frac{2c^2((1-e^{-c}) - \frac{c}{2})^{k-1}}{(1-e^{-c})^k} \right) + \sum_{t=1}^{T} \sqrt{\frac{6}{\delta_2 k} + \frac{2c^2((1-e^{-c}) - \frac{c}{2})^{k-1}}{\delta_2 (1-e^{-c})^k}} \\ &= \frac{6T}{k} + \frac{2Tc^2((1-e^{-c}) - \frac{c}{2})^{k-1}}{(1-e^{-c})^k} + T\sqrt{\frac{6}{\delta_2 k} + \frac{2c^2((1-e^{-c}) - \frac{c}{2})^{k-1}}{\delta_2 (1-e^{-c})^k}}\end{aligned} \tag{13}$$

where $\overset{(1)}{\le}$ uses Lemma B.3 directly.

Then, it follows from Eq. 12 and 13 that with probability $1-\delta$ and $\delta_2 = \sqrt{\delta}$, the **attained cumulative regret** $\tilde{R}_T$ at iteration T is upper bounded by:

$$\tilde{R}_T \le O\left(\sqrt{T}(B\sqrt{\gamma_T} + \frac{c^2 \gamma_T}{4}) \right) + \frac{6T}{k} + \frac{2Tc^2((1-e^{-c}) - \frac{c}{2})^{k-1}}{(1-e^{-c})^k} + T\sqrt{\frac{6}{\delta_2 k} + \frac{2c^2((1-e^{-c}) - \frac{c}{2})^{k-1}}{\delta_2 (1-e^{-c})^k}} \tag{14}$$

Finally we set $A_{c,k} = \frac{c^2(1-e^{-c}-\frac{c}{2})^{k-1}}{(1-e^{-c})^k}$. As $T \to \infty$, with probability $1-\delta$ and $\delta_2 = \sqrt{\delta}$, the attained *average* regret converges to:

$$\begin{aligned}\lim_{T\to\infty} \frac{\tilde{R}_T}{T} &\overset{(1)}{\le} \frac{6}{k} + \frac{2((1-e^{-c}) - \frac{c}{2})^{k-1}}{(1-e^{-c})^k} + \sqrt{\frac{6}{\delta_2 k} + \frac{2((1-e^{-c}) - \frac{c}{2})^{k-1}}{\delta_2 (1-e^{-c})^k}} \\ &\overset{(2)}{\le} \frac{6}{k} + \sqrt{\frac{6}{\delta_2 k}} + 2A_{c,k} + \sqrt{\frac{2A_{c,k}}{\delta_2}} \\ &\le \frac{6(\sqrt[4]{\delta} + \sqrt{k})}{\sqrt[4]{\delta} k} + 2A_{c,k} + \sqrt{\frac{2A_{c,k}}{\delta_2}}\end{aligned} \tag{15}$$

where $\overset{(1)}{\le}$ divides Eq. 14 by T throughout, eliminating the O expression and $\overset{(2)}{\le}$ uses the subsitition of $A_{c,k}$ and triangle inequality. This completes our proof for the attained average regret in Theorem 4.1. □

B.4 Extending theoretical analysis based on different data selection methods

Readers might be interested in how different data selection methods used to create different estimators affect our theoretical analysis. Here, we provide details on how one could replicate our paper's theoretical analysis to different estimators.

Step 1. Establish the sampling distribution of $\mathcal{L}_{\textbf{eval}}(\theta_{\mathcal{X}})$. Using a particular data selection method, one obtains k data mixture samples $\{\mathcal{X}_1, \ldots, \mathcal{X}_k\}$ (in our paper, these samples are obtained via weighted sampling based on each data point's IF values). Then, one trains an ML model for each data mixture and obtain the evaluation task loss for each resulting ML model, yielding $\{\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_1}), \ldots, \mathcal{L}_{\text{eval}}\theta_{\mathcal{X}_k}\}$. From this set, one can empirically derive the sampling distribution of each sample $\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_i})$. In Theorem. 3.2, we assumed that each sample $\mathcal{L}_{\text{eval}}(\theta_{\mathcal{X}_i})$ follows the truncated exponential distribution. However, different data selection methods would certainly lead to different empirical sampling distributions.

Step 2. Derive an estimator's empirical distribution. Next, we need to theoretically derive the 1st-order statistic Arnold et al. (2008) of the empirical sampling distribution from Step 1, since we use the 1st-order statistic as our estimator. The procedure to do so is shown in App. B.2 and uses a fairly standard procedure to derive the distribution of order statistics. For subsequent analysis to be tractable, the PDF of the 1st-order statistic should have a closed form (hence, a simpler sampling distribution in Step 1 is preferred). More importantly, the estimator's empirical distribution **should be R-sub-gaussian** for a fixed $R > 0$. This is because for the regret-analysis proof in Eq. 12 to hold true, the observation noise in the BO process should be R-sub-Gaussian. Fortunately, a large family of random distributions, including our IF-driven estimator introduced in this paper, are all R-sub-Gaussian (e.g., exponential family, all bounded random variables).

Step 3. Derive the upper bound of estimator's expectation and variance. Next, we derive the upper bound of the 1st-order statistic's expectation and variance as shown in Lemma. B.3.

Step 4. Derive attainable cumulative regret. Lastly, we analyze the convergence rate of our algorithm using the growth of *attained cumulative regret* Chen et al. (2024b) $\tilde{R}_T = \sum_{t=1}^{T} |\widetilde{y^*_{r_t}} - f(r_t)| = \sum_{t=1}^{T} |f(r^*) + \epsilon_t - f(r_t)|$ for T BO iterations. Since the error term ϵ_t has the same expectation and variance of our estimator, we can use the results from Step 3 to derive our regret bound (as shown in Eq. 12).

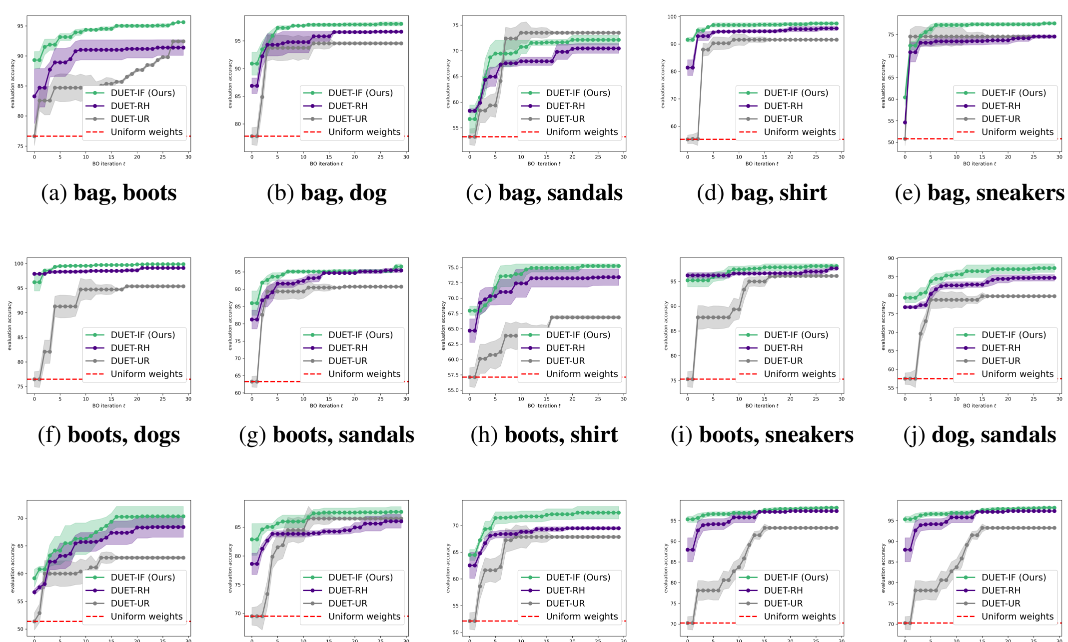
C Additional Experimental Results and Discussions

C.1 Additional details on experimental setup

In this section, we provide additional details in our experiments for ease of reproduceability. Throughout our experiments, we used the SE kernel with lengthscale parameters learnt from historical observations via maximum-likelihood Williams & Rasmussen (2006). In our LCB acquisition function Greenhill et al. (2020), we set $\beta_t = 0.1$ (see Alg. 1) throughout our experiments. Furthermore, we need to perform constrained BO Gardner et al. (2014) in our experiments because the inputs to our optimization problem is a data mixing ratio r whose sum of entries is constrained to 1. BoTorch allows us to implement such constraints (`botorch.org/docs/constraints`) easily. We used an exploration parameter of $\beta_t = 0.5$ in our BO acquisition step. For the image classification task, we purposely flip 10 percent of the training image labels to make our datasets noiser. Lastly, all evaluation for language tasks is done on **llm-harness** Gao et al. (2024) with default 0-shot settings. Hence, it is possible some of our paper's results differ from those reported in other papers (due to different prompting and inference settings). However, our paper's emphasis is on improving the ML model's performance with a few rounds refinement on the training data mixture. Hence, we expect DUET to work well even in other inference settings.

C.2 Additional experimental results on different combination of evaluation task domains

We also conducted experiments with different combinations of evaluation task domains for the image classification task. From the results, we can see DUET-IF with the IF-driven estimator (green plot) consistently outperforms other baselines that use different data selection methods (introduced in Sec. 5.2).



(a) **bag, boots** (b) **bag, dog** (c) **bag, sandals** (d) **bag, shirt** (e) **bag, sneakers**

(f) **boots, dogs** (g) **boots, sandals** (h) **boots, shirt** (i) **boots, sneakers** (j) **dog, sandals**

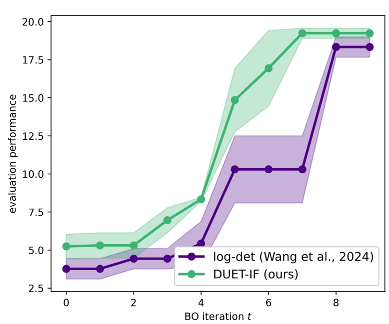
(k) **dog, shirt** (l) **dog, sneakers** (m) **sandals, shirt** (n) **sandal, sneakers** (o) **shirt, sneakers**

Figure 7: Results on different combination of image classification evaluation tasks to demonstrate the performance of DUET to refine the training data mixture as compared to other estimators and uniform weights. To reduce plot clutter, we have removed DoReMi Xie et al. (2023) because we found that it does not perform better than DUET across different combinations.

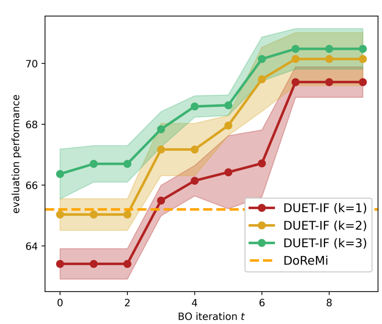
C.3 Additional ablation studies with diversity-driven data selection and different sampling size k

Comparison with diversity-driven data selection methods While our paper introduced IF values as a data selection method to solve the inner problem (see Alg. 2 in App. A.3), other data selection methods can be used to approximately solve the inner problem (4) as well. One class of work is diversity-driven subset selection Wang et al. (2024b) that selects a subset of data that is the most diverse and representative of the original dataset. This is done by finding a data mixture with the largest log-determinant for its data-feature kernel. We use this method as an estimator to estimate the solution of our inner problem (4) and compare its performance with our IF-driven estimator in Fig. 8(a), under the same out-of-domain gsm8k setting as that in Fig. 5.2. Due to the large computational complexity of computing matrix determinants, we restricted the total number of data points to $M = 1000$ (instead of 10000 used in our main results). This large complexity arises because practical methods to calculate the determinant of a $n \times n$ matrix typically have a $\mathcal{O}(n^3)$ runtime complexity. On top of this, the greedy implementation Chen et al. (2018) to find the data mixture with highest log-determinant data features has a runtime of $\mathcal{O}(mn)$, where m is the number of data points we need to retrieve at each BO iteration. Hence, this results in a runtime complexity of $\mathcal{O}(mn^4)$, which is too slow for larger datasets. Some implementation tricks (such as caching) can be used to speed up the computation of, but we still find diversity-driven data selection methods too slow to be practical in DUET. In fact, computing a single round of inner problem approximation took around 14 hours when computing the log-determinant (since we need to iterate through all data points and recompute the determinant of data feature matrix repeatedly). On the other hand, computing the IF-driven estimator only took less than 1 hour.

Ablation study with varying sampling size k Theorem 3.2 & 4.1 have highlighted how sampling size k could theoretically affect the performance of DUET. In our main result, we showed that using a sampling size $k = 1$ is sufficient for us to achieve better data mixtures than other baselines. In Fig. 8(b), we evaluated DUET with increasing number of sampling size k when using the IF-estimator. Our results show that DUET with larger sampling size k (green plot) leads to an ML model with better performance than that with a smaller sampling size. This agrees with our theory that larger k can reduce the estimation error of our estimator for the inner optimization problem (4) and also leads to a smaller attained cumulative regret (Theorem 4.1).



(a) **DUET vs. log-det Wang et al. (2024b)**



(b) **Ablation study w.r.t. sampling size k**

Figure 8: (a): Comparison of DUET paired with diversity-driven data selection methods Wang et al. (2024b) (marked as **log-det** in our plots) and DUET paired with IF-estimator (DUET-IF). (b): Ablation study of DUET-IF on sampling size k.