A Lagrangian Duality Approach to Active Learning

Juan Elenter
University of Pennsylvania
elenter@seas.upenn.edu

Navid NaderiAlizadeh
University of Pennsylvania
nnaderi@seas.upenn.edu

Alejandro Ribeiro
University of Pennsylvania
aribeiro@seas.upenn.edu

Abstract

We consider the pool-based active learning problem, where only a subset of the training data is labeled, and the goal is to query a batch of unlabeled samples to be labeled so as to maximally improve model performance. We formulate the problem using constrained learning, where a set of constraints bounds the performance of the model on labeled samples. Considering a primal-dual approach, we optimize the primal variables, corresponding to the model parameters, as well as the dual variables, corresponding to the constraints. As each dual variable indicates how significantly the perturbation of the respective constraint affects the optimal value of the objective function, we use it as a proxy of the informativeness of the corresponding training sample. Our approach, which we refer to as Active Learning via Lagrangian dualitY, or ALLY, leverages this fact to select a diverse set of unlabeled samples with the highest estimated dual variables as our query set. We demonstrate the benefits of our approach in a variety of classification and regression tasks and discuss its limitations depending on the capacity of the model used and the degree of redundancy in the dataset. We also examine the impact of the distribution shift induced by active sampling and show that ALLY can be used in a generative mode to create novel, maximally-informative samples.

1 Introduction

One of the key drivers of the recent progress in machine learning is the availability of massive, high-quality datasets, which enables training models comprising millions, or even billions, of parameters [1, 2, 3]. Nevertheless, in some areas, such as healthcare, obtaining *labeled* training data is challenging and/or expensive [4, 5, 6]. This has given rise to a class of approaches, collectively referred to as *active learning*, whose goal is to minimize the labeling effort for training machine learning models.

Active learning methods aim to improve data efficiency by querying the labels of samples presumed informative, in a feedback-driven fashion. In recent years, the pool-based active learning setting, in which queries are drawn from a large, static pool of unlabeled samples has drawn significant attention [7, 8, 9]. This is due to the abundance of such unlabeled pools and the compatibility of the pool-based setting with the training of deep neural networks.

Most active learners rely on defining a notion of *informativeness* of a given sample, such as model uncertainty [10, 11], expected model change [12, 13] or expected error reduction [14, 15]. However, in the batch setting, where multiple samples are queried simultaneously, not contemplating the information overlap between the samples can lead to sub-optimal queries. Consequently, batch *diversity* needs to be taken into account, often at the expense of individual sample informativeness [16].

In this paper, we demonstrate how a *constrained learning* formulation of the problem enables the use of *Lagrangian duality* for detecting informative samples. In particular, we bound the loss incurred by each sample, and use the dual variables associated to these constraints as a measure of informativeness.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

We show that dual variables are directly related to the variations of the average optimal loss over the entire data distribution, which motivates our approach.

Through an iterative primal-dual strategy, we optimize the model parameters as well as the dual variables. We then leverage the learned embedding space [17, 18] to train a *dual regression head* that estimates the dual variable associated to each unlabeled sample. Our proposed active learning approach, which we refer to as Active Learning via Lagrangian dualitY, or ALLY can then be used to select a diverse and informative set of samples. We also argue that, under certain conditions, the strategy put forward is not severely impacted by the distribution shift induced by active sampling.

We evaluate the performance of ALLY on a suite of classification and regression tasks, and show that it performs similar to or better than state-of-the-art batch active learning methods. We further demonstrate how the trained backbone, alongside the dual regression head, enable the generation of novel samples that can be optimized to be maximally informative, shedding light on the interpretability of the proposed active learning framework.

2 Related Work

2.1 Active Learning

The literature on active learning is voluminous and a myriad of strategies for the pool-based setting have been proposed [7]. In what follows, we describe some of the approaches most connected to our work.

A simple way of measuring model uncertainty is by computing the entropy of the predicted class distribution. Designed for the sequential case, Entropy Sampling [10] selects the unlabeled sample with highest associated output entropy. Among the relevant methods is BADGE [15], which employs a lower bound on the norm of the gradients in the final layer of the network as a measure of informativeness. BADGE balances diversity and informativeness by using the k-MEANS++ seeding algorithm to select a batch with large Gram determinant in the gradient space. BAIT [19] builds on traditional, Information Matrix based methods to efficiently select a batch that optimizes a bound on the MLE error in a two stage manner. Other methods that propose notions of informativeness are BALD [20], which uses the mutual information between predictions and model parameters as an uncertainty measure, and Learning Loss [21], which trains a loss prediction module and queries the samples that hypothetically generate high errors (and, thus, large model updates). VAAL [22] and WAAL [23] are methodologically similar to [21] in the sense that they all use a multipartite system consisting of a feature encoder, a prediction head and an auxiliary estimator. This is also the case for the strategy presented in this paper.

Some diversity-promoting approaches are compatible with many informativeness measures. A popular approach is to cluster the samples of the unlabeled set and then select informative points from each cluster [24, 25, 26]. In [27], Monte Carlo sampling is used to simulate sequences of length b of the sequential algorithm, and then a *best-matching* combination of the sequences is used to build a batch. A simpler approach is to select the b most informative points after a stochastic perturbation of the informativeness scores [28].

Some methods do not enforce informativeness explicitly, but rather query a set of data points that is maximally representative of the entire unlabeled set. Coreset [8], for instance, formulates pool-based active learning as a core-set selection problem, and aims to identify a set of points that geometrically covers the entire representation space. To do this, Coreset selects the batch that when added to the labeled set, minimizes the maximum distance between labeled and unlabeled examples. Coreset is compatible with deep neural networks and can be used in both regression and classification settings. Similarly, DAL [29] emphasizes representativeness by framing active learning as a binary classification task and selecting queries that maximize the similarity between the labeled and unlabeled set.

2.2 Constrained Learning

The need to tailor the behavior of machine learning systems has led to the development of a constrained learning theory. These developments [30, 31, 32] have shown that, from a PAC (Probably Approximately Correct) perspective, learning under requirements is as hard as classical learning and

that it can be done in practice through primal-dual learners. This has led to numerous applications across several areas of ML such as federated learning [33], fairness [30], stability of neural networks [34, 35], adversarial robustness [36] and data augmentation [37].

3 Problem Formulation

3.1 Batch Active Learning

Let $\mathfrak{D}$ denote a probability distribution over data pairs $(\boldsymbol{x}, y)$, where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ represents a feature vector (e.g., the pixels of an image) and $y \in \mathcal{Y} \subseteq \mathbb{R}$ represents a label or measurement. In classification tasks, $\mathcal{Y}$ is a subset of $\mathbb{N}$, whereas in regression, $\mathcal{Y} = \mathbb{R}$.

Initially, a set $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i \in \mathcal{N}_\mathcal{L}}$ of data pairs, or labeled samples, is available, coming from a probability distribution $\mathfrak{D}$. This set is used to learn a predictor $f(\mathbf{x}; \mathcal{L}) : \mathcal{X} \to \mathcal{Y}$ from a hypothesis class $\mathcal{F}$. Note that dependence on the set used to learn f is made explicit. Then, a batch $\mathcal{B}$ of samples, or *queries*, is selected from a pool of unlabeled samples $\mathcal{U} = \{\boldsymbol{x}_i\}_{i \in \mathcal{N}_\mathcal{U}}$ and sent to an oracle for labeling. The goal is to the select the batch that minimizes the expected loss over the natural data distribution. More precisely, we formulate the Batch Active Learning (BAL) problem as

$$\mathcal{B}^\star = \underset{\mathcal{B} \subseteq \mathcal{U} \,:\, |\mathcal{B}| \leq b}{\arg\min} \; \min_{f \in \mathcal{F}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathfrak{D}} \left[\ell \left(f(\mathbf{x}; \mathcal{L} \cup \mathcal{B}), y \right) \right], \tag{BAL}$$

where b, referred to as the *budget*, represents the maximum cardinality of $\mathcal{B}$ and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function (e.g., cross-entropy loss or mean-squared error).

This process is typically repeated multiple times. At each iteration, two main steps are performed: (i) selecting $\mathcal{B}$ and updating the sets: $\mathcal{L}^{(t)} = \mathcal{L}^{(t-1)} \cup \mathcal{B}$ and $\mathcal{U}^{(t)} = \mathcal{U}^{(t-1)} \setminus \mathcal{B}$, and (ii) obtaining the predictor f with the aggregate set of labeled samples $\mathcal{L}^{(t)}$. Steps (i) and (ii) correspond to the outer and inner minimization problems in (BAL), respectively. In what follows, we focus on a single iteration, and thus obviate the dependence on the iteration t and the set used to ease the notation.

3.2 Constrained Statistical Learning

Most active learning methods in the literature [8, 15, 20, 5, 7] formulate step (ii) above as an *unconstrained* Statistical Risk Minimization (SRM) problem [38],

$$\min_{f \in \mathcal{F}} \; \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathfrak{D}} \left[\ell \left(f(\boldsymbol{x}), y \right) \right]. \tag{SRM}$$

Our approach, alternatively, uses a *Constrained* Statistical Learning (CSL) formulation,

$$P^\star = \min_{f \in \mathcal{F}} \quad \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathfrak{D}} \left[\ell \left(f(\boldsymbol{x}), y \right) \right] \tag{CSL-a}$$

$$\text{s.t.} \quad \ell' \left(f(\boldsymbol{x}), y \right) \leq \epsilon(\boldsymbol{x}), \quad \mathfrak{D}_{\boldsymbol{x}}\text{–a.e.} \tag{CSL-b}$$

where $\ell' : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a secondary loss function, $\epsilon : \mathcal{X} \to \mathbb{R}$ is a mapping from each data point to a corresponding constraint upper bound, and $\mathfrak{D}_{\boldsymbol{x}}$ denotes the marginal distribution over $\mathcal{X}$. Note that the objective function in (CSL-a) is the same as in (SRM), but the secondary loss is required to be bounded $\mathfrak{D}_{\boldsymbol{x}}-$almost everywhere.

Letting $\lambda : \mathcal{X} \to \mathbb{R}^+$ denote the dual variable function, the Lagrangian associated to (CSL) can be written as

$$\begin{aligned} L(f, \lambda) &= \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathfrak{D}} \left[\ell(f(\boldsymbol{x}), y) \right] \quad + \int_{\mathcal{X}, \mathcal{Y}} \lambda(\boldsymbol{x}) (\ell'(f(\boldsymbol{x}), y) - \epsilon(\boldsymbol{x})) p(\boldsymbol{x}, y) d\boldsymbol{x} dy \\ &= \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathfrak{D}} \Big[\ell(f(\boldsymbol{x}), y) + \lambda(\boldsymbol{x}) (\ell'(f(\boldsymbol{x}), y) - \epsilon(\boldsymbol{x})) \Big], \end{aligned}$$

where it is implicitly assumed that the conditional distribution $p(y|\boldsymbol{x})$ is a Dirac delta distribution, i.e., y is a deterministic function of $\boldsymbol{x}$. This leads to the dual problem,

$$D^\star = \max_{\lambda \in \Lambda} \; \min_{f \in \mathcal{F}} L \left(f, \lambda(\boldsymbol{x}) \right), \tag{D-CSL}$$

where $\Lambda := \{\lambda \,|\, \lambda(\boldsymbol{x}) \geq 0, \; \mathfrak{D}_{\boldsymbol{x}}\text{–a.e.}\}$. There are three main motivations for this infinite programming formulation:

1. **Access to variations of $P^\star$**: As we show in Theorem 3.2, this formulation gives us access to $\partial P^\star(\epsilon(\boldsymbol{x}))$, enabling the use of *dual variables* as an indicator of the *informativeness* of the training samples.
2. **Considering both average and worst-case losses**: The most informative samples often lie in the tails of the distribution $\mathfrak{D}$. Those samples appear less frequently in the dataset and thus, models obtained by solving (SRM) can achieve low *average* errors without learning to classify/regress them correctly. Our formulation bridges the gap between the (SRM) and a worst-case Feasibility Statistical Learning (FSL) formulation,

$$P^\star = \min_{f \in \mathcal{F}} \quad 0 \tag{FSL-a}$$

$$\text{s.t.} \quad \ell\left(f(\boldsymbol{x}), y\right) \leq \epsilon(\boldsymbol{x}), \quad \mathfrak{D}_{\boldsymbol{x}}\text{–a.e.} \tag{FSL-b}$$

   which considers the worst-case loss.
3. **Adaptive regularization**: For a fixed dual variable λ, the Lagrangian is a regularized objective, where ℓ' acts as a regularizing functional. Thus, the max-min formulation in D-CSL can be viewed as a regularized minimization, where the regularization weight is updated during the training procedure according to the degree of constraint satisfaction or violation.

The dual problem can be interpreted as finding the tightest lower bound on $P^\star$. In the general case, $D^\star \leq P^\star$, which is known as weak duality. Nevertheless, under certain conditions, $D^\star$ attains $P^\star$ (strong duality) and we can derive a relation between the solution of (D-CSL) and the sensitivity of $P^\star$ with respect to $\epsilon(\boldsymbol{x})$. See Appendix A for more details.

In the following, we define the *Fréchet subdifferential* of a convex function, which allows us to justify the use of dual variables as a measure of *informativeness* of a sample.

Definition 3.1. Let U, V be Banach spaces. The Fréchet subdifferential of a convex function $P : U \to V$ at $u \in U$ is defined as:

$$\partial P(u) = \{z \in U^* \,:\, P(v) - P(u) \geq \langle z\,,\, v - u\rangle \text{ for all } v \in U\},$$

where U^* denotes the topological dual space of U, and $\langle z\,,\, v - u\rangle = \mathbb{E}_{\mathfrak{D}}\left[z(\boldsymbol{x})(v(\boldsymbol{x}) - u(\boldsymbol{x}))\right]$.

Having the above definition, we state following theorem, which characterizes the variations of $P^\star$ (the optimum value of CSL) as a function of the constraint tightness $\epsilon(\boldsymbol{x})$.

Theorem 3.2. *If the problem* (CSL) *is strongly dual, then for any $\boldsymbol{x} \in \mathcal{X}$, we have*

$$-\lambda^\star(\boldsymbol{x}) \;\in\; \partial P^\star(\epsilon(\boldsymbol{x})),$$

where $\partial P^\star(\epsilon(\boldsymbol{x}))$ denotes the Fréchet subdifferential of $P^\star$ with respect to $\epsilon(\boldsymbol{x})$, and $\lambda^\star(\boldsymbol{x})$ is the optimal dual variable associated to the constraint on $\boldsymbol{x}$.

Proof. See Appendix B. □

For any $\boldsymbol{x_0} \in \mathcal{X}$, let $\delta_{\boldsymbol{x_0}}(\boldsymbol{x})$ be a bump function of radius $r > 0$, centered at $\boldsymbol{x_0}$ (i.e., a continuous, radially-increasing function with support in $\|\boldsymbol{x} - \boldsymbol{x_0}\| \leq r$). Theorem 3.2 implies that

$$P^\star(\epsilon(\boldsymbol{x}) + \delta_{\boldsymbol{x_0}}(\boldsymbol{x})) - P^\star(\epsilon(\boldsymbol{x})) \geq \langle\, -\lambda^\star(\boldsymbol{x_0}),\, \delta_{\boldsymbol{x_0}}(\boldsymbol{x})\,\rangle.$$

The problem (CSL) typically includes an infinite number of constraints. Theorem 3.2 implies that the constraint whose perturbation has the most *potential impact* on the optimal value of (CSL) is the constraint with the highest associated optimal dual variable. For instance, infinitesimally tightening the constraint in a neighbourhood $\boldsymbol{x_0}$ would restrict the feasible set, causing an increase of the optimal value of (CSL) at a rate larger than $\lambda^\star(\boldsymbol{x_0})$. In that sense, the magnitude of the dual variables can be used as a measure of informativeness. Similarly to non-support vectors in SVMs [39], samples associated to inactive constraints (i.e., $\{\boldsymbol{x_0} : \lambda^\star(\boldsymbol{x_0}) = 0\}$), are considered uninformative.

3.3 On the Statistical Bias Induced by Active Sampling

In pool-based active learning, the resulting training set may not be representative of the natural data distribution $\mathfrak{D}$ [40, 41, 42]. This is due to the fact that queried samples are not randomly selected and often lie in the tails of $\mathfrak{D}$. Therefore, when performing several active learning iterations, we undertake a biased version of (BAL):

$$\mathcal{B}^\star = \underset{\mathcal{B}\subseteq\mathcal{U}^{(t)}\,:\,|\mathcal{B}|\leq b}{\arg\min} \; \min_{f\in\mathcal{F}} \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{A}^{(t)}} \left[\ell\left(f(\mathbf{x};\mathcal{L}^{(t)}\cup\mathcal{B}),y\right)\right], \tag{bBAL}$$

where $\mathfrak{A}^{(t)}$ represents the biased distribution underlying the actively sampled set $\mathcal{L}^{(t)}$. Even though, at each iteration, the queried samples have some desired property (e.g., large impact on the expected loss), the learned predictor $f(\mathbf{x};\mathcal{L}^{(t)}\cup\mathcal{B})$ is usually sub-optimal for the natural data distribution.

One way of undertaking the inner minimization in (bBAL) is with the worst-case/feasibility formulation,

$$P^\star = \min_{f\in\mathcal{F}} \quad 0 \tag{bFSL-a}$$

$$\text{s.t.} \quad \ell\left(f(\boldsymbol{x}),y\right) \leq \epsilon(\boldsymbol{x}), \quad \mathfrak{A}^t_{\boldsymbol{x}}\text{–a.e.} \tag{bFSL-b}$$

This formulation is closely related to the constrained formulation in (CSL). In Appendix C, we relate the dual problem of (bFSL) with (D-CSL) to argue that the impact of the inconsistency between distributions on our method is small. The key observation is that, if the marginal distributions $\mathfrak{D}_{\mathbf{x}}$ and $\mathfrak{A}^{(t)}_{\mathbf{x}}$ have the same support, then the respective feasibility formulations are equivalent, regardless of the probability distribution.

4 Proposed Approach

In light of the results mentioned in Section 3.2 on the usefulness of dual variables in constrained statistical learning as a measure of sample informativeness, we present our proposed method, ALLY, in this section. We start by introducing a primal-dual procedure to empirically solve the constrained learning problem, and we will then proceed to describe our active learning algorithm in detail.

4.1 Constrained Empirical Risk Minimization

The formulation in (D-CSL) poses two challenges: (i) the distribution $\mathfrak{D}$ is usually unknown and (ii) it is an infinite-dimensional problem since it optimizes over the functional spaces $\mathcal{F}$ and Λ. We handle the former by replacing expectations by their sample means over a set of labeled samples $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{N}_\mathcal{L}}$, as described in the classical Empirical Risk Minimization (ERM) theory [38, 43]. In order to resolve the latter, we introduce a *parameterization* of the hypothesis class $\mathcal{F}$ as $\mathcal{P} = \{f_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta}\in\Theta\}$, while we create a separate dual variable λ_i for each training sample $\boldsymbol{x}_i$. These modifications lead to the Constrained Empirical Risk Minimization (CERM) problem,

$$\hat{P}^\star = \min_{\boldsymbol{\theta}\in\Theta} \quad \frac{1}{|\mathcal{N}_\mathcal{L}|}\sum_{i\in\mathcal{N}_\mathcal{L}} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}),y_i\right) \tag{CERM-a}$$

$$\text{s.t.} \quad \ell'\left(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}),y_i\right) \leq \epsilon_i, \; \forall i\in\mathcal{N}_\mathcal{L}. \tag{CERM-b}$$

This, in turn, results in the corresponding empirical dual problem,

$$\hat{D}^\star = \max_{\boldsymbol{\lambda}\geq\mathbf{0}} \; \min_{\boldsymbol{\theta}\in\Theta} \hat{L}(\boldsymbol{\theta},\boldsymbol{\lambda}), \tag{D-CERM}$$

where $\boldsymbol{\lambda} = \{\lambda_i\}_{i\in\mathcal{N}_\mathcal{L}}$, $\boldsymbol{\lambda}\geq\mathbf{0}$ represents element-wise non-negativity, ϵ_i denotes the constraint upper bound associated to the ith point-wise constraint, and the *empirical* Lagrangian, $\hat{L}(\boldsymbol{\theta},\boldsymbol{\lambda})$, is defined as

$$\hat{L}(\boldsymbol{\theta},\boldsymbol{\lambda}) = \frac{1}{|\mathcal{N}_\mathcal{L}|}\sum_{i\in\mathcal{N}_\mathcal{L}} \Big[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}),y_i) + \lambda_i\left[\ell'(f_{\boldsymbol{\theta}}(\boldsymbol{x_i}),y_i) - \epsilon_i\right)\Big].$$

The max-min problem (D-CERM) can be undertaken by alternating the minimization with respect to $\boldsymbol{\theta}$ and the maximization with respect to $\boldsymbol{\lambda}$ [44, 30, 32], which leads to the primal-dual constrained

Algorithm 1 Primal-dual constrained learning (PDCL)

Input: Labeled dataset $\mathcal{L}$, primal learning rate η_p, dual learning rate η_d, number of iterations T, number of primal steps per iteration T_p, constraint vector $\boldsymbol{\epsilon}$.
Initialize: $\boldsymbol{\theta}, \boldsymbol{\lambda} \leftarrow \mathbf{0}$.
for $t = 1, \ldots, T$ **do**
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_p \nabla_\theta \hat{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}) \quad (\times T_p)$ // `Update primal variables (`T_p` SGD steps)`
  $s_i \leftarrow \ell'\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right), y_i\right) - \epsilon_i, \ \forall i \in \mathcal{N}_{\mathcal{L}}$. // `Evaluate constraint slacks`
  $\lambda_i \leftarrow \left[\lambda_i + \eta_d s_i\right]_+, \ \forall i \in \mathcal{N}_{\mathcal{L}}$. // `Update dual variables`
end for
Return: $\boldsymbol{\theta}, \boldsymbol{\lambda}$.

Algorithm 2 Active learning via Lagrangian duality (ALLY)

Input: Labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$, primal learning rate η_p, dual learning rate η_d, number of PDCL iterations T, number of primal steps per iteration T_p, constraint vector $\boldsymbol{\epsilon}$.
$\boldsymbol{\theta}^\star, \ \boldsymbol{\lambda}^\star \leftarrow \text{PDCL}(\mathcal{L}, \eta_p, \eta_d, T, T_p, \boldsymbol{\epsilon})$. // `Run the Primal-Dual Algorithm`
$\boldsymbol{\omega}^\star \leftarrow \arg\min_{\boldsymbol{\omega}} \frac{1}{|\mathcal{N}_{\mathcal{L}}|} \sum_{i \in \mathcal{N}_{\mathcal{L}}} \left\| f_{\boldsymbol{\omega}}(f_{\boldsymbol{\phi}^\star}(\boldsymbol{x}_i)) - \lambda_i^\star \right\|^2$. // `Train the dual regression head`
$j^\star \leftarrow \arg\max_{j \in \mathcal{N}_{\mathcal{U}}} f_{\boldsymbol{\omega}^\star}(f_{\boldsymbol{\phi}^\star}(\boldsymbol{x}_j))$ // `Find sample with highest dual variable`
Return: $j^\star$.

learning procedure in Algorithm 1. Notice that $\min_{\boldsymbol{\theta} \in \Theta} \hat{L}(\boldsymbol{\theta}, \boldsymbol{\lambda})$ is the minimum of a family of affine functions on $\boldsymbol{\lambda}$, and thus is concave. Consequently, the outer problem corresponds to the maximization of a concave function and can be solved via gradient ascent. The inner minimization, however, is generally non-convex, but there is empirical evidence that deep neural networks can attain *good* local minima when trained with stochastic gradient descent [45, 46]. Some theoretical remarks on Algorithm 1 can be found in Appendix D.

As shown in Algorithm 1, the dual variables accumulate the slacks (i.e., distances between the per-sample secondary loss and constraint values) over the entire learning procedure. This allows the dual variables to be used as a measure of informativeness, while at the same time affecting the local optimum to which the algorithm converges. Quite interestingly, similar ideas on monitoring the evolution of the loss for specific training samples in order to recognize impactful instances have been used in several generalization analyses [47, 48].

4.2 ALLY: Active Learning via Lagrangian Duality

Our proposed active learning algorithm, ALLY, is presented in Algorithm 2 (for the case of $b = 1$). Given a set of labeled samples, $\mathcal{L}$, we first obtain the model parameters $\boldsymbol{\theta}^\star$ and the dual variables associated to samples in $\mathcal{L}$ using the primal-dual constrained learning approach in Algorithm 1. Taking a representation learning approach [17, 18, 49, 50], we then partition the model $f_{\boldsymbol{\theta}^\star}$ to a *backbone* $f_{\boldsymbol{\phi}^\star} : \mathcal{X} \to \mathbb{R}^d$, where d denotes the dimensionality of the *embedding space*, and a *prediction head* $f_{\boldsymbol{\psi}^\star} : \mathbb{R}^d \to \mathcal{Y}$, such that $f_{\boldsymbol{\theta}^\star} = f_{\boldsymbol{\phi}^\star} \circ f_{\boldsymbol{\psi}^\star}$ and $\boldsymbol{\theta}^\star = \boldsymbol{\phi}^\star \cup \boldsymbol{\psi}^\star$. In order to estimate the informativeness of the samples in the unlabeled dataset $\mathcal{U}$, we train a *dual regression head* $f_{\boldsymbol{\omega}} : \mathbb{R}^d \to \mathbb{R}^+$ on the embeddings generated by $f_{\boldsymbol{\phi}^\star}$ by minimizing the mean-squared error

$$L_{\boldsymbol{\lambda}}(\boldsymbol{\omega}) = \frac{1}{|\mathcal{N}_{\mathcal{L}}|} \sum_{i \in \mathcal{N}_{\mathcal{L}}} \left\| f_{\boldsymbol{\omega}}(f_{\boldsymbol{\phi}^\star}(\boldsymbol{x}_i)) - \lambda_i^\star \right\|^2, \tag{1}$$

while the parameters $\boldsymbol{\phi}^\star$, hence the embeddings, are kept frozen. It should be noted that the idea of mapping embeddings to dual variables is present in other machine learning settings [51]. Once the dual regression head is trained, we evaluate it on the embeddings corresponding to the unlabeled samples, and identify the sample with the highest predicted dual variable.

As explained in Section 2.1, in the batch setting, selecting the b samples with the highest associated dual variables is not optimal, due to the potential information overlap of such samples [7, 16]. Our method is compatible with any batch diversity approach that takes informativeness scores and unlabeled data points (or embeddings) as inputs.

4.3 Connection to BADGE [15]

The BADGE method [15] uses the gradient of the loss function with respect to the parameters of the last layer, denoted by $\boldsymbol{\theta}_L$, as a measure of informativeness, i.e.,

$$\frac{\partial \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \hat{y}(\boldsymbol{x}))}{\partial \boldsymbol{\theta}_L}, \tag{2}$$

where $\hat{y}(\boldsymbol{x})$ is the *hypothetical* label of $\boldsymbol{x}$, defined as $\hat{y}(\boldsymbol{x}) := \arg\max_{y \in \mathcal{Y}} [f_{\boldsymbol{\theta}}(\boldsymbol{x})]_y$. In contrast, as discussed in Theorem 3.2, to evaluate the informativeness of a given sample, ALLY observes

$$\frac{\partial P^\star}{\partial \epsilon(\boldsymbol{x})} = \frac{\partial P^\star}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \epsilon(\boldsymbol{x})} = \frac{\partial \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}} \left[\ell\left(f_{\boldsymbol{\theta}^\star}(\boldsymbol{x}), y\right)\right]}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \epsilon(\boldsymbol{x})}. \tag{3}$$

There are three main differences between these informativeness measures:

- ALLY uses the derivative of the average optimal loss *over the entire distribution*, whereas BADGE only considers the point-wise derivative. In fact, most strategies evaluate their scoring function, or informativeness measure, on a single sample. Note that the term $\frac{\partial \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}[\ell(f_{\boldsymbol{\theta}^\star}(\boldsymbol{x}),y)]}{\partial \boldsymbol{\theta}}$ is constant for all $\mathbf{x}$.
- ALLY observes the gradient with respect to *all* model parameters, not only the ones in the last layer.
- Aside from the derivative of the loss with respect to the model parameters, ALLY also considers an additional term $\frac{\partial \boldsymbol{\theta}}{\partial \epsilon(\boldsymbol{x})}$, which models how the results of the optimization (i.e., model parameters) change when the constraint function is perturbed.

4.4 Interpreting the Informativeness Score of ALLY

Our proposed framework allows us to leverage the trained backbone and dual regression head in a *generative* manner to create novel samples that are most informative. Here, we focus on the MNIST dataset, and use the trained model to generate synthetic images with maximal associated dual variables. We begin by training a MLP on 10% of the MNIST dataset. Then, similarly to [52], we perform gradient ascent on images that are initially considered uninformative, so as to maximize their predicted dual variables. The progression of the resulting images is shown in Figure 1.

As the predicted dual variable increases, patterns corresponding to other digits appear, increasing the uncertainty on the true label of the image. For instance, the third column of Figure 1 shows a handwritten '1' that is progressively transformed into a blurred superposition of '7,' '2' and '1'. Images in the last row of Figure 1 can be interpreted as lying in the tails of the distribution $\mathfrak{D}$, or close to the decision boundary of the end-to-end model $f_{\boldsymbol{\theta}}$. This also suggests that informative samples and

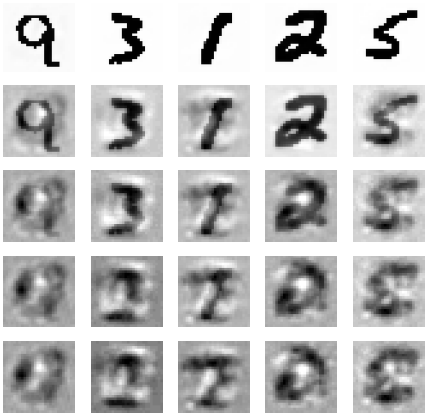

Figure 1: Sample generation by maximization of predicted dual variables. The top row shows the initial images from MNIST, to which the predictor associates a low dual variable. Rows 2-5 display the images resulting from subsequent iterations of gradient ascent.

outliers (such as mislabeled samples) may be hard to distinguish. Recent empirical findings indicate that many active learning algorithms consistently prefer to acquire samples that traditional models fail to learn [53]. Thus, modifying ALLY in order to avoid sampling these so-called *collective outliers* (e.g., by setting an upper bound on the dual variable associated to the queries) would be desirable in datasets that are not highly curated.

5 Experimental Evaluation

5.1 Settings

We consider four image classification tasks and one biomedical, non-image regression task. In the classification setting, we use standard datasets that commonly appear in the active learning literature, namely STL-10 [54], CIFAR-10 [55], SVHN [56] and MNIST [57]. Lacking an established benchmark regression dataset for active learning, we evaluate ALLY on the Parkinsons Telemonitoring dataset (PTD) [58]. In this regression task, the goal is to predict UPDRS (Unified Parkinson's Disease Rating Scale) scores from dysphonia measurements such as variation in fundamental frequency. Since measurements in this dataset are the result of a *costly* clinical trial that requires expert knowledge, this task is a prime example in which active learning might be essential.

In all experiments, the initial labeled set $\mathcal{L}_0$ consists of 200 randomly drawn samples, and the budget is set to either $b = 200$ or $b = 1000$. We use a ResNet-18 architecture [59] with an embedding size of 128. In the case of MNIST and PTD, which are simpler tasks, we use a multi-layer perceptron (MLP) with two hidden layers, each with 256 neurons and rectified linear unit (ReLU) activation, leading to an embedding size of 256. The dual regression head f_ω is a MLP with 5 hidden layers, ReLU activations and batch normalization.

As done in [24, 25], to ensure diversity in the batch, we cluster the embeddings of the unlabeled samples, i.e., $\{f_{\boldsymbol{\phi}^\star}(\boldsymbol{x}_j)\}_{j \in \mathcal{N}_\mathcal{U}}$, using the k-MEANS clustering algorithm [60], where $k \leq b$ is a hyperparameter. We then select the samples with the highest associated dual variables from each cluster, while maintaining equity among the number of samples per cluster. As shown in Appendix F, the performance of ALLY improves with an increased number of clusters k, since it leads to a more diverse batch of selected samples. Therefore, in all our experiments, we set $k = b$.

Regarding the role of the secondary loss, we opted for a generic formulation as the performed sensitivity analysis in Theorem 3.2 holds for various choices of ℓ'. However, in all our experiments, we set $\ell'(\cdot, \cdot) = \ell(\cdot, \cdot)$. We believe that using unsupervised or self-supervised losses for ℓ' is a promising research direction, and we leave it for future work.

We compare our algorithm with Entropy sampling, BADGE, Coreset and BAIT. While Entropy sampling focuses purely on uncertainty, BADGE and BAIT balance both diversity and informativeness with different approaches. Coreset, also considered a state-of-the-art method, differs from the previous methods in that it focuses on batch representativeness by framing active learning as a coreset selection

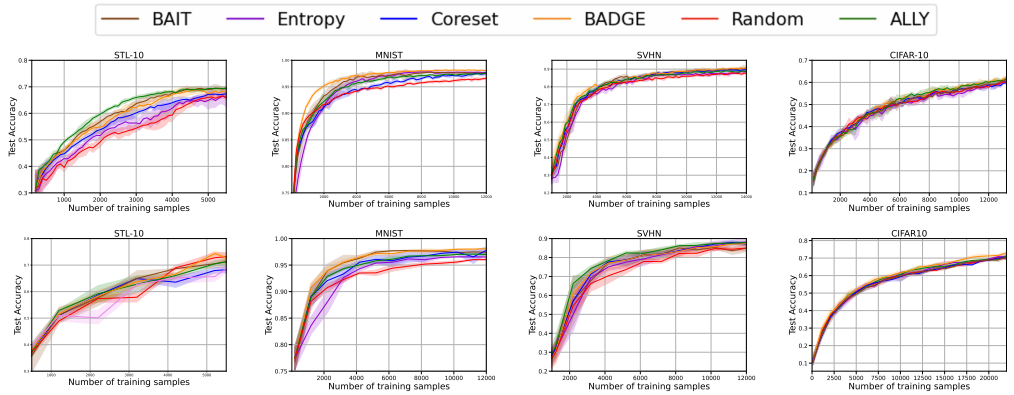

Figure 2: Accuracy in the test set as a function of the number of training samples in four classification settings and two budgets, 200 (top) and 1000 (bottom). Solid curves represent the mean across five different random seeds, while shaded regions correspond to the standard deviation.

problem. It has been observed that, in some scenarios, several active learning methods fail to consistently outperform Random Sampling [53, 61, 29]. We thus include it as one of the five baselines. As explained in section 4, ALLY uses a primal-dual approach to learn $f(\mathbf{x}; \mathcal{L} \cup \mathcal{U}))$, while all other baselines are optimized using stochastic gradient descent (ADAM). We adopt the PyTorch [62] implementation of the baselines from [63].

5.2 Classification

The experiments on STL-10, CIFAR-10, SVHN and MNIST are all 10-class, image classification tasks. We use the cross-entropy loss for both $\ell(\cdot, \cdot)$ and $\ell'(\cdot, \cdot)$ and set $\epsilon(\boldsymbol{x}) = 0.2$, $\forall \boldsymbol{x}$. As shown in Figure 2, ALLY outperforms other baselines in STL-10 (budget 200), SVHN (budget 1000) and CIFAR-10 (budget 200). In these three datasets, the improvement in the number of samples needed by ALLY to achieve 97% of the final accuracy, in comparison with the best baseline, is 9%, 8% and 2%, respectively. In MNIST, however, BADGE and BAIT consistently outperform ALLY. This may be due to the fact that the MLP, being less expressive, yields embeddings of lower quality, hindering the prediction of dual variables.

5.3 Regression

We use mean-squared error for both $\ell(\cdot, y)$ and $\ell'(\cdot, y)$ and set $\epsilon(\boldsymbol{x}) = 0.1$, $\forall \boldsymbol{x}$. As seen in Figure 3, ALLY outperforms both Random and Coreset in this regression task, the gap being larger at the beginning of the learning curve. Note that BADGE and Entropy are not applicable, since they are limited to classification scenarios.

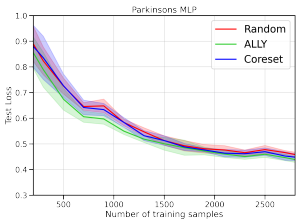

Figure 3: Mean-squared error in the test set as a function of the number of training samples in the Parkinson's telemonitoring dataset.

5.4 Ablation on the Constraint Tightness

Figure 4 illustrates the role ϵ plays in the optimization procedure. For very large values of epsilon, the constraint slacks become negative for all samples, and thus all dual variables become zero, making them uninformative (analogous to an unconstrained problem) and the method performs similarly to random sampling. Our ablations suggest that values in the range $[1.05p_u\ ,\ 1.25p_u]$, where p_u is the average loss observed when training the model without constraints, work well in practice. The sensitivity of the method with respect to the constraint level is impacted both by the task/dataset at hand and the diversity technique used. It is reasonable to expect K-means to be less sensitive to the constraint level than stochastic perturbations of the informativeness scores.

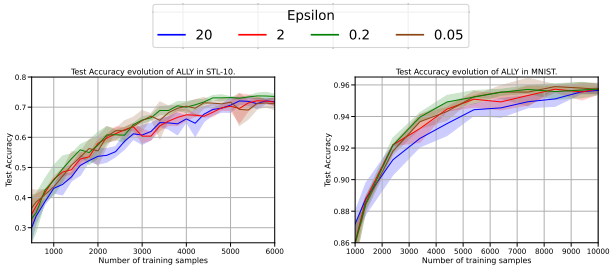

Figure 4: Ablation on the value of the constraint tightness ϵ on the performance of ALLY in STL-10 and MNIST.

5.5 Dataset Redundancy

We study the performance of ALLY and Entropy Sampling with varying levels of dataset redundancy in STL-10. We define the level of dataset redundancy as the number of copies of each sample present in the training set. We then analyze the evolution of test accuracy over several rounds of active learning with a query budget of 200.

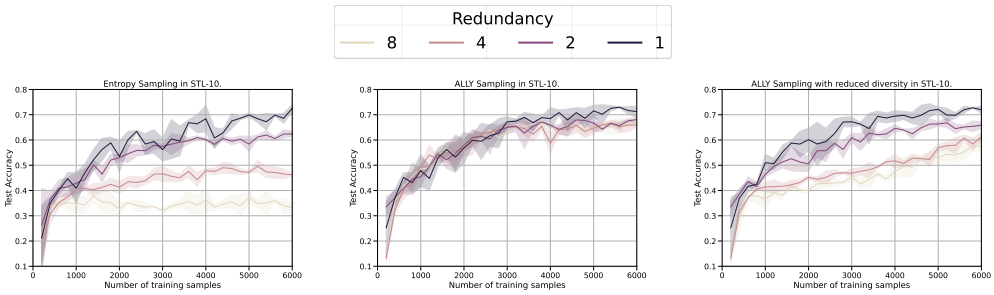

Figure 5: Impact of dataset redundancy on the performance of Entropy Sampling (left), ALLY (center) and low-diversity ALLY (right).

We observe that a strategy solely based on informativeness (e.g., Entropy Sampling) is very sensitive to this type of dataset redundancy. This can be attributed to the fact that, at each round, copies of the most informative samples are queried simultaneously, leading to low batch diversity. Moreover, ALLY appears to be more robust than Entropy Sampling to this type of dataset corruption. This is not surprising, since only one sample is queried from each cluster, avoiding batch information overlap. However, when increasing the number of samples queried from each batch, the performance of ALLY is degraded, resembling that of Entropy Sampling. This experiment suggests that the impact of dataset redundancy on active learning strategies is more linked to the diversity technique used than the informativeness measure. Whether this behaviour generalizes to other types of redundancy is left for future work.

6 Concluding Remarks

We presented ALLY, a principled batch active learning method based on Lagrangian duality. Our method formulates the learning problem using constrained optimization and solves it in the Lagrangian dual domain via a primal-dual approach. We leverage the fact that the magnitude of the dual variables can be viewed as a measure of informativeness of the corresponding training sample, as it indicates the sensibility of the optimum value of the objective function with respect to a perturbation in the respective constraint.

Following the completion of the primal-dual learning phase, we employ the learned sample representations, as well as their respective dual variables, to train a dual regression head. This predictor is used to estimate the dual variables associated to unlabeled samples. The resulting informativeness measure is compatible with several batch diversity promoting techniques. Using the k-MEANS algorithm, we demonstrated that this principled method exhibits competitive performance in several classification and regression experiments. We also showed that under certain conditions, the impact of the distribution shift induced by active sampling is small.

In our experiments, we have set the secondary loss to be identical to the primary supervised loss (i.e., cross-entropy loss for classification, and mean-squared error for regression). However, evaluating the performance of ALLY under alternative *unsupervised* or *self-supervised* secondary losses is a promising future direction. Finally, it would be interesting to evaluate the performance of ALLY under different diversity measures, comparing their computational burden.

7 Acknowldegments

Supported by NSF-Simons MoDL, Award 2031985, NSF AI Institutes program, Award 2112665, and NSF HDR TRipods Award 1934960.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[2] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=qrwe7XHTmYb`.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

[4] Ying Liu. Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of chemical information and computer sciences*, 44 6:1936–41, 2004.

[5] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 417–424, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143897. URL `https://doi.org/10.1145/1143844.1143897`.

[6] V. Nath, Dong Yang, Bennett A. Landman, Daguang Xu, and Holger R. Roth. Diminishing uncertainty within the training pool: Active learning for medical image segmentation. *IEEE Transactions on Medical Imaging*, 40:2534–2547, 2021.

[7] Burr Settles. Active learning literature survey. 2009.

[8] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach, 2018.

[9] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. *CoRR*, abs/2107.14263, 2021. URL `https://arxiv.org/abs/2107.14263`.

[10] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. *CoRR*, abs/cmp-lg/9407020, 1994. URL `http://arxiv.org/abs/cmp-lg/9407020`.

[11] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S. Shankar Sasrty. A convex optimization framework for active learning. In *2013 IEEE International Conference on Computer Vision*, pages 209–216, 2013. doi: 10.1109/ICCV.2013.33.

[12] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper/2007/file/a1519de5b5d44b31a01de013b9b51a80-Paper.pdf`.

[13] Wenbin Cai, Ya Zhang, and Jun Zhou. Maximizing expected model change for active learning in regression. In *2013 IEEE 13th International Conference on Data Mining*, pages 51–60, 2013. doi: 10.1109/ICDM.2013.104.

[14] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 441–448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

[15] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *CoRR*, abs/1906.03671, 2019. URL `http://arxiv.org/abs/1906.03671`.

[16] K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2000)*, 2003.

[17] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[19] Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham M. Kakade. Gone fishing: Neural active learning with fisher embeddings. *CoRR*, abs/2106.09675, 2021. URL `https://arxiv.org/abs/2106.09675`.

[20] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *CoRR*, abs/1703.02910, 2017. URL `http://arxiv.org/abs/1703.02910`.

[21] Donggeun Yoo and In So Kweon. Learning loss for active learning. *CoRR*, abs/1905.03677, 2019. URL `http://arxiv.org/abs/1905.03677`.

[22] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. *CoRR*, abs/1904.00370, 2019. URL `http://arxiv.org/abs/1904.00370`.

[23] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. *CoRR*, abs/1911.09162, 2019. URL `http://arxiv.org/abs/1911.09162`.

[24] Fedor Zhdanov. Diverse mini-batch active learning. *CoRR*, abs/1901.05954, 2019. URL `http://arxiv.org/abs/1901.05954`.

[25] Zalán Bodó, Zsolt Minier, and Lehel Csató. Active learning with clustering. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, editors, *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pages 127–139, Sardinia, Italy, 16 May 2011. PMLR. URL `https://proceedings.mlr.press/v16/bodo11a.html`.

[26] Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B. Chan. A comparative survey of deep active learning, 2022. URL `https://arxiv.org/abs/2203.13450`.

[27] Javad Azimi, Alan Fern, Xiaoli Zhang-Fern, Glencora Borradaile, and Brent Heeringa. Batch active learning via coordinated matching, 2012. URL `https://arxiv.org/abs/1206.6458`.

[28] Andreas Kirsch, Sebastian Farquhar, and Yarin Gal. A simple baseline for batch active learning with stochastic acquisition functions. *CoRR*, abs/2106.12059, 2021. URL `https://arxiv.org/abs/2106.12059`.

[29] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *CoRR*, abs/1907.06347, 2019. URL `http://arxiv.org/abs/1907.06347`.

[30] Luiz F. O. Chamon and Alejandro Ribeiro. Probably approximately correct constrained learning, 2021.

[31] Luiz F. O. Chamon, Santiago Paternain, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained learning with non-convex losses. *CoRR*, abs/2103.05134, 2021. URL `https://arxiv.org/abs/2103.05134`.

[32] Ferdinando Fioretto, Pascal Van Hentenryck, Terrence W. K. Mak, Cuong Tran, Federico Baldo, and Michele Lombardi. Lagrangian duality for constrained deep learning. In *ECML/PKDD (5)*, pages 118–135, 2020. URL `https://doi.org/10.1007/978-3-030-67670-4_8`.

[33] Juan Cerviño, Luana Ruiz, and Alejandro Ribeiro. Training stable graph neural networks through constrained learning. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4223–4227, 2022. doi: 10.1109/ICASSP43922.2022.9746912.

[34] Zebang Shen, Juan Cervino, Hamed Hassani, and Alejandro Ribeiro. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=Xo0lbDt975`.

[35] Raghu Arghal, Eric Lei, and Shirin Saeedi Bidokhti. Robust graph neural networks via probabilistic lipschitz constraints. *CoRR*, abs/2112.07575, 2021. URL https://arxiv.org/abs/2112.07575.

[36] Alexander Robey, Luiz Chamon, George J. Pappas, Hamed Hassani, and Alejandro Ribeiro. Adversarial robustness with semi-infinite constrained learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6198–6215. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/312ecfdfa8b239e076b114498ce21905-Paper.pdf.

[37] Ignacio Hounie, Luiz F. O. Chamon, and Alejandro Ribeiro. Automatic data augmentation via invariance-constrained learning. URL https://arxiv.org/abs/2209.15031.

[38] V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999. doi: 10.1109/72.788640.

[39] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

[40] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. On statistical bias in active learning: How and when to fix it, 2021. URL https://arxiv.org/abs/2101.11665.

[41] Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781, 2011. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2010.12.054. URL https://www.sciencedirect.com/science/article/pii/S0304397510007620. Algorithmic Learning Theory (ALT 2009).

[42] Chelsea Murray, James Urquhart Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Addressing bias in active learning with depth uncertainty networks... or not. *CoRR*, abs/2112.06926, 2021. URL https://arxiv.org/abs/2112.06926.

[43] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014. ISBN 1107057132.

[44] L. Hurwicz K. J. Arrow and H. Uzawa. Studies in linear and non-linear programming, by k. j. arrow, l. hurwicz and h. uzawa. stanford university press, 1958. 229 pages. *Canadian Mathematical Bulletin*, 3(3):196–198, 1960. doi: 10.1017/S0008439500025522.

[45] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. URL http://arxiv.org/abs/1611.03530.

[46] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[47] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

[48] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. *CoRR*, abs/1803.00942, 2018. URL http://arxiv.org/abs/1803.00942.

[49] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[50] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[51] Harikrishna Narasimhan, Andrew Cotter, Yichen Zhou, Serena Wang, and Wenshuo Guo. Approximate heavily-constrained learning with lagrange multiplier models. In *Advances in Neural Information Processing Systems*, volume 33, pages 8693–8703, 2020.

[52] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

[53] Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei 0001, and Christopher D. Manning. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In Chengqing Zong, Fei Xia, Wenjie Li 0002, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7265–7281. Association for Computational Linguistics, 2021. ISBN 978-1-954085-52-7. URL https://aclanthology.org/2021.acl-long.564.

[54] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

[55] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[56] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[57] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[58] Athanasios Tsanas, Max A. Little, Patrick E. McSharry, and Lorraine O. Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, 2010. doi: 10.1109/TBME.2009.2036000.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[60] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28 (2):129–137, 1982.

[61] Yonatan Geifman and Ran El-Yaniv. Deep active learning over the long tail. *CoRR*, abs/1711.00941, 2017. URL http://arxiv.org/abs/1711.00941.

[62] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[63] Yao-Yuan Yang, Shao-Chuan Lee, Yu-An Chung, Tung-En Wu, Si-An Chen, and Hsuan-Tien Lin. libact: Pool-based active learning in python. Technical report, National Taiwan University, October 2017. URL https://github.com/ntucllab/libact. available as arXiv preprint https://arxiv.org/abs/1710.00379.

[64] Adam M. Oberman and Jeff Calder. Lipschitz regularized deep neural networks converge and generalize. *CoRR*, abs/1808.09540, 2018. URL http://arxiv.org/abs/1808.09540.

[65] Dimitri Bertsekas, Angelia Nedic, and Asuman Ozdaglar. *Convex analysis and optimization*, volume 1. Athena Scientific, 2003.

[66] Alexander Shapiro. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, 58(2):133–161, 2009. doi: 10.1080/02331930902730070. URL https://doi.org/10.1080/02331930902730070.

[67] J.Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013. URL https://web.archive.org/web/20170809131322id_/http://www2.isye.gatech.edu/~ashapiro/publications/book-Jan00_typeset.pdf.

[68] Alexander Shapiro. Directional differentiability of the optimal value function in convex semi-infinite programming. *Math. Program.*, 70(1–3):149–157, oct 1995. ISSN 0025-5610. doi: 10.1007/BF01585933. URL https://doi.org/10.1007/BF01585933.

[69] R. Hettich P. Zencke. Directional derivatives for the value-function in semi-infinite programming. 38, 1987.

[70] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

[71] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[72] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Strong Duality of the CSL Problem

Assumption 1. The losses $\ell(\cdot, y)$ and $\ell'(\cdot, y)$, are convex functions for all $y \in \mathcal{Y}$.

In Assumption 1, the convexity of the losses is taken with respect to the model's output, and not the model parameters. The cross-entropy loss, commonly used in classification with a softmax layer satisfies strong convexity when considering a probability simplex [64]. Typical losses for regression (e.g., mean-squared error, L1 loss) also satisfy this assumption.

Assumption 2. The hypothesis class $\mathcal{F}$ is convex.

To obtain a convex hypothesis class, as required by Assumption 2, it suffices to take the convex hull of the function class originally considered.

Assumption 3. There exists $f \in \mathcal{F}$ strictly feasible for (CSL) (i.e., $\ell'\left(f(\boldsymbol{x}), y\right) < \epsilon(\boldsymbol{x}), \quad \mathfrak{D}$–a.e.)

Assumption 3 guarantees that the problem (CSL) is feasible and that its dual is well-posed.

Proposition A.1. *Under Assumptions 1-3,* (CSL) *and* (D-CSL) *are strongly dual, i.e.,* $P^* = D^*$.

Proof. Note that assumptions 1 and 2 imply that (CSL) is a *convex* program. Under the strict feasibility assumption, (CSL) satisfies the constraint qualification known as *Slater's condition*, from which strongly duality follows [65, 66]. □

B Proof of Theorem 3.2 (Sensitivity of $P^\star$)

This result stems from a sensitivity analysis on the constraint of problem (CSL) and is well-known in the convex optimization literature. More general versions of this theorem are shown in [67] (Section 4), [68] or [69].

We start by viewing (CSL) as an optimization problem parameterized by the function $\epsilon(\boldsymbol{x})$:

$$\begin{aligned} P^\star(\epsilon(\boldsymbol{x})) = \min_{f \in \mathcal{F}} \quad & \mathbb{E}_{\mathfrak{D}}\left[\ell\left(f(\boldsymbol{x}), y\right)\right] \\ \text{s.t.} \quad & \ell'\left(f(\boldsymbol{x}), y\right) \leq \epsilon(\boldsymbol{x}), \quad \mathfrak{D}_{\mathbf{x}}\text{–a.e.} \end{aligned}$$

Define the Lagrangian $L(f, \lambda(\boldsymbol{x}); \epsilon(\boldsymbol{x}))$ as

$$L(f, \lambda(\boldsymbol{x}); \epsilon(\boldsymbol{x})) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell(f(\boldsymbol{x}), y) + \lambda(\boldsymbol{x})(\ell'(f(\boldsymbol{x}), y) - \epsilon(\boldsymbol{x}))\right],$$

where the dependence on $\epsilon(\boldsymbol{x})$ is explicitly shown. Then, following the definition of $P^\star(\epsilon(\boldsymbol{x}))$ and using strong duality, we have

$$P^\star(\epsilon(\boldsymbol{x})) = \min_f L(f, \lambda^\star(\boldsymbol{x}; \epsilon(\boldsymbol{x})); \epsilon(\boldsymbol{x})) \leq L(f, \lambda^\star(\boldsymbol{x}; \epsilon(\boldsymbol{x})); \epsilon(\boldsymbol{x}))$$

with the inequality being true for any function f, and where the dependence of $\lambda^\star$ on $\epsilon(\boldsymbol{x})$ is also explicitly shown. Now, consider an arbitrary function $\epsilon'(\boldsymbol{x})$ and the respective primal function $f^\star(\cdot; \epsilon'(\boldsymbol{x}))$ which minimizes its corresponding Lagrangian. Plugging $f^\star(\cdot; \epsilon'(\boldsymbol{x}))$ into the above inequality, we have

$$\begin{aligned} P^\star(\epsilon(\boldsymbol{x})) &\leq L(f^\star(\cdot; \epsilon'(\boldsymbol{x})), \lambda^\star(\boldsymbol{x}; \epsilon(\boldsymbol{x})); \epsilon(\boldsymbol{x})) \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell(f^\star(\boldsymbol{x}; \epsilon'(\boldsymbol{x})), y) + \lambda^\star(\boldsymbol{x}; \epsilon(\boldsymbol{x}))(\ell'(f^\star(\boldsymbol{x}; \epsilon'(\boldsymbol{x})), y) - \epsilon(\boldsymbol{x}))\right] \end{aligned}$$

Now, since $f^\star(\cdot; \epsilon'(\boldsymbol{x}))$ is *optimal* for constraint bounds given by $\epsilon'(\boldsymbol{x})$ and complementary slackness holds, we have $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell(f^\star(\boldsymbol{x}; \epsilon'(\boldsymbol{x})), y)\right] = P^\star(\epsilon'(\boldsymbol{x}))$. Moreover, since $f^\star(\cdot; \epsilon'(\boldsymbol{x}))$ is feasible for constraint bounds given by $\epsilon'(\boldsymbol{x})$, we have $\ell'(f^\star(\boldsymbol{x}; \epsilon'(\boldsymbol{x})), y) \leq \epsilon'(\boldsymbol{x}),\ \mathfrak{D}_{\mathbf{x}}$-a.e. Combining the above, we get

$$\begin{aligned} P^\star(\epsilon(\boldsymbol{x})) &\leq P^\star(\epsilon'(\boldsymbol{x})) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\lambda^\star(\boldsymbol{x}; \epsilon(\boldsymbol{x}))(\epsilon'(\boldsymbol{x}) - \epsilon(\boldsymbol{x}))\right] \\ &= P^\star(\epsilon'(\boldsymbol{x})) + \langle \lambda^\star(\boldsymbol{x}; \epsilon(\boldsymbol{x}))(\epsilon'(\boldsymbol{x}) - \epsilon(\boldsymbol{x}))\rangle, \end{aligned}$$

or eqivalently,

$$P^\star(\epsilon'(\boldsymbol{x})) - P^\star(\epsilon(\boldsymbol{x})) \geq \langle -\lambda^\star(\boldsymbol{x};\epsilon(\boldsymbol{x}))(\epsilon'(\boldsymbol{x}) - \epsilon(\boldsymbol{x}))\rangle,$$

which exactly matches the definition of the Fréchet subdifferential in Definition 3.1, hence completing the proof.

This shows that $-\lambda^\star(\boldsymbol{x})$ is a sub-gradient of $P^*(\epsilon(\boldsymbol{x}))$. Let ∂P^* be the sub-differential of P^* (i.e., the set of all sub-gradients). By definition, P^* is differentiable if its sub-differential is a singleton: $\partial P^* = \{-\lambda^\star(\boldsymbol{x})\}$. This holds in our problem if the Lagrangian minimizers are unique, which is the case for a strongly convex objective. However, it does not hold in the general case. In [68], right-side differentiability is shown by further assuming that $\exists \alpha > P^*$ and a compact set S such that set of feasible points where the objective does not exceed α is contained in S.

C The Statistical Bias Induced by Active Sampling

As mentioned in Section 3.3, when performing several active learning iterations, we undertake a biased version of (BAL):

$$\mathcal{B}^* = \underset{\mathcal{B}\subseteq\mathcal{U}_t\,:\,|\mathcal{B}|\leq b}{\arg\min}\ \min_{f\in\mathcal{F}} \mathbb{E}_{(\mathbf{x},y)\sim\mathfrak{A}^{(t)}}\left[\ell\left(f(\mathbf{x};\mathcal{L}_t\cup\mathcal{B}),y\right)\right], \tag{b-BAL}$$

where $\mathfrak{A}^{(t)}$ represents the biased/shifted distribution underlying the actively sampled set $\mathcal{L}^{(t)}$.. This makes the learnt predictor $f(\mathbf{x};\mathcal{L}^{(t)}\cup\mathcal{B})$ sub-optimal for the natural data distribution $\mathfrak{D}$.

We focus our attention on the setting with $\ell = \ell'$. Observe that if $\mathfrak{D}_{\mathbf{x}}$ and $\mathfrak{A}_{\mathbf{x}}$ have the same support, then the feasibility formulation of the inner minimization is the same whether we consider the natural or biased data distribution. That is,

$$\begin{aligned} P^\star = \min_{f\in\mathcal{F}} \quad & 0 \\ \text{s.t.} \quad & \ell\left(f(\boldsymbol{x}),y\right) \leq \epsilon(\boldsymbol{x}), \quad \mathfrak{D}_{\boldsymbol{x}}\text{–a.e.} \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} P^\star = \min_{f\in\mathcal{F}} \quad & 0 \\ \text{s.t.} \quad & \ell\left(f(\boldsymbol{x}),y\right) \leq \epsilon(\boldsymbol{x}), \quad \mathfrak{A}_{\boldsymbol{x}}\text{–a.e.} \end{aligned}$$

This is because,

$$\begin{aligned} \ell(f(\mathbf{x}),y) \leq \epsilon(\mathbf{x}) \quad \mathfrak{D}_{\boldsymbol{x}}\text{–a.e.} \quad & \Longleftrightarrow \quad \ell(f(\mathbf{x}),y)p_{\mathfrak{D}_{\boldsymbol{x}}}(\mathbf{x}) \leq \epsilon(\mathbf{x})p_{\mathfrak{D}_{\boldsymbol{x}}}(\mathbf{x}) \quad \forall\mathbf{x} \\ \ell(f(\mathbf{x}),y) \leq \epsilon(\mathbf{x}) \quad \mathfrak{A}_{\boldsymbol{x}}\text{–a.e.} \quad & \Longleftrightarrow \quad \ell(f(\mathbf{x}),y)p_{\mathfrak{D}'_{\boldsymbol{x}}}(\mathbf{x}) \leq \epsilon(\mathbf{x})p_{\mathfrak{A}_{\boldsymbol{x}}}(\mathbf{x}) \quad \forall\mathbf{x} \end{aligned}$$

Notice that for both distributions, the constraints are *only* enforced when the respective density is non-zero. When the density is exactly zero, the constraint is satisfied trivially. Therefore, if

$$\{x\,:\,p_{\mathfrak{D}_{\boldsymbol{x}}}(\mathbf{x}) > 0\} = \{x\,:\,p_{\mathfrak{A}_{\boldsymbol{x}}}(\mathbf{x}) > 0\}$$

then the feasible set is the same for both problems. Since the objectives also coincide, the problems are identical.
In fact, we can replace the objective in the above problems by any functional $R(f)$ that does not depend on the data distribution (e.g: regularizers or complexity measures such as $\|f\|$) and the equivalence still holds. When setting a regularizing functional as the objective, the resulting dual problem is:

$$D^\star = \max_{\lambda\in\Lambda}\ \min_{f\in\mathcal{F}} R(f) + \lambda(\boldsymbol{x})(\ell(f(\boldsymbol{x}),y) - \epsilon(\boldsymbol{x})) \tag{D-FSL}$$

We know that the primal problem of (D-FSL) is agnostic to the underlying data distribution. If we further assume the existence of a striclty feasible solution, strong duality holds (see Appendix A). Thus, the solution of (D-FSL) is also unimpacted by the distribution shift induced by active sampling.

Observe that the dual problem in the original ALLY formulation, with $\ell = \ell'$, is:

$$D^\star = \max_{\lambda\in\Lambda}\ \min_{f\in\mathcal{F}} (\lambda(\boldsymbol{x}) + \mathbf{1})\ell(f(\boldsymbol{x}),y) - \lambda(\boldsymbol{x})\epsilon(\boldsymbol{x}) \tag{D-CSL}$$

Note that the dual problems (D-CSL) and (D-FSL) differ in two aspects: the presence of a regularizer $R(f)$ and a unit shift in the dual variable function $\lambda(\mathbf{x})$. However, a unit shift in the dual variable

function does not affect the relative impact of each sample on the expected loss. That is, the ranking of informativeness scores remains unchanged. Thus, assuming the supports of $\mathfrak{D}_{\mathbf{x}}$ and $\mathfrak{A}_{\mathbf{x}}$ coincide, ALLY is equivalent to a statistically consistent active learning method modulus the use of a dual function regularizer.

D Empirical Primal-Dual Learning Procedure in Algorithm 1

Even in the case were (CSL) and (D-CSL) are strongly dual, in practice we undertake the empirical version of (D-CSL) due to the challenges mentioned in Section 4.1 (i.e: $\mathfrak{D}$ is unknown and $\mathcal{F}$ is infinite dimensional). This requires introducing a *parameterization* of the hypothesis class $\mathcal{F}$ as $\mathcal{P} = \{f_{\boldsymbol{\theta}} \,|\, \boldsymbol{\theta} \in \Theta\}$ and replacing expectations by sample means. In the following section, we present a high-level overview of some results from [30] on the implications of these two changes and the solution yielded by Algorithm PDCL.

D.1 Approximation Error

On one hand, approximating the function class $\mathcal{F}$ with a finite dimensional parametrization $\mathcal{P}$ transforms the dual problem (D-CSL) into:

$$\tilde{D}^{\star} = \max_{\lambda \in \Lambda} \min_{\theta \in \Theta} \tilde{L}\left(f_{\theta}, \lambda(\boldsymbol{x})\right), \tag{$\tilde{D}$-CSL}$$

where

$$\tilde{L}\left(f_{\theta}, \lambda(\boldsymbol{x})\right) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathfrak{D}}\left[\ell(f_{\theta}(\boldsymbol{x}), y) + \lambda(\boldsymbol{x})(\ell'(f_{\theta}(\boldsymbol{x}), y) - \epsilon(\boldsymbol{x}))\right].$$

Assuming that $\mathcal{P}$ is PAC learnable and that there is $\nu > 0$ such that for each $f \in \mathcal{F}$ there exists $f_{\theta} \in P$ that satisifes $\sup_{\boldsymbol{x}\in\mathcal{X}} |f_{\theta}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \nu$, $f_{\theta^{\star}}$ is a *near optimal solution* of (CSL) [30] (Proposition 2).

Note that the assumption mentioned above is connected to the richness of the parameterization $\mathcal{P}$. For instance, $\mathcal{F}$ can denote C^0 (i.e: space of continuous functions) and $\mathcal{P}$ be a neural network, which meets the universal approximation assumption [70].

D.2 Estimation Error

On the other hand, approximating expectations by their sample means transforms the statistical dual problem ($\tilde{D}$-CSL) into its empirical counterpart (D-CERM):

$$\hat{D}^{\star} = \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\boldsymbol{\theta} \in \Theta} \hat{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}), \tag{D-CERM}$$

This modification creates a difference between the optimal value of the parametrized dual problem and the optimal value of the empirical dual problem. However, assuming that the losses in question are $[0, B]$-valued and M-Lipschitz continuous -conditions which can be satisfied in the case of Cross-Entropy and Mean-Squared Error by setting bounds- this difference is bounded by a constant. In addition, this constant depends on the number of samples, the VC dimension of the parametrization $\mathcal{P}$, B and M. See [30] (Proposition 3) for a more detailed analysis.

In [30] (Theorem 3), the approximation and estimation errors are combined, and the sub-optimality of Algorithm PDCL with respect to (CSL) is bounded.

E Additional Experimental Settings

We do not use warm-starting or data augmentation. We employ early stopping on a validation set from the unlabelled pool and run each experiment five times with different random seeds. The patience in the early stopping callback was set to 2 epochs. We report the mean and standard deviations across the different seeds. We smooth the curves using exponential moving average for clarity and focus on discriminative regions of the learning curves. The dropout parameters for the dual prediction head were determined by cross-validation and are set to 0.3 and 0.25. The parameters of the neural networks are updated once at each iteration ($T_p = 1$) using the ADAM optimizer [71], and a learning

rate of $\eta_p = 0.001$ for SVHN, STL and CIFAR and $\eta_p = 0.005$ on MNIST. We adapt the architecture for STL-10 images ($96 \times 96 \times 3$) and split the dataset so as to have 11000 train samples and 2000 test samples (the original version has 5000 training samples and 8000 testing samples). The dual variables are updated with stochastic gradient ascent and a learning rate of $\eta_d = 0.05$. During primal dual steps we use a diminishing update rule for the step size. The number of layers of the dual variable prediction head was set by cross validation. All experiments were carried out on a system with the following specifications: Ubuntu 20.04, AMD Threadripper 3960X CPU and RTX 3090 GPU.

F Ablation on the Number of Clusters

We perform an ablation study on the number of clusters used by the k-MEANS algorithm. We use the MNIST dataset and the MLP architecture described in section 5. As shown in Figure 6, the performance of ALLY improves as the number of clusters grows. This suggests that, in this setting, prioritizing diversity over individual sample informativeness is beneficial in terms of average test accuracy.

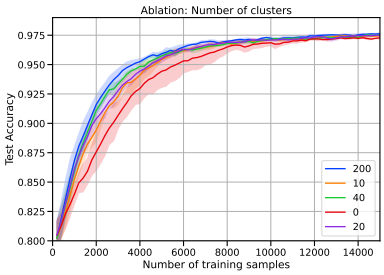

Figure 6: Ablation study on the number of clusters k used in ALLY.

G Experiment on Tiny ImageNet

Figure 7 compares the performance of ALLY with baselines methods on the Tiny ImageNet dataset [72], which is a subset of the ImageNet dataset (ILSVRC2012) consisting of $64 \times 64 \times 3$ images categorized in 200 classes.

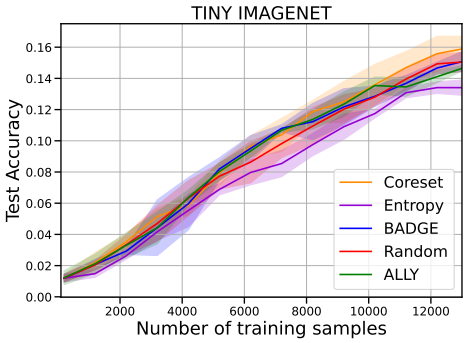

Figure 7: Empirical evaluation on the Tiny ImageNet dataset with $b = 1000$.