# SoCal: Selective Oracle Questioning for Consistency-based Active Learning of Physiological Signals

**Anonymous authors**
Paper under double-blind review

## Abstract

The ubiquity and rate of collection of physiological signals produce large, unlabelled datasets. Active learning (AL) can exploit such datasets by incorporating human annotators (oracles) to improve generalization performance. However, the over-reliance of existing algorithms on oracles continues to burden physicians. To minimize this burden, we propose SoCal, a consistency-based AL framework that dynamically determines whether to request a label from an oracle or to generate a pseudo-label instead. We show that our framework decreases the labelling burden while maintaining strong performance, even in the presence of a noisy oracle.

## 1 Introduction

The success of modern-day deep learning algorithms in the medical domain has been contingent upon the availability of large, labelled datasets (Poplin et al., 2018; Tomašev et al., 2019; Attia et al., 2019). Curating such datasets, however, is a challenge due to the time-consuming nature of, and high costs associated with, labelling. This is particularly the case in the medical domain where the input of expert medical professionals is required. One way of overcoming this challenge and exploiting large, *unlabelled* datasets is via the active learning (AL) framework (Settles, 2009). This framework iterates over three main steps: 1) a learner is tasked with acquiring unlabelled instances, usually through an acquisition function, 2) an oracle (e.g. physician) is tasked with labelling such acquired instances, and 3) the learner is trained on the existing and newly-labelled instances.

By altering the way in which acquisitions are performed and the degree of involvement of the oracle, the active learning framework aims to improve the performance of a network while minimizing the burden of labelling on the oracle. One principal desideratum for an acquisition function is its ability to reduce the size of the version space, the set of hypotheses (decision boundaries) consistent with the labelled training instances. This ability is highly dependent upon the approximation of the version space, a goal that Monte Carlo Dropout (MCD) attempts to achieve (see Fig. 1a). For example, state-of-the-art uncertainty-based acquisition functions, such as BALD (Houlsby et al., 2011), used alongside MCD acquire instances that lie in a region of uncertainty, a region where there is high disagreement between the hypotheses about a particular instance. In many scenarios, however, estimating this region of uncertainty is nontrivial. Furthermore, existing AL frameworks are overly *reliant* on the presence of an oracle. Such over-reliance precludes the applicability of AL algorithms to certain environments, such as low-resource healthcare settings, where an oracle is either unavailable or ill-trained for the task at hand.

**Contributions.** In this work, we aim to design an active learning framework that better estimates the region of uncertainty and decreases its reliance on an oracle. Our contributions are as follows:

1. **Consistency-based Active Learning Framework**: we propose a novel framework that stochastically perturbs inputs, network parameters, or both to guide the acquisition of unlabelled instances.

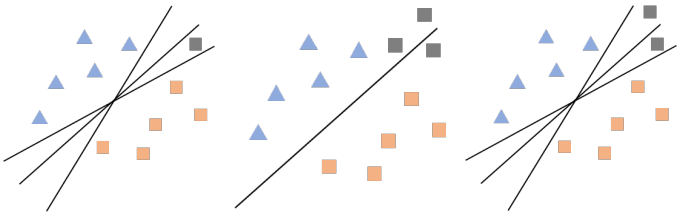
2. **Selective Oracle Questioning**: we propose a dynamic strategy which learns, for an acquired unlabelled instance, whether to request a label from an oracle or to generate a pseudo-label.

## 2 RELATED WORK

**Active learning** methdologies were recently reviewed by Settles (2009). In the healthcare domain, Gong et al. (2019) propose to acquire instances from an electronic health record (EHR) database using a Bayesian deep latent Gaussian model to improve mortality prediction. Smailagic et al. (2018; 2019) acquire unannotated medical images by measuring their distance in a latent space to images in the training set. The work of Wang et al. (2019) is similar to ours in that they focus on the electrocardiogram (ECG). Gal et al. (2017) adopt BALD (Houlsby et al., 2011) in the context of Monte Carlo Dropout to acquire datapoints that maximize the Jensen-Shannon divergence (JSD) across MC samples. Previous work attempts to learn from multiple or imperfect oracles (Dekel et al., 2012; Zhang & Chaudhuri, 2015; Sinha et al., 2019). For example, Urner et al. (2012) propose choosing the oracle that should label a particular instance. Unlike our approach, they do not explore independence from an oracle. Yan et al. (2016) consider oracle abstention in an AL setting. Instead, we place the decision of abstention under the control of the learner. To the best of our knowledge, previous work, in contrast to ours, has assumed the existence of an oracle and has not explored a dynamic oracle selection strategy.

**Consistency training** in the context of semi-supervised learning helps enforce the smoothness assumption (Zhu, 2005). For example, Interpolation Consistency Training Verma et al. (2019) penalizes networks for not generating a linear combination of outputs in response to a linear combination of inputs. Similarly, Xie et al. (2019) penalizes networks for generating drastically different outputs in response to perturbed instances. In the process, networks learn perturbation-invariant representations. McCallumzy & Nigamy (1998) introduce an acquisition function that calculates the average Kullback-Leibler divergence, $\mathcal{D}_{KL}$, between the output of a network and the consensus output across all networks in an ensemble. Unlike ours, their approach does not exploit perturbations. Similar to our work is that of Gao et al. (2019) which incorporates into the objective function a consistency-loss based on the $\mathcal{D}_{KL}$ and actively acquires instances using the variance of the probability assigned to each class by the network in response to perturbed versions of the same instance.

**Selective classification** imbues a network with the ability to abstain from making predictions. Chow (1970); El-Yaniv & Wiener (2010) introduce the risk-coverage trade-off whereby the empirical risk of a model is inversely related to its rate of abstentions. Wiener & El-Yaniv (2011) use a support vector machine (SVM) to rank and reject instances based on the degree of disagreement between hypotheses. In some frameworks, these are the same instances that active learning views as most informative. Cortes et al. (2016) outline an objective function that penalizes abstentions that are inappropriate and frequent. Most recently, Liu et al. (2019) propose the gambler's loss to learn a selection function that determines whether instances are rejected. However, this approach is not implemented in the context of AL. Most similar to our work is SelectiveNet (Geifman & El-Yaniv, 2019) where a multi-head architecture is used alongside an empirical selective risk objective function and a percentile threshold. However, their work assumes the presence of ground-truth labels and, therefore, does not extend to unlabelled instances.



(a) MCD (b) MCP (c) BALC

Figure 1: Labelled instances from two classes and unlabelled instances (gray) alongside the Version space of (a) MCD where each MC sample is viewed as a distinct hypothesis (decision boundary), (b) MCP where there is one hypothesis but several perturbations of the unlabelled instance, and (c) BALC where there are several hypotheses in addition to the unlabelled instance and its perturbed counterpart.

# 3 BACKGROUND

## 3.1 ACTIVE LEARNING

Consider a learner $f_\omega : x \in \mathbb{R}^m \rightarrow v \in \mathbb{R}^d$, parameterized by $\omega$, that maps an $m$-dimensional input, $x$, to a $d$-dimensional representation, $v$. Further consider $g_\phi : v \in \mathbb{R}^d \rightarrow y \in \mathbb{R}^C$ that maps a $d$-dimensional representation, $v$, to a $C$-dimensional output, $y$, where $c$ is the number of classes. After training on a pool of labelled data $L = (X_L, Y_L)$ for $\tau$ epochs, the learner is tasked with querying the unlabelled pool of data $U = (X_U, Y_U)$ and acquiring the top $b$ fraction of instances, $x_b \sim X_U$, that it deems to be most informative. The degree of informativeness of an instance is determined by an acquisition function, $\alpha$, such as BALD (Houlsby et al., 2011). Additional acquisition functions can be found in Appendix A. These are typically used in conjunction with Monte Carlo Dropout (Gal & Ghahramani, 2016) to identify instances that lie in the region of uncertainty, a region in which hypotheses disagree the most about instances.

# 4 METHODS

## 4.1 CONSISTENCY-BASED ACTIVE LEARNING

### 4.1.1 MONTE CARLO PERTURBATIONS

Unlabelled instances in proximity to the decision boundary are likely to be more informative for training than those further away. To identify such instances, we stochastically perturb them and observe the network's outputs. The intuition is that such outputs will differ significantly across the perturbations for instances close to the decision boundary (see Fig. 1b). We refer to this setup as Monte Carlo Perturbations (MCP) and illustrate its derivation in Appendix B.

### 4.1.2 BAYESIAN ACTIVE LEARNING BY CONSISTENCY

Acquisition functions dependent upon perturbations applied to either the inputs (MCP) or the network parameters (MCD) alone can fail to identify instances that lie in the region of uncertainty. Such instances are thus *erroneously* deemed uninformative and will not be acquired. To avoid missing these instances, we propose a framework, entitled Bayesian Active Learning by Consistency (BALC) (see Fig. 1c), that consists of three main steps: 1) we perturb an instance, $x$, to generate $z$, 2) we perturb the network parameters, $\omega$, to generate $\omega'$, and 3) we pass both instances through the *perturbed* network, generating outputs, $p(y|x, \omega')$ and $p(y|z, \omega') \in \mathbb{R}^C$, respectively. We perform these steps for $T$ stochastic perturbations and generate two matrices of network outputs, $G(x), G'(z) \in \mathbb{R}^{T \times C}$.

To leverage $G$ and $G'$, we propose two divergence-based acquisition functions that acquire instances that the network is least robust to. The intuition is that such instances are likely to be in proximity to the decision boundary. In $\text{BALC}_{\text{KLD}}$, we calculate the $D_{KL}$ between two $C$-dimensional Gaussians that are empirically fit to $G$ and $G'$.

$$\text{BALC}_{\text{KLD}} = \mathcal{D}_{KL}(\mathcal{N}(\mu(x), \Sigma(x) \parallel \mathcal{N}(\mu(z), \Sigma(z))) \tag{1}$$

where $\mu = \frac{1}{T}\sum_i^T G$ and $\Sigma = (G - \mu)^T(G - \mu)$ represent the empirical mean vector and covariance matrix of the network outputs, respectively. $\text{BALC}_{\text{KLD}}$ is likely to detect output variations due to input perturbations. To detect variations due to *both* input and parameter perturbations, we introduce $\text{BALC}_{\text{JSD}}$ whose full derivation and motivation can be found in Appendix C and D, respectively.

$$\text{BALC}_{\text{JSD}} = \overbrace{\mathbb{E}_{i \in T}\left[\mathcal{D}_{KL}(G_i(x) \parallel G'_i(z))\right]}^{\text{across parameter perturbations}} - \overbrace{\mathcal{D}_{KL}(\mathbb{E}_{i \in T}[G(x)] \parallel \mathbb{E}_{i \in T}[G'(z)])}^{\text{across input perturbations}} \tag{2}$$

## 4.2 TRACKED ACQUISITION FUNCTION

Deriving the informativeness of unlabelled instances based solely on the value of the acquisition function at a *single* epoch can be erroneous. This is partially driven by limitations in the approximation of the version space, which is known to hinder performance (Cohn et al., 1994). To improve this approximation, we propose to *track* an acquisition function over time (e.g., epochs) before employing it to acquire instances. The intuition is that by incorporating temporal information, we accumulate

hypotheses in the version space and thus obtain a more reliable estimate of the relative informativeness of each instance. Acquiring such instances would help reduce the size of the version space at a greater rate. For any tracked acquisition function, $\alpha(t)$, the corresponding area under the temporal acquisition function, AUTAF $\in \mathbb{R}^1$, is calculated as follows:

$$\text{AUTAF} = \int_0^{\tau} \alpha(t)dt \approx \sum_{t=0}^{\tau} \left( \frac{\alpha(t + \Delta t) + \alpha(t)}{2} \right) \Delta t \tag{3}$$

where the integral is approximated using the trapezoidal rule, $\Delta t$ is the time-step (in epochs) between epochs at which ordinary acquisition values are calculated, and $\tau$ is the epoch at which the AUTAF is calculated and an acquisition of unlabelled instances is performed.

### 4.3 Selective Oracle Questioning

Alongside our consistency-based AL framework, we aim to minimize the burden of labelling on an oracle. To do so, we learn a network that dynamically determines whether to request a label from an oracle or to generate a pseudo-label for each acquired unlabelled instance. We refer to this strategy as selective oracle questioning in active learning (SoQal).

**Oracle selection network.** In addition to the learners outlined in Sec. 3.1, $f_\omega$ and $g_\phi$, we introduce an oracle selection network, $h_\theta : v \in \mathbb{R}^d \rightarrow t \in [0, 1]$ parameterized by $\theta$, that maps $d$-dimensional representations, $v$, to a scalar, $t$, as shown in Fig. 2.

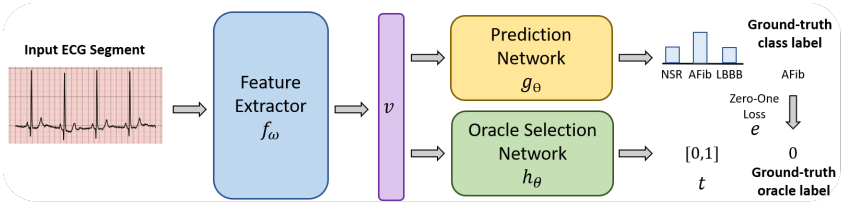



Figure 2: Selective oracle questioning framework. An instance, $x$, is provided as input to the feature extractor, $f_\omega$. The representation, $v$, is passed through $g_\phi$ to generate task predictions, and through $h_\theta$ to help determine whether a label is requested from an oracle. The zero-one loss of $g_\phi$ acts as the ground-truth label for $t$. $t \approx 1$ indicates a hard-to-classify instance and thus would benefit from an oracle label.

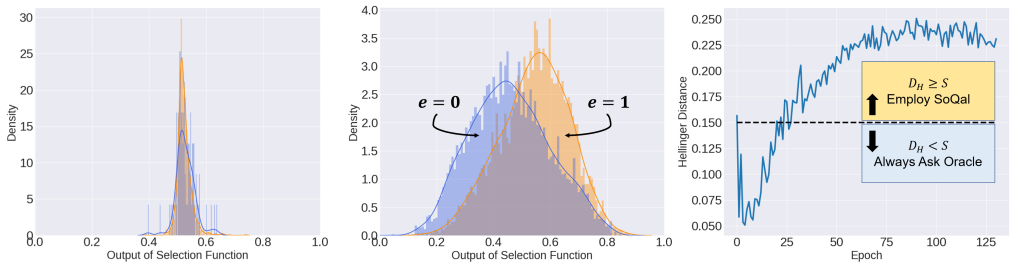
**Proxy for misclassifications.** Ideally, a network should only be reliant on an oracle when it misclassifies an instance itself. Quantifying these misclassifications is trivial in the presence of ground-truth labels. However, in the AL framework, we are interested in making decisions on *unlabelled* instances. Therefore, we need a reliable proxy for such misclassifications. We propose that this proxy be the output of the oracle selection network, $t$. For example, low and high values of $t$ can indicate correct and incorrect network predictions, respectively. To learn this behaviour, $h_\theta$ needs an appropriate supervisory signal. For this, we choose the zero-one loss, $e$, of the prediction network, $g_\phi$, as the ground-truth label (see Fig. 2). For a mini-batch of size, $B$, our objective function thus consists of two terms: 1) a cross-entropy class prediction loss for the main task, and 2) a binary cross-entropy oracle selection loss, with a weighting coefficient, $\beta$, (described next).

$$\mathcal{L} = \sum_{i=1}^{B} \overbrace{-\log\left(p(y_i = c|x_i, \omega, \phi)\right)}^{\text{class prediction loss}} - \overbrace{\beta e_i \log\left(h_\theta(t|x_i)\right) - (1 - e_i)\log\left(1 - h_\theta(t|x_i)\right)}^{\text{oracle selection loss}} \tag{4}$$

where $c$ is the target class. During the early stages of training on *labelled* data, a network struggles to classify instances correctly. This means that the ratio of zero to one losses will be skewed toward the latter. As training progresses and the network becomes more adept at classifying instances, this ratio becomes skewed in the opposite direction. Such an imbalance in the ground-truth labels, $e$, and their subsequent shift send strong supervisory signals to $h_\theta$. For example, during the early stages of training, the outputs of $h_\theta$ will be high and pulled towards $t = 1$ (high error) even if the corresponding instance was correctly classified. Such behaviour makes it difficult to ascertain whether instances have been misclassified, hindering the reliability of $t$ as a proxy. To offset the aforementioned class

imbalance, we introduce a dynamic hyperparameter, $\beta = \frac{\sum \delta_{e=0}}{\sum \delta_{e=1}}$, where $\delta$ is the Kronecker delta function. As training progress, $\beta < 1 \rightarrow \beta > 1$, as the ratio of correctly classified ($e = 0$) to misclassified ($e = 1$) instances within a mini-batch changes.

**Decision-making with proxy.** We aim to exploit $t$ for the binary decision of either requesting a label from an oracle or generating a pseudo-label. One way to do so is via a simple threshold at $0.5$. However, this threshold may not be appropriate. In designing a robust selection strategy, we must account for the distribution of the $t$ values that corresponds to each decision and the separability of such distributions. In Figs. 3a and 3b, we illustrate these distributions during the early and late stages of training, colour-coded based on whether the $t$ values correspond to correctly-classified ($e = 0$) or misclassified ($e = 1$) *labelled* training instances.




(a) Early stage of training (b) Late stage of training (c) Hellinger distance

Figure 3: Density of the outputs, $t$, of $g_\theta$ colour-coded based on the zero-one classification error during the (a) early and (b) late stages of training. (c) the Hellinger distance, $\mathcal{D}_H$, between the two distributions of $t$ during training increases as they become more separable.

We now outline how the binary decision is made. After each training epoch, we fit the $t$ values in Fig. 3b to two unimodal Gaussian distributions. This generates $\mathcal{N}_0(\mu_0, \sigma_0^2)$ and $\mathcal{N}_1(\mu_1, \sigma_1^2)$ for $e = 0$ and $e = 1$, respectively. We choose to quantify the separability of these two distributions using the Hellinger distance, $\mathcal{D}_H \in [0, 1]$, as it allows for a straightforward threshold. Low separability expressed as $\mathcal{D}_H < S$ implies that $h_\theta$ has yet to generate a reliable proxy and thus an oracle is requested for a label. We note that the value of $S$ can be altered depending on the relative level of trust one has in the network and oracle. When $\mathcal{D}_H \geq S$, we evaluate $\mathcal{N}_0$ and $\mathcal{N}_1$ at the $t$ output for each acquired unlabelled instance and define the probability of requesting a label from an oracle, $p(\text{A})$, as:

$$p(\text{A}) = \begin{cases} 1, & \mathcal{D}_H < S \\ 1, & \mathcal{N}\left(t|\mu_1, \sigma_1^2, e = 1\right) > \mathcal{N}\left(t|\mu_0, \sigma_0^2, e = 0\right) \text{ and } \mathcal{D}_H \geq S \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

We refer readers to Appendix F for algorithms elucidating the entire active learning framework.

## 5 Experimental Design

### 5.1 Datasets

We conduct experiments in PyTorch (Paszke et al., 2019) on four publically-available datasets. These datasets consist of physiological time-series data such as the photoplethysmogram (PPG) and the electrocardiogram (ECG) alongside cardiac arrhythmia labels. We use $\boldsymbol{\mathcal{D}_1}$ = PhysioNet 2015 PPG, $\boldsymbol{\mathcal{D}_2}$ = PhysioNet 2015 ECG (Clifford et al., 2015) (5-way), $\boldsymbol{\mathcal{D}_3}$ = PhysioNet 2017 ECG (Clifford et al., 2017) (4-way), $\boldsymbol{\mathcal{D}_4}$ = Cardiology ECG (Hannun et al., 2019) (12-way).

To observe the impact of the availability of labelled training data on the active learning procedure, we take a fraction $\beta$ = (0.1, 0.3, 0.5, 0.7, 0.9) of the training dataset and place it into the labelled set. Its complement is placed into the unlabelled set. Details about the data splits, preprocessing steps, and the network architecture can be found in Appendices G and H.

### 5.2 BASELINES

**Acquisition Functions.** We compare our novel active learning framework to the state-of-the-art acquisition functions used in conjunction with MCD. These include **Var Ratio**, **Entropy**, and **BALD**, definitions of which can be found in Appendix A. We also compare to the scenario in which active learning is not employed **(No AL)**.

**Selective Oracle Questioning.** We experiment with baselines that exhibit varying degrees of oracle dependence. **No Oracle** is a scenario in which 0% of the labels that correspond to unlabelled instances are oracle-based and are instead pseudo-labelled based on network predictions. **Epsilon Greedy** is a stochastic strategy Watkins (1989) that we adapt to exponentially decay the reliance of the network on an oracle as a function of the number of acquisition epochs. **Entropy Response** assumes that high entropy predictions generated by a network are indicative of instances that the network is unsure of. Therefore, we introduce a threshold, $S_{\text{Entropy}}$, such that if it is exceeded, an oracle is requested to label the chosen instance (see Appendix H). The most dependent baseline is **100% Oracle**, a traditionally-employed strategy in AL where 100% of the labels are oracle-based.

We do not compare our methods to Softmax Response (Geifman & El-Yaniv, 2017) and SelectiveNet (Geifman & El-Yaniv, 2019), despite their strong performance for selective classification, as they do not trivially extend to the setting in which labels are unavailable.

### 5.3 HYPERPARAMETERS

For all experiments, we chose the number of MC samples $T = 20$ to balance between computational complexity and accuracy of the approximation of the version space. We acquire unlabelled instances at pre-defined epochs during training which we refer to as acquisition epochs, $\tau = 5n$, $n \in \mathbb{N}^+$. During each acquisition epoch, we acquire $b = 2\%$ of the remaining unlabelled instances. We also investigate the effect of such hyperparameters on performance (see Appendices N-P). When experimenting with tracked acquisition functions, we chose the temporal period, $\Delta t = 1$.

**Selective Oracle Questioning.** Recall that we delegate selective oracle questioning to the network only when $\mathcal{D}_H \geq S$. Given $\mathcal{D}_H$'s increasing trend during training (see Fig. 3c), we chose $S = 0.15$ to balance between the reliability of the proxy and the independence of the network from an oracle. We also explore the sensitivity of SoQal to this choice of $S$.

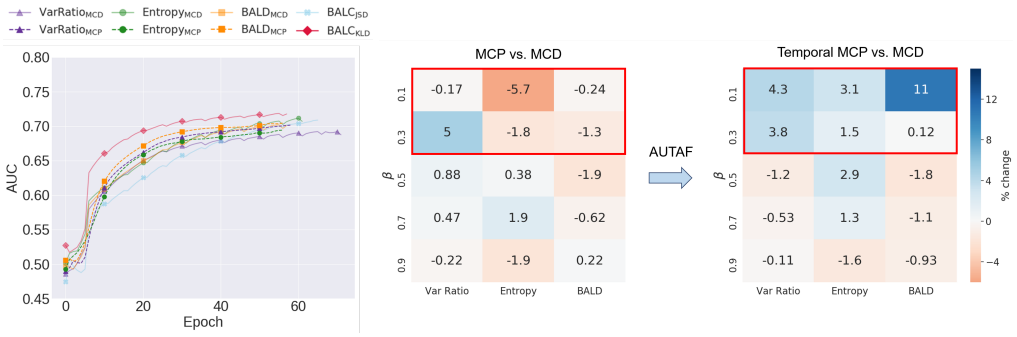
## 6 EXPERIMENTAL RESULTS

### 6.1 ACTIVE LEARNING WITHOUT ORACLE

The AL framework assumes the presence of an oracle (expert annotator). In this section, we evaluate the ability of our framework to operate *without* an oracle. In Fig. 4a, we illustrate the validation AUC of the various methods when initially exposed to only 30% of the labelled training data of $\mathcal{D}_2$.

We see that $\text{BALC}_{\text{KLD}}$ converges faster and better than the remaining methods. For example, after 20 epochs, it achieves the same AUC $\approx 0.69$ as $\text{BALD}_{\text{MCD}}$ does at epoch 40, representing a two-fold reduction in training time. It also achieves a higher final AUC $\approx 0.72$ relative to the other methods. We hypothesize that such behaviour is due to $\text{BALC}_{\text{KLD}}$'s improved ability to estimate the region of uncertainty, and thus acquires more informative instances. Secondly, since these experiments are conducted without an oracle, $\text{BALC}_{\text{KLD}}$ appears to acquire informative instances that are also pseudo-labelled correctly. Satisfying both of these criteria suggests that such instances are *closer* to the decision boundary than their non-acquired counterparts yet are still on the correct side of the boundary. We arrive at similar conclusions for the remaining experiments (see Appendix I).

In attempt to better quantify the benefit of our methods, we illustrate in Fig. 4b the percent change in the AUC when using MCP, with and without tracked acquisition functions, relative to MCD. We find that tracked acquisition functions are most useful when the initial size of the labelled dataset is small ($\downarrow \beta$ values) (red rectangle). For example, incorporating temporal information into BALD at $\beta = 0.1$ improves the generalization performance by 11%. We hypothesize that this outcome stems from an improvement in the approximation of the version space, brought about by the increased enumeration of hypotheses over time.




(a) $\mathcal{D}_2$ at $\beta = 0.3$

(b) Utility of tracked acquisition functions on $\mathcal{D}_1$

Figure 4: **(a)** Validation AUC on $\mathcal{D}_2$ at $\beta = 0.3$. **(b)** Mean percent change in test AUC when comparing MCP with (left) static and (right) tracked acquisition functions to MCD with their static counterparts. We show results for Var Ratio, Entropy, and BALD, at all fractions, $\beta$.

### 6.2 Active Learning with Noise-free Oracle

The ability of a learner to appropriately determine *when* to request labels from an oracle can significantly alleviate the associated labelling burden. In this section, we evaluate this ability under the assumption of an oracle that provides accurate labels, i.e., is noise-free. In Table 1, we illustrate the test-set AUC of the oracle questioning strategies on all datasets at $\beta = 0.1$.

Table 1: Mean test AUC of oracle questioning strategies in the presence of a noise-free oracle. Results are shown for a subset of the acquisition functions on $\mathcal{D}_1 - \mathcal{D}_4$ and are averaged across five seeds.

| Dataset | Ac. Function $\alpha$ | Oracle Questioning Method | | | | | |
|---|---|---|---|---|---|---|---|
| | | No Oracle | Entropy Response | Epsilon Greedy | SoQal (ours) | 100% Oracle | No AL |
| $\mathcal{D}_1$ | $\text{BALD}_{\text{MCD}}$ | $0.465 \pm 0.017$ | $0.496 \pm 0.039$ | $0.491 \pm 0.028$ | $\mathbf{0.621 \pm 0.021}$ | $0.653 \pm 0.013$ | $0.577 \pm 0.014$ |
| | $\text{BALD}_{\text{MCP}}$ | $0.464 \pm 0.023$ | $0.517 \pm 0.043$ | $0.501 \pm 0.043$ | $\mathbf{0.645 \pm 0.015}$ | $0.676 \pm 0.020$ | |
| | $\text{BALC}_{\text{KLD}}$ | $0.500 \pm 0.023$ | $0.548 \pm 0.034$ | $0.548 \pm 0.042$ | $\mathbf{0.598 \pm 0.055}$ | $0.634 \pm 0.030$ | |
| | Temporal $\text{BALC}_{\text{KLD}}$ | $0.496 \pm 0.024$ | $0.536 \pm 0.040$ | $0.521 \pm 0.059$ | $\mathbf{0.646 \pm 0.067}$ | $0.659 \pm 0.033$ | |
| $\mathcal{D}_2$ | $\text{BALD}_{\text{MCD}}$ | $0.573 \pm 0.063$ | $0.584 \pm 0.041$ | $0.609 \pm 0.071$ | $\mathbf{0.707 \pm 0.038}$ | $0.713 \pm 0.053$ | $0.679 \pm 0.040$ |
| | $\text{BALD}_{\text{MCP}}$ | $0.589 \pm 0.045$ | $0.638 \pm 0.043$ | $0.637 \pm 0.044$ | $\mathbf{0.677 \pm 0.042}$ | $0.735 \pm 0.028$ | |
| | $\text{BALC}_{\text{KLD}}$ | $0.602 \pm 0.044$ | $0.582 \pm 0.017$ | $0.643 \pm 0.033$ | $\mathbf{0.677 \pm 0.024}$ | $0.722 \pm 0.018$ | |
| | Temporal $\text{BALC}_{\text{KLD}}$ | $0.575 \pm 0.017$ | $0.612 \pm 0.050$ | $0.605 \pm 0.019$ | $\mathbf{0.648 \pm 0.057}$ | $0.735 \pm 0.011$ | |
| $\mathcal{D}_3$ | $\text{BALD}_{\text{MCD}}$ | $0.581 \pm 0.014$ | $0.588 \pm 0.013$ | $0.673 \pm 0.015$ | $\mathbf{0.721 \pm 0.025}$ | $0.802 \pm 0.008$ | $0.716 \pm 0.012$ |
| | $\text{BALD}_{\text{MCP}}$ | $0.623 \pm 0.020$ | $0.676 \pm 0.058$ | $0.665 \pm 0.028$ | $\mathbf{0.720 \pm 0.044}$ | $0.798 \pm 0.007$ | |
| | $\text{BALC}_{\text{KLD}}$ | $0.631 \pm 0.010$ | $0.629 \pm 0.004$ | $0.643 \pm 0.041$ | $\mathbf{0.731 \pm 0.033}$ | $0.787 \pm 0.008$ | |
| | Temporal $\text{BALC}_{\text{KLD}}$ | $0.600 \pm 0.005$ | $0.630 \pm 0.014$ | $0.654 \pm 0.019$ | $\mathbf{0.730 \pm 0.024}$ | $0.794 \pm 0.002$ | |
| $\mathcal{D}_4$ | $\text{BALD}_{\text{MCD}}$ | $0.486 \pm 0.011$ | $0.489 \pm 0.030$ | $0.474 \pm 0.037$ | $0.468 \pm 0.021$ | $0.585 \pm 0.011$ | $0.486 \pm 0.023$ |
| | $\text{BALD}_{\text{MCP}}$ | $0.493 \pm 0.030$ | $0.504 \pm 0.026$ | $0.492 \pm 0.024$ | $0.499 \pm 0.029$ | $0.605 \pm 0.024$ | |
| | $\text{BALC}_{\text{KLD}}$ | $0.505 \pm 0.032$ | $0.504 \pm 0.039$ | $0.473 \pm 0.010$ | $0.495 \pm 0.012$ | $0.588 \pm 0.033$ | |
| | Temporal $\text{BALC}_{\text{KLD}}$ | $0.511 \pm 0.030$ | $0.496 \pm 0.023$ | $0.496 \pm 0.023$ | $0.503 \pm 0.010$ | $0.532 \pm 0.027$ | |

We show that SoQal consistently outperforms its counterparts across $\mathcal{D}_1$ - $\mathcal{D}_3$,. For example, while using $\text{BALD}_{\text{MCD}}$ on $\mathcal{D}_2$, SoQal achieves an $\text{AUC} = 0.707$ compared to $\text{AUC} = 0.609$ and $0.584$ for Epsilon Greedy and Entropy Response, respectively. These findings suggest that SoQal is better equipped to know *when* and for which *instance* a label should be requested from an oracle. One could argue that SoQal's superiority is due to its high dependence on the oracle. In fact, we show that this is not the case in Appendix J. On the other hand, we observe that SoQal performs on par with the other methods on $\mathcal{D}_4$. We hypothesize that this outcome is due to the cold-start problem (Konyushkova et al., 2017) whereby AL algorithms fail to learn due to the limited availability of labelled training data. We support this claim with experiments in Appendix K. Moreover, we remind readers that by increasing the value of $S$ in the SoQal experiments, networks can cede more control to the oracle and thus further improve performance, an effect we quantify in Appendix L.

### 6.3 Active Learning with Noisy Oracle

So far, we have presented scenarios in which oracles are either absent or present with the ability to provide accurate labels. In healthcare, however, physicians may be ill-trained, fatigued, or unable

to diagnose a case due to its difficulty. We simulate this scenario by introducing two types of label noise. We stochastically flip each label 1) to any other label randomly **(Random)**, or 2) to its nearest neighbour from a *different* class in a compressed subspace **(Nearest Neighbour)**. Whereas the first form of noise is extreme, the latter is more realistic as it may represent uncertainty in physician diagnoses. To simulate various magnitudes of noise, we chose the probability of introducing noise, $\gamma = [0.05, 0.1, 0.2, 0.4, 0.8]$. In Fig. 5, we illustrate the effect of label noise on the test AUC for the various oracle questioning strategies.

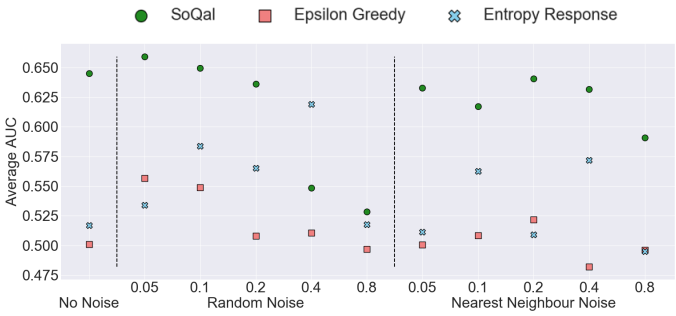



Figure 5: Average test AUC for the oracle questioning strategies in the absence and presence, of various magnitudes, of label noise on $\mathcal{D}_1$ using $\text{BALD}_{\text{MCP}}$. With up to 80% random or nearest neighbour label noise, SoQal still outperforms its counterpart methods that are trained *without* label noise.

We find that SoQal outperforms the remaining strategies regardless of noise type and magnitude (except at 40% random noise). For example, at 5% random noise, SoQal achieves an $\text{AUC} \approx 0.66$ compared to $\text{AUC} \approx 0.56$ and $\approx 0.53$ for Epsilon Greedy and Entropy Response, respectively. Surprisingly, we also observe that the introduction of label noise can sometimes improve performance. This can be seen by the increase in SoQal's $\text{AUC} \approx 0.64 \rightarrow 0.66$ with no noise and 5% random noise, respectively. We hypothesize that this is due to inherent label noise in the datasets. Therefore, by introducing further noise, we may be nudging these labels towards their ground-truth values. Moreover, SoQal is better able to deal with label noise than its counterparts. Specifically, SoQal at 80% random noise achieves $\text{AUC} \approx 0.53$ which is greater than $\text{AUC} \approx 0.50$ and $\approx 0.52$ achieved by Epsilon Greedy and Entropy Response trained *without noise*, respectively. This effect, which is even more pronounced when dealing with nearest neighbour noise, indicates the utility of SoQal in the presence of a noisy oracle. We arrive at similar conclusions when experimenting with other datasets and acquisition functions (see Appendix M).

## 7 Discussion and Future Work

In this paper, we propose a novel consistency-based active learning framework which perturbs both inputs and network parameters and acquires instances to which the network is least robust. We illustrate the utility of this approach in the *absence* of an oracle. Moreover, we propose a strategy that dynamically determines whether an oracle should be requested for a label. We empirically show that this approach is better able to deal with noisy oracles than the baseline methods. We now elucidate several avenues worth exploring.

**Incorporating Prior Information.** The default mode for SoQal is deferral to an oracle. However, relevant a priori information, such as the degree of noise inherent in the oracle's labels, can be incorporated to alter either the default mode or the Hellinger threshold, $S$.

**Incorporating Multiple Oracles.** In this work, we focused on the presence of a single oracle. Scenarios in which multiple oracles exist may better reflect clinical environments which include multiple experts of various levels of competency. Therefore, dynamically querying these oracles might be of interest.

## REFERENCES

Zachi I Attia, Suraj Kapa, Francisco Lopez-Jimenez, Paul M McKie, Dorothy J Ladewig, Gaurav Satam, Patricia A Pellikka, Maurice Enriquez-Sarano, Peter A Noseworthy, Thomas M Munger, et al. Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram. *Nature Medicine*, 25(1):70–74, 2019.

C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Gari D Clifford, Ikaro Silva, Benjamin Moody, Qiao Li, Danesh Kella, Abdullah Shahin, Tristan Kooistra, Diane Perry, and Roger G Mark. The physionet/computing in cardiology challenge 2015: reducing false arrhythmia alarms in the icu. In *2015 Computing in Cardiology Conference*, pp. 273–276, 2015.

Gari D Clifford, Chengyu Liu, Benjamin Moody, H Lehman Li-wei, Ikaro Silva, Qiao Li, AE Johnson, and Roger G Mark. Af classification from a short single lead ECG recording: the physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology*, pp. 1–4, 2017.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pp. 67–82. Springer, 2016.

Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13(Sep):2655–2697, 2012.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(May):1605–1641, 2010.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international Conference on Machine Learning*, pp. 1050–1059, 2016.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1183–1192. JMLR. org, 2017.

Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan O Arik, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: towards minimizing labeling cost. *arXiv preprint arXiv:1910.07153*, 2019.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in neural information processing systems*, pp. 4878–4887, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. *arXiv preprint arXiv:1901.09192*, 2019.

Wenbo Gong, Sebastian Tschiatschek, Richard Turner, Sebastian Nowozin, and José Miguel Hernández-Lobato. Icebreaker: element-wise active information acquisition with bayesian deep latent gaussian model. *arXiv preprint arXiv:1908.04537*, 2019.

Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25(1):65, 2019.

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

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pp. 4225–4235, 2017.

Ziyin Liu, Zhikang Wang, Paul Pu Liang, Russ R Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: Learning to abstain with portfolio theory. In *Advances in Neural Information Processing Systems*, pp. 10622–10632, 2019.

Andrew Kachites McCallumzy and Kamal Nigamy. Employing em and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning*, pp. 359–367, 1998.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Ryan Poplin, Avinash V Varadarajan, Katy Blumer, Yun Liu, Michael V McConnell, Greg S Corrado, Lily Peng, and Dale R Webster. Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. *Nature Biomedical Engineering*, 2(3):158, 2018.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, Department of Computer Sciences, 2009.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5972–5981, 2019.

Asim Smailagic, Pedro Costa, Hae Young Noh, Devesh Walawalkar, Kartik Khandelwal, Adrian Galdran, Mostafa Mirshekari, Jonathon Fagert, Susu Xu, Pei Zhang, et al. Medal: Accurate and robust deep active learning for medical image analysis. In *IEEE International Conference on Machine Learning and Applications*, pp. 481–488, 2018.

Asim Smailagic, Pedro Costa, Alex Gaudio, Kartik Khandelwal, Mostafa Mirshekari, Jonathon Fagert, Devesh Walawalkar, Susu Xu, Adrian Galdran, Pei Zhang, et al. O-medal: Online active deep learning for medical image analysis. *arXiv preprint arXiv:1908.10508*, 2019.

Nenad Tomašev, Xavier Glorot, Jack W Rae, Michal Zielinski, Harry Askham, Andre Saraiva, Anne Mottram, Clemens Meyer, Suman Ravuri, Ivan Protsyuk, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 572(7767):116–119, 2019.

Ruth Urner, Shai Ben David, and Ohad Shamir. Learning from weak teachers. In *Artificial intelligence and statistics*, pp. 1252–1260, 2012.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

Guijin Wang, Chenshuang Zhang, Yongpan Liu, Huazhong Yang, Dapeng Fu, Haiqing Wang, and Ping Zhang. A global and updatable ecg beat classification system based on recurrent neural networks and active learning. *Information Sciences*, 501:523–542, 2019.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Yair Wiener and Ran El-Yaniv. Agnostic selective classification. In *Advances in Neural Information Processing Systems*, pp. 1665–1673, 2011.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. 2019.

Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active learning from imperfect labelers. In *Advances in Neural Information Processing Systems*, pp. 2128–2136, 2016.

Chicheng Zhang and Kamalika Chaudhuri. Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, pp. 703–711, 2015.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison, Department of Computer Sciences, 2005.