Is the Most Accurate AI the Best Teammate? Optimizing AI for Teamwork

Gagan Bansal[1] **Besmira Nushi**[2] **Ece Kamar**[2] **Eric Horvitz**[2] **Daniel S. Weld**[1,3]
[1]University of Washington [2]Microsoft Research [3]Allen Institute for AI

Abstract

AI practitioners typically strive to develop the most *accurate* systems, making an implicit assumption that the AI system will function autonomously. However, in practice, AI systems often are used to provide *advice* to people in domains ranging from criminal justice and finance to healthcare. In such *AI-advised decision making*, humans and machines form a *team*, where the human is responsible for making final decisions. But is the most accurate AI the best teammate? We argue "No" — predictable performance may be worth a slight sacrifice in AI accuracy. Instead, we argue that AI systems should be trained in a human-centered manner, directly optimized for *team performance*. We study this proposal for a specific type of human-AI teaming, where the human overseer chooses to either *accept* the AI recommendation or *solve* the task themselves. To optimize the team performance for this setting we maximize the team's *expected utility*, expressed in terms of the quality of the final decision, cost of verifying, and individual accuracies of people and machines. Our experiments with linear and non-linear models on real-world, high-stakes datasets show that the most accuracy AI may not lead to highest team performance and show the benefit of modeling teamwork during training through improvements in expected team utility across datasets, considering parameters such as human skill and the cost of mistakes. We discuss the shortcoming of current optimization approaches beyond well-studied loss functions such as log-loss, and encourage future work on AI optimization problems motivated by human-AI collaboration.

1 Introduction

Many AI systems are developed for use in collaborative settings, where people work with an AI teammate. For example, numerous applications of AI have been designed as advisory tools, providing input to people who are tasked with making final decisions. Beyond the appropriateness of people making the final calls, the advisory role of AI systems may be obligatory; legal requirements may prohibit complete automation (GDPR 2020; Nickelsburg 2020). Studies have demonstrated domains and tasks where human-AI teams may perform better than either the AI or human alone (Nagar and Malone 2011; Patel et al. 2019; Kamar, Hacker, and Horvitz 2012). For human-AI teams, optimizing the performance of the whole team is more important than optimizing the performance of an individual member. Yet, to date, the AI community has focused on maximizing the individual accuracy of machine-learned models, assuming implicitly that this will optimize team performance. This raises an important question: Is the most accurate AI the best possible teammate for a human?

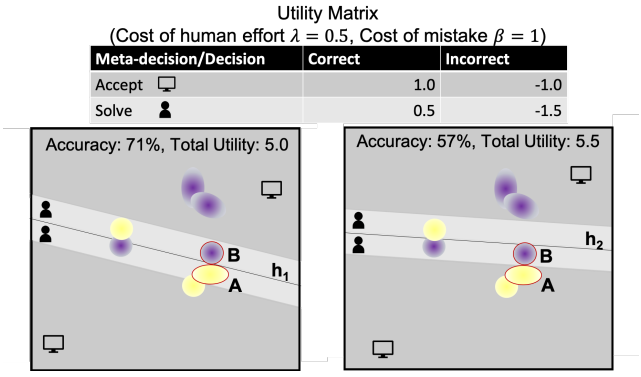

Figure 1: Consider a binary classification problem (purple vs. yellow). Assume each blob is uniformly distributed and of the same size. In a human-AI team, a more accurate classifier (h_1, left pane, learned using log-loss) may produce lower *team utility* than a less accurate model (h_2, right pane). Suppose the human can either quickly *accept* the AI's recommendation or *solve* the task themselves, incurring a cost λ in time or effort, to yield a more reliable result. The payoff matrix describes the utility of different outcomes. We explore the policy where humans accept recommendations when the AI is confident, but verify uncertain predictions (shown in the light grey region surrounding each hyperplane). While h_2 is less accurate than h_1 (because B is incorrectly classified), it results in a higher team utility: Since h_2 moved A outside the verify region, there are more *correctly classified* inputs on which the user can rely on the system.

We argue that the most accurate model is not *necessarily* the best teammate. We show this formally, but the intuition is simple. Considering human-human teams, *Is the best-ranked tennis player necessarily the best doubles teammate?* Clearly not—teamwork puts additional demands on participants that extend beyond individual performance on

tasks, such as ability to complement and coordinate with one's partner. Similarly, creating high-performing human-AI teams may require training AI systems that exhibit additional *human-centered* properties, e.g., facilitating appropriate levels of trust and delegation. Implicitly, this is the motivation behind much work in intelligible AI, including efforts aimed at enhancing the understandability of complex AI inference (Horvitz et al. 1986), interpretability of machine-learned models (Caruana et al. 2015; Weld and Bansal 2019), and performing post-hoc explanations of the output of models (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017). We move beyond such general motivation and highlight the value of developing methods to model and optimize the collaborative process.

For example, consider the scenario when the system generates advice in which it is uncertain. In practice, users are likely to distrust such recommendations, and rightly so, because a low confidence is often correlated with erroneous predictions (Bansal et al. 2020; Hendrycks and Gimpel 2017). In this work, we assume that, when systems have low confidence in their inferences, users will discard the recommendation and *solve* the task themselves, incurring a cost based in the required additional human effort. As a result, team performance depends on the AI accuracy only in the *accept region*, *i.e.*, the region where a user is actually likely to rely on AI. The singular objective of optimizing for AI accuracy (*e.g.*, using log-loss) may hurt team performance when the model has fixed inductive bias. Team performance will benefit from improving AI in the accept regions even if at the cost of performance over the complementary *solve regions* (Figure 1). While there exist other aspects of collaboration that can also be addressed via optimization techniques, such as model interpretability, supporting complementary skills (Wilder, Horvitz, and Kamar 2020), or enabling learning among partners, the problem we address in this paper is to account for team-based utility as a basis for collaboration. In sum:

1. We highlight an important direction in the field of human-centered AI: When paired with a human overseer, the most accurate ML model may not lead to the highest *team performance*. Specifically, we consider settings where, during training the system considers humans' mental model of the AI and how they make use of its recommendations. This setting complements recent advances in *learning to defer* where systems are trained when to refuse to share a recommendation to the overseer.
2. For a simple yet ubiquitous form of teamwork, we show that log-loss, the most popular loss function for optimizing AI accuracy, can lead to suboptimal team performance and instead propose directly optimizing for human-AI team's utility. During training, the new objective considers and guides AI performance by considering various human and domain parameters, such as human accuracy, cost of human effort, and cost of mistakes.
3. We present experiments on real-world datasets and models that show improvements in expected team utility achieved by our method. We present qualitative analyses to understand how the re-trained model differs from the most accurate AI, and how the improvements in utility change as a function of domain parameters. We conclude with discussing optimization issues, loss-metric mismatch, and implications for optimizing team performance for more complex human-AI teams.

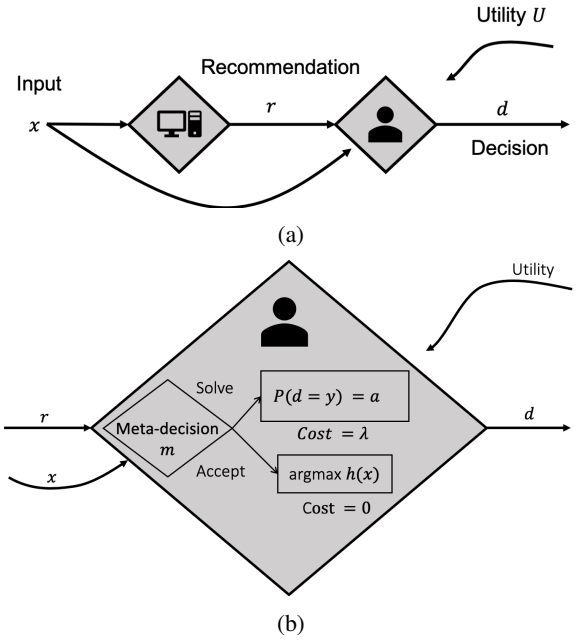

Figure 2: (a) AI-advised decision making. (b) To make a decision, the human either accepts or overrides a recommendation. The `Solve` meta-decision is costlier than `Accept`.

2 Problem Description

We focus on AI-advised decision making scenarios where a classifier h gives recommendations to a human decision maker to help them make decisions (Figure 2a). Suppose x is an n-dimensional feature vector (i.e., $X \subset R^n$) and Y is a finite set of possible decisions. For example, for binary classification $Y = \{+, -\}$. If $h(x)$ denotes the classifier's output (*i.e.*, a probability distribution over the set of possible outcomes $\mathcal{Y}$) the recommendation r is a tuple consisting of the predicted label $\hat{y} = \arg\max h(x)$ and a confidence value $\max h(x)$, *i.e.*, $r := (\hat{y}, \max h(x))$. Using this recommendation, the user computes a final decision d. The environment, in response, returns a utility which depends on the quality of the final decision and any cost incurred due to human effort. If the team classifies a sequence of instances, the objective is to maximize the cumulative utility. Before deriving a closed form equation for the objective, we describe the form of the human-AI collaboration we consider along with our assumptions. We study this simple setting as a step to exploring broader opportunities and challenges in team-centric optimization.

1. *User either accepts the recommendation or solves the task themselves:* The human computes the final decision by first making a meta-decision m: `Accept` or `Solve` (Figure 2b). In `Accept`, the user passes off the AI recom-

Meta-decision/Decision	Correct	Incorrect
`Accept` [A]	1	$-\beta$
`Solve` [S]	$1-\lambda$	$-\beta-\lambda$

Table 1: Utility as a function of meta-decision and decision.

mendation as the final decision. In contrast, in `Solve`, the user ignores the recommendation and computes the final decision themselves. Let m denote the function that maps an input instance and recommendation to a meta-decision in $\mathcal{M} = \{\texttt{Accept}, \texttt{Solve}\}$. Further, U denotes the utility function, which depends on the human meta-decision and final decision d (Figure 1). As a result, the optimal classifier h^* would maximize the team's expected utility:

$$h^* = \arg\max_h \mathbb{E}_{x,y}[U(m,d)] \tag{1}$$

2. *Mistakes are costly:* A correct decision results in unit reward. An incorrect decision results in a penalty $\beta \geq 1$.
3. *Solving the task is costly:* Since it takes time and effort for the human to perform the task themselves (*e.g.*, cognitive effort), we realistically assume that the `Solve` meta-decision costs more than `Accept`. Further, without loss of generality, we assume λ units of cost to `Solve` and zero cost to `Accept`. Note that even when the cost of `Accept` is non-zero and the reward for a correct decision is different than one, the utility function can still be transformed and simplified to the same form as in Table 1 and be optimized in the same way as we describe henceforth.
   Following the above specifications, we obtain the utility function in Figure 1. The values in the table originate from subtracting the cost of the action from the reward.
4. *Human is uniformly accurate across decisions:* Let $a \in [0,1]$ denote the conditional probability that if the user solves the task, they will make the correct decision.
$$P(d = y | m = \texttt{S}) = a \tag{2}$$
5. *Human is rational:* The user chooses the meta-decision that results in highest expected utility. Further, the user trusts the classifier's confidence $h(x)[\hat{y}]$ as an accurate indicator of the recommendation's reliability, i.e., true conditional probability of prediction $\hat{y}$ being correct. As a result, the user will choose `Accept` if and only if the expected utility of `Accept` is greater than that of `Solve`.
$$\begin{aligned}\mathbb{E}[U(m=\texttt{A})] &\geq \mathbb{E}[U(m=\texttt{S})]\\ h(x)[\hat{y}] - (1-h(x)[\hat{y}])\cdot\beta &\geq a - (1-a)\cdot\beta - \lambda\\ h(x)[\hat{y}] &\geq a - \frac{\lambda}{1+\beta}\end{aligned}$$
   Let $c(\beta, \lambda, a)$ denote the minimum value of confidence for which the user's meta-decision is `Accept`.
$$c(\beta,\lambda,a) = a - \frac{\lambda}{1+\beta} \tag{3}$$
   This implies the human will follow the following threshold-based policy to make meta-decisions:
$$P(m=\texttt{A}) = \begin{cases} 1 & \text{if } h(x)[\hat{y}] \geq c(\beta,\lambda,a) \\ 0 & \text{otherwise}\end{cases} \tag{4}$$

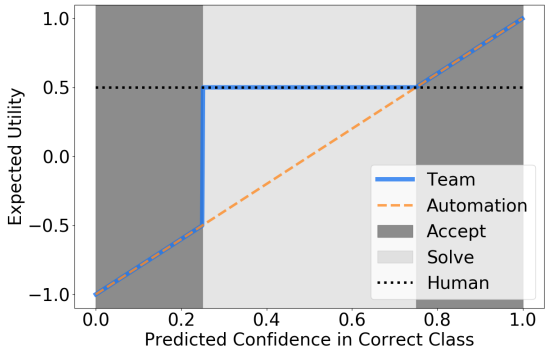

Figure 3: Visualization of expected utility when $\lambda = 0.5, \beta = 1$, and $a = 1$ (*i.e.*, the human is perfectly accurate but it costs them half a unit of utility to solve the task). In the `Accept` region, expected utility of the team is equal to expected utility of the automation, while in the `Solve` region it equals to the human utility. The negative team utility in the left-most region results from over-confident but incorrect recommendations to the human.

Expected Team Utility

We now derive the equation for expected utility of recommendations for the teamwork that we described above. Let ψ denote the expected team utility on a given example.

$$\begin{aligned}\psi(x,y) &= \mathbb{E}[U(m,d)]\\ &= P(m=\texttt{A})\cdot\Big[P(d=y|m=\texttt{A})\cdot 1\\ &\quad + P(d\neq y|m=\texttt{A})\cdot(-\beta)\Big]\\ &\quad + P(m=\texttt{S})\cdot\Big[P(d=y|m=\texttt{S})\cdot(1-\lambda)\\ &\quad + P(d\neq y|m=\texttt{S})\cdot(-\beta-\lambda)\Big]\end{aligned}$$

Since upon `Accept`, the human returns the classifier's recommendation, the probability that the final decision is correct is the same as the classifier's predicted probability of the correct decision, *i.e.*, $P(d = y|m = \texttt{A}) = h(x)[y]$. Substituting this and Equation 2 we obtain:

$$\begin{aligned}\psi(x,y) &= P(m=\texttt{A})\cdot\Big[(1+\beta)\cdot h(x)[y] - \beta\Big]\\ &\quad + P(m=\texttt{S})\cdot\Big[(1+\beta)\cdot a - \beta - \lambda\Big]\\ &= P(m=\texttt{A})\cdot\Big[(1+\beta)\cdot(h(x)[y]-a) + \lambda\Big]\\ &\quad + \underbrace{\Big[(1+\beta)\cdot a - \beta - \lambda\Big]}_{\text{constant}}\end{aligned}$$

Substituting human policy (Equation 4) we obtain:

$$\psi(x,y) = \begin{cases}(1+\beta)\cdot h(x)[y] - \beta & \text{if } h(x)[\hat{y}] \geq c(\beta,\lambda,a)\\ (1+\beta)\cdot a - \beta - \lambda & \text{otherwise}\end{cases} \tag{5}$$

Figure 3 visualizes the expected team utility of the classifier predictions as a function of confidence in the true label.

Dataset	#Features	Size	Frac. Pos.
Scenario1	2	10000	0.43
Moons	2	10000	0.50
German	24	1000	0.30
Fico	39	9861	0.52
Recidivism	13	6172	0.46
MIMIC	714	21139	0.13

Table 2: Number of features and size of binary classification datasets used for experiments. The original Fico dataset contains 23 features but 39 after preprocessing categorical features into binary features.

We convert expected utility into a loss function by negating it, *i.e.*, $-\psi(x, y)$.

3 Experiments

Experiments in this section address the following questions:

RQ1 Can we train a classifier with higher utility than the most accurate classifier?

RQ2 How does the new model qualitatively differ from the most accurate model?

RQ3 How do the properties of the task affect improvements in utility (*e.g.*, human skill and cost of mistake)?

Datasets We experimented with two synthetic datasets and four real-world binary classification datasets: German credit lending dataset (Hofmann 1994), FICO credit risk assessment (Fico 2018), recidivism prediction (ProPublica 2016), and MIMIC-3 mortality prediction (Harutyunyan et al. 2019). The real datasets are drawn from high-stakes domains where machine learning has already been deployed or has been discussed being employed to assist human decision makers. On the synthetic datasets, Scenario1 dataset refers to a dataset we created by sampling 10000 points from the data distribution similar to Figure 1. Moons refers to the classic two moons non-linear classification problem.[1]

Model Training We experimented with two types of models: logistic regression and multi-layered perceptron (two hidden layers with 50 and 10 units). For each task (defined by a choice of task parameters, dataset, model, and loss) we optimized the loss using the Adam optimizer and also used standard, well-known training practices such as regularization, check-pointing the model best validation performance, and learning rate schedulers. We selected the best hyperparameters using five-fold cross validation, including values for the learning rate, batch size, patience, decay factor of the learning rate scheduler, and the L2 regularization weight. (Range of parameters detailed in the Appendix).

In initial experiments to optimize team utility, we observed that the classifier's loss (in this case, negative of expected utility) remained constant over the optimization process. This happened because, in practice, random initializations resulted in classifiers that were uncertain on most of the data distributions considered. By definition, the expected utility is flat and constant in regions of uncertainty (see Figure 3). Thus, the gradient was zero and uninformative over these ranges. To overcome this issue, we initialized the classifiers with the (already converged) most accurate classifier.

Metrics: Empirical and Expected Utility We evaluated our systems on two metrics of team utility: expected team utility (Equation 5) and empirical team utility, which draws discrete rewards from the pay-off described in Table 1. A key difference between expected and empirical utilities is that the former incentivizes systems that output a calibrated belief, *i.e.*, in the `Accept` region it assigns a score proportional to the system's confidence in the correct class (Figure 3). Empirical utility, in contrast, does not differentiate between a low- and a high-confidence recommendation in the `Accept` region as long as they are both correct (or both are incorrect).

Each metric offers different advantages. Maximizing empirical utility aligns well with existing non-probabilistic discrete metrics for evaluating ML classifiers (such as, accuracy, F1-score, and AUPRC), which exclusively focus on the discriminative power of models. In contrast, maximizing expected utility is critical for decision making under uncertainty, *i.e.*, when the outcome of decisions may be probabilistic and thus a rational agent should maximize for its decision's expected utility. In fact, the primary result of utility theory, the accepted, normative theory of action under uncertainty, is that ideal decisions are those that maximize expected utility (Morgenstern and Von Neumann 1953). Maximizing expected utility requires the use of calibrated probabilities, which is an aspect that is not reflected in empirical utility. Moreover, expected utility optimization is useful in cases when empirical evaluation of metrics is not feasible due to delayed reward in the real world or when the definition of empirical ground truth labels is soft and non-discrete.

Results

RQ1: Table 3 shows that the new classifier can improve expected team utility over log-loss. These improvements are often achieved by sacrificing the classifier's individual accuracy. For example, on Scenario1 the new linear classifier improved expected utility from 0.524 to 0.606 even though it was less accurate.

When we considered *empirical* utility, our method did not always result in improvements. For example, for the linear classifier, while on Scenario1, the empirical utility increased from 0.593 to 0.654, but on MIMIC it decreased from 0.8 to 0.765. Ideally, one would expect that an increase in expected team utility would be accompanied with proportional increase in empirical team utility. However, as Table 3 shows, this was often not the case.

While this mismatch between empirical and expected utilities seems counterintuitive, it is a well know problem; Huang et al. (2019) noticed a mismatch between various common ML evaluation metrics, such as log-loss, zero-one loss, and AUPRC. However, we still considered the possibility that, in practice, the mismatch perhaps resulted from stochastic optimization getting stuck in local minimas, and that a better optimization procedure would alleviate this mismatch. To pursue this conjecture, we developed two-

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html

		Logloss			Expected Utility Loss		
Classifier	**Dataset**	**Accuracy**	**Expected Util.**	**Emp. Util.**	**Δ Accuracy**	**Δ Expected Util.**	**Δ Emp. Util.**
Linear	Fico	0.729	0.487	0.575	-0.247	**0.013**	-0.075
	German	0.754	0.529	0.594	-0.015	0	-0.019
	MIMIC	0.881	0.694	0.8	-0.004	**0.066**	-0.035
	Moons	0.885	0.687	0.79	-0.02	**0.079**	-0.006
	recidivism	0.669	0.485	0.52	-0.17	**0.015**	-0.02
	Scenario1	0.858	0.524	0.593	-0.165	**0.102**	**0.061**
MLP	Fico	0.725	0.472	0.574	-0.244	**0.028**	-0.074
	German	0.752	0.53	0.618	-0.036	-0.027	-0.056
	MIMIC	0.881	0.719	0.799	-0.001	**0.049**	-0.029
	Moons	1	0.944	0.989	0	**0.049**	**0.006**
	Recidivism	0.674	0.467	0.521	-0.168	**0.033**	-0.021
	Scenario1	1	0.826	0.854	-0.1	**0.08**	**0.057**

Table 3: Comparison of accuracy, expected and empirical team utilities of classifiers optimized for log-loss (with a checkpoint on accuracy) and expected team utility (with a checkpoint on expected utility) using Adam for $\lambda = 0.5$, $a = 1.0$, $\beta = 1.0$. Observations averaged over 50 train/test splits. Δ indicates difference with respect to log-loss. Classifier trained to optimize expected team utility achieves higher expected utility at the cost of automation accuracy. However, we notice a mismatch between expected and empirical utilities– empirical utility *decreased* even though expected utility increased.

Dataset	**Expected Util** LL	**Emp. Util** LL	**ΔExpected Util (A)**	**Δ Emp. Util (B)**	**Δ^* Emp. Util (C)**
Fico-2d	0.475	0.511	0.025	-0.011	-0.004
German-2d	0.514	0.6	0.076	-0.004	-0.016
MIMIC-2d	0.641	0.772	0.121	-0.009	0.005
Moons	0.767	0.813	0.016	-0.006	0.034
Recidivism-2d	0.478	0.518	0.022	-0.017	0.007
Scenario1	0.707	0.715	0.045	0.069	0.068

Table 4: Test performance of linear classifier that optimizes log-loss and team utility using brute-force optimization on two-dimensional domains. While we observe consistent improvements in the team's expected utility (column marked A) across domains, improvements in expected utility did not translate to improvements in *empirical utility* (values in column marked B are negative), indicating a mismatch between the expected and empirical metrics of team utilities. At the same time, exhaustive search shows existence of linear classifiers with higher empirical utility (column marked C). Values were averaged over five seeds. Observations in column C on Fico-2d and German-2d were negative on test set due to over-fitting.

dimensional versions of our dataset (by selecting two top most informative features) and trained linear classifiers using exhaustive search, which by definition cannot get stuck in local minimas. We again found a persistence of the mismatch between expected and empirical utilities (Table 4). In addition, we also noticed that there exist classifiers with higher empirical utility if the exhaustive search maximizes directly for empirical utility (column C in Table 4), which further demonstrates the existence of the mismatch. [2]

These results provide evidence that the challenge with achieving comparable increases in empirical utility to those in expected utility is not only due to optimization issues (*e.g.*, local minimas and plateaus due to flatness of the expected utility curve in the `Solve` region). There exists a fundamental ML challenge of loss-metric mismatch, which was prominent in our setup. In the rest of the section, we present further analyses of improvements in the normative decision making metric of expected utility, which as described earlier, is useful in decision-making under uncertainty.

RQ2: While the metrics in Table 3 (change in accuracy and utility) provide a global understanding of the classifier behavior, here we attempt to understand *how* these improvements were achieved and whether the behavior of the new models is consistent with the original intuition. Figure 4 displays the difference in behavior (averaged over 50 seeds) between the classifiers produced by log-loss and the one that maximizes team utility on the Scenario1 and MIMIC dataset. Specifically, as shown in Figure 4, we visualize and compare the following behaviors of the two classifiers:

V1. Calibration using *reliability curves*, which compare system confidence and its true accuracy. A perfectly calibrated system, for example, will be 80% accurate on regions that is 80% confident. However, in practice, systems may be over- or under-confident.

V2. Distributions of confidence in predictions. For example, in Figure 4, the new classifier makes more high-confidence predictions than the most accurate classifier.

V3. Density of system accuracy as function of confidence in true label. Thus, the area under this curve indicates the system's total accuracy. Note that, for our setup, the area under the curve in the `Accept` region is more crucial.

V4. Density of expected utility as a function of confidence.

The classifier optimized for the team's expected utility re-

[2]Note that directly optimizing for empirical utility is not effective via stochastic optimization.

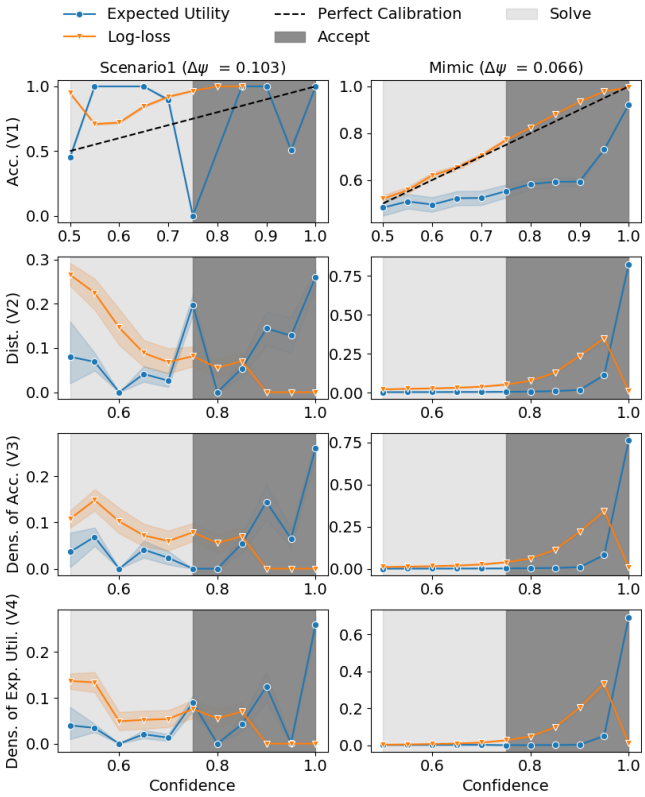

Figure 4: Behavior of linear classifiers that optimize log-loss and expected team utility on the Scenario1 and MIMIC datasets (observations averaged over 50 runs). The latter makes fewer predictions in the `Solve` region and also sacrifices accuracy in that region to increase it in `Accept`. We observed a behavior similar for the MLP model on all datasets (ommitted due to space constraints).

sults in dramatically different predictions than the classifier trained using log-loss: The new classifier sacrifices accuracy on the uncertain examples (`Solve` region) to make higher numbers of high-confidence predictions (`Accept` region). Most importantly, it also increases the density of system accuracy in the `Accept` region, which is where the system accuracy matters and contributes to team utility. Figure 4 illustrates the same behavior on the MIMIC in-hospital mortality prediction dataset.

An interesting exception was Fico, where the system learned to always be uncertain. This may make sense for the Fico domain because, as shown in Table 3, even though the most accurate linear classifier is 73% accurate on Fico, it achieves an expected team utility of 0.487. This is less than the expected utility achieved if humans solved the task alone. Hence, the more accurate classifier leads to lower expected team utility. We observed a similar behavior on recidivism prediction where the linear classifier led to team performance lower than that associated with people making decisions unaided, even though the classifer had a 67.4% accuracy (Table 3). These cases illustrate timely concerns and questions of when and if an AI should be deployed to assist human decision-making, which we further discuss in the ethical statement.

dataset	**a=0.8**	**a=0.9**	**a=1**
Fico	0.257 (0.133)	0.337 (0.071)	0.487 (0.013)
German	0.397 (0.046)	0.444 (0.035)	0.529 (0)
MIMIC	0.625 (0.127)	0.644 (0.111)	0.694 (0.066)
Moons	0.582 (0.162)	0.616 (0.139)	0.687 (0.079)
Recidivism	0.155 (0.073)	0.292 (0)	0.485 (0.015)
Scenario1	0.224 (0.324)	0.364 (0.248)	0.524 (0.102)

Table 5: Expected utility of log-loss and improvements for linear classifiers (Δ Expected Util. shown in brackets) with varying human accuracy (a) and ($\lambda = 0.5$ and $\beta = 1.0$). Results averaged over 50 random seeds. Improvements in expected utility are higher when the human is less accurate.

dataset	β**=1**	β**=3**	β**=5**
Fico	0.487 (0.013)	0.474 (0.026)	0.481 (0.019)
German	0.529 (0)	0.427 (0.057)	0.367 (0.118)
MIMIC	0.694 (0.066)	0.58 (0.008)	0.543 (0)
Moons	0.687 (0.079)	0.637 (0.065)	0.594 (0.085)
Recidivism	0.485 (0.015)	0.495 (0.004)	0.498 (0.001)
Scenario1	0.524 (0.102)	0.501 (0.02)	0.5 (0)

Table 6: Expected uility of log-loss and improvements for linear classifiers (*i.e.*, Δ Expected Util., shown in brackets) with varying cost of mistakes (β) and ($\lambda = 0.5, a = 1.0$). Results averaged over 50 random seeds. On most datasets, gains diminish as the cost of mistakes increases.

RQ3: Since properties such as the accuracy of users and penalty of mistakes may be task-dependant (*e.g.*, an incorrect diagnosis may be costlier than incorrect loan approval), we varied human accuracy a and mistake penalty β to study the sensitivity in improvements in team utility to a wider range of these task parameters.

Table 5 shows improvements in expected utility as we vary human accuracy from 80% to 100% while keeping λ and β constant to 0.5 and 1, respectively. These three values of a result in three new values of optimal threshold $c(\beta, \lambda, a)$: 0.55, 0.65, and 0.75, thus gradually expanding the confidence region in which the user is likely to `Solve` because they themselves are more accurate. We notice higher improvements in expected utility from deploying a system when humans are less accurate, *e.g.*, Table 5 shows that, on Fico, improvement in expected utility is 0.133 when the human is 80% accurate whereas it is 0.013 when they are perfect. One explanation for this behavior is that when humans are less accurate there is greater value from system recommendations, which widens the `Accept` region and increases the scope where the AI can provide value to the team.

Similarly, Table 6 shows the impact of varying cost of mistakes β on improvements. The three values of β increase the `Accept` threshold gradually from 0.75 to 0.91, and therefore shrink the size of the `Accept` region. Hence, we start observing smaller gains when the cost of mistake is high, *e.g.*, on the MIMIC dataset there are no gains, although the trend is also subject to the shape of expected utility and how easy it is to optimize it. In overall, the trend emphasizes once

again that for extremely high-stake decisions, automation or AI recommendation may not always provide value.

4 Discussion and Future Work

Implications for complex human-AI teams While we investigated a simplified human-AI teamwork (as defined in Section 2), our setup allows extensions to more complex team and users. For example, one can relax our assumption that users are rational by modifying the human-policy in Equation 4, so that when the prediction confidence is greater than the threshold, the user chooses `Accept` with probability $p < 1$, instead of 1.0. Here, $1 - p$ denotes the probability of the user being irrational — assessed from historical data, if available. Similarly, in more complex situations users may make `Accept` and `Solve` decisions using a richer, more complex mental model instead of relying on just model confidence. Such scenarios are common in cases where the system confidence is an unreliable indicator of performance (*e.g.*, due to poor calibration), and, as a result, the user develops an understanding of system failures in terms of domain features. For example, Tesla drivers may learn to override the Autopilot considering such features as road, sun glare, and weather conditions. We can reduce the case where users have a complex mental model to the policy that we studied. Specifically, we can construct a new loss function in terms of human utility (in this case, constant) when the prediction belongs to the `Solve` region (as described by the user's mental model) and automation utility otherwise.

While the above extensions to our model are a start, even they may present challenges– If we cannot optimize empirical utility for our simplified case, it may be harder to optimize performance in the extensions as they contain more complex user behavior and the resultant loss surface is likely to be more complex, containing combinations of plateaus and local optima. In addition to these extensions, future work should also consider more general uses of AI recommendations in support of human decision making. For example, we need to consider common uses that are not constrained to policies where a user either accepts an AI recommendation or relies completely on their own reasoning. It is natural to expect that users in human-AI teams will employ their own *evidential reasoning* to fuse AI inferences (and associated confidences if shared) with their own assessments. Furthermore, user's mental models may not be static; instead, they may change with time as users learn more about the AI. Mental models may also vary across users, as different people might have different propensity to accept machine recommendations.

Human-subject evaluations are an important next step to understand how factors such as biases, variations in user expertise, irrational behavior come to play in practice. Is our simple model of human behavior sufficient for our approach to yield gains in practice? We view our work as a fundamental first step showing the potential impact of a human-centered model and motivating additional work including real-world studies with human subjects. Over time, we hope to learn and incorporate rich (and individualized) models of human behavior into our framework and test them in real-world human-AI teams.

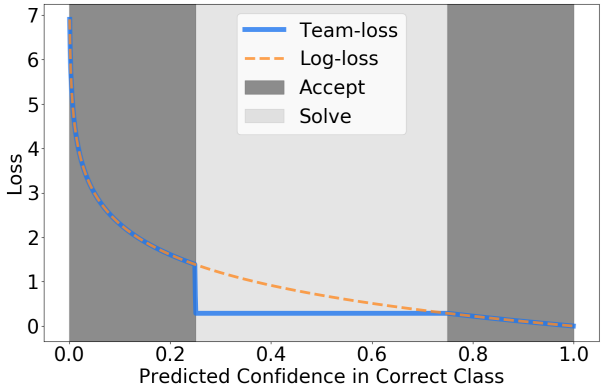

Figure 5: An example of an auxiliary loss function Team-loss defined as $= log(\psi(x, y) + K)$, which is equal to Log-loss in `Accept` region and constant otherwise. Here, K is a positive constant we added so that the logarithmic is valid.

Empirical utility and auxiliary loss functions While optimizing for teamwork, we faced two fundamental optimization challenges. First, we observed an an inherent mismatch between empirical and expected utility as shown in the exhaustive experiments for two dimensional data, which hindered optimization on the empirical metric, which is often a central consideration in ML. Second, current optimization techniques were not always effective and in fact sometimes they did not change model behavior because the optimization approaches got stuck due to zero gradients and local minimas `Solve` region.

To support empirical utility maximization, in our initial analysis, we also experimented with an auxiliary loss function, as shown in Figure 5. However, in our experiments this loss function did not lead always lead to significant gains in empirical utility and when it did it only lead to marginal improvements. Based on these theoretical and practical challenges, we invite future work on machine learning optimization and human-AI collaboration to develop new optimization techniques that work well beyond , robustly over a more general set of loss of functions that can capture team utility.

Mental models and explainable AI To increase team performance we focused on adapting one AI property to user mental models– the AI should be more accurate on instances when the user is more likely to trust model recommendations. Similarly, future work should study whether other aspects of human-AI collaboration can be improved by considering user mental models. For instance, mental models could help inform the nature of explanation given to the users. Such as, when users already trust the model, the system may be better off offering concise explanations. In contrast, when the users are likely to distrust the model, they system should be ready to offer detailed explanations/arguments supporting its prediction (Bansal et al. 2020). Eventually, work on explainable AI aims to improves human-AI collaboration by providing a layer of communication between users and AI

systems. Since explainability does not guarantee improvements in collaboration (Bansal et al. 2020), there is need to bring collaboration as an objective to every step of system development, starting from the training objective. We hope that this work paves the work for future directions towards uncovering how to develop AI systems for collaboration.

5 Related Work

Our approach is closely related to *maximum-margin classifiers*, such as an SVM optimized with the hinge loss (Burges 1998), where a larger soft margin can be used to make high-confidence and accurate predictions. However, unlike our approach, it is not possible to directly plug the domain's payoff matrix (*e.g.*, in Table 1) into such a model. Furthermore, the SVM's output and margin do not have an immediate probabilistic interpretation, which is crucial for our problem setting. One possible (though computationally intensive) solution direction is to convert margin into probabilities, *e.g.*, using post-hoc calibration (*e.g.*, Platt scaling (Platt 1999)), and use cross-validation for selecting margin parameters to optimize team utility. While it is still an open question whether such an approach would be effective for SVM classifiers, in this work we focused our attention on gradient-based optimization.

Another related problem is *cost-sensitive learning*, where different mistakes incur different penalties; for example, false-negatives may be costlier than false-positives (Zadrozny, Langford, and Abe 2003; Bach, Heckerman, and Horvitz 2006). A common solution here is up-weighting the inputs where the mistakes are costlier. Also relevant is work on *importance-based learning* where re-weighting helps learn from imbalanced data or speed-up training. However, in our setup, re-weighting the inputs makes less sense— the weights would depend on the classifier's output, which has not been trained yet. An iterative approach may be possible, but our initial analysis showed this approach is prone to oscillations. We leave exploring this avenue for future work.

A fundamental line of work that renders AI predictions more actionable (for humans) and better suitable for teaming is *confidence-calibration*, for example, using Bayesian models (Ghahramani 2015; Beach 1975; Gal and Ghahramani 2016) or via *post-hoc* calibration (Platt 1999; Zadrozny and Elkan 2001; Guo et al. 2017; Niculescu-Mizil and Caruana 2005). A key difference between these methods and our approach is that *team-loss re-trains* the model to improve on inputs on which users are more likely to rely on the AI predictions. The same contrast distinguishes our approach from *outlier detection* techniques (Hendrycks, Mazeika, and Dietterich 2018; Lee et al. 2017; Hodge and Austin 2004).

Closely related is research on *learning to defer* (Madras, Pitassi, and Zemel 2018; Mozannar and Sontag 2020) and *learning to complement* (Wilder, Horvitz, and Kamar 2020), where the classifier can abstain and defer/query the task to the user, while accounting for costs and benefits of intervention. While the "Solve" meta-decision in our framework corresponds to the defer action, our work differs from these works in two important ways. First, the defer action in prior work is system-initiated whereas in our case it is user-initiated and based on their mental model. Second, learning to defer does not preclude our methods, since users may create mental models even when the system does not defer and so the team may still benefit from training a model that accounts for user's mental model.

Other recent work that adjusts model behavior to accommodate collaboration includes *backward-compatibility for AI* (Bansal et al. 2019b), where the model considers user interactions with a previous version of the system to preserve trust across updates. Recent user studies showed that when users develop mental models of AI system, properties besides accuracy are also desirable, such as *parsimonious* and *deterministic* error boundaries (Bansal et al. 2019a). Our approach is a first step towards implementing these desiderata within ML optimization itself. Other approaches regularize or constrain model optimization for other human-centered requirements such as local- or global-interpretability (Wu et al. 2020) or fairness (Jung et al. 2019; Zafar et al. 2017).

6 Conclusions

We studied opportunity to train classifiers that optimize human-AI team performance. We showed the value of optimizing the expected utility of decision making of human-AI teams in contrast to traditional model optimization focusing solely on automation accuracy. Investigations and visualizations of classifier behavior before and after proposed optimization show that the methods can be harnessed to fundamentally change model behavior and improve the team utility. Changes in model behavior include (i) sacrificing model accuracy in low confidence regions for more accurate high-confidence predictions and (ii) increasing accuracy and number of high-confidence predictions. Such behaviors were observed in both synthetic and real-world datasets where AI is known to be employed as support for human decision makers, and across various domain parameters such as human accuracy and cost of mistake.

Ethical Statement

A broader contribution of this work is to rethink how ML models are defined and optimized when they are deployed in human-AI collaboration scenarios, *e.g.*, for supporting human decision making in high-stakes areas (including healthcare and criminal justice) where AI systems already influence user decisions with important consequences for individuals and society. Since most AI systems are optimized automation performance, more research is needed to create effective advisory systems by integrating team-centered considerations in the formal machinery of optimization used to build and execute these AI systems. We examined one approach to raising the expected value of AI-aided human decision making by considering teamwork in the optimization objective.

Beyond the direct use of the methods for optimizing human-AI teamwork, the methods can be valuable for building insights on teaming. For example, results showed that there exist regions in the space of collaboration parameters where automated assistance, or even providing an AI recommender, may not make sense– when the cost of mistakes was high and human decisions are sufficiently accurate, the algorithm always hands over control to humans, discouraging the need for algorithmic support. Such an analysis highlights the importance of carefully questioning and evaluating whether AI deployment is beneficial from a team perspective. A more rigorous evaluation requires robust and online estimation of costs and user behavior to ensure that the training and real-world objectives align. While we did not address the problem of estimating and updating such parameters, we wish to bring attention to the fact that problems such as underestimation of costs (or overestimation of rewards) may still lead to high-cost mistakes even when following the optimization approach we proposed in this paper. We hope that advances in interdisciplinary research on measuring the impact and costs in socio-technical systems will further inform decisions and designs about the role and behavior of AI in human-AI teamwork in future work.

Finally, we recognize that significant ethical issues are raised by the nature of human oversight and agency over AI in our simplified human-AI teaming. Our formulation of user policy assumed that, when the AI system is confident, the user completely trusts AI inferences and forgoes further human deliberation. Such a policy can lead to inappropriate transfers of responsibility in realistic settings. Even if the model is confident and has been historically correct, humans will still need to stay cognizant of the potential for poorly characterized and unexpected modes of AI failure, *e.g.*, due to distributional shifts or changes in the influences of latent variables with changes in context or workload. Thus, in real-world settings, the policy that we studied can be dangerous. In uses of AI, where high-confidence recommendations are typically trusted and there is a practice of little or no human deliberation about the validity of automated output, human overseers of AI should be aware of their reliance and their need to take full accountability for outcomes linked to the inferences.

7 Acknowledgments

This material is based upon research initially performed during Gagan Bansal's summer internship at Microsoft Research, with continuing support by ONR grant N00014-18-1-2193, NSF RAPID grant 2040196, the University of Washington WRF/Cable Professorship, and the Allen Institute for Artificial Intelligence (AI2). The authors thank Zeyuan Allen-Zhu, Rich Caruana, Bryan Wilder, and the anonymous reviewers for helpful comments.

References

Bach, F. R.; Heckerman, D.; and Horvitz, E. 2006. Considering cost asymmetry in learning classifiers. *JMLR* .

Bansal, G.; Nushi, B.; Kamar, E.; Lasecki, W. S.; Weld, D. S.; and Horvitz, E. 2019a. Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance. In *HCOMP*.

Bansal, G.; Nushi, B.; Kamar, E.; Weld, D. S.; Lasecki, W. S.; and Horvitz, E. 2019b. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *AAAI*.

Bansal, G.; Wu, T.; Zhou, J.; Fok, R.; Nushi, B.; Kamar, E.; Ribeiro, M. T.; and Weld, D. S. 2020. Does the Whole Exceed its Parts? The Effect of Explanations on Complementary Team Performance. *ArXiv* URL https://arxiv.org/abs/2006.14779.

Beach, B. H. 1975. Expert judgment about uncertainty: Bayesian decision making in realistic settings. *Organizational Behavior and Human Performance* 14(1): 10–59.

Burges, C. J. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2): 121–167.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*.

Fico. 2018. Explainable Machine Learning Challenge. https://community.fico.com/s/explainable-machine-learning-challenge.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 1050–1059.

GDPR. 2020. Art. 22 GDPR, Automated individual decision-making, including profiling. https://gdpr-info.eu/art-22-gdpr/. [Online; accessed 14-January-2020].

Ghahramani, Z. 2015. Probabilistic machine learning and artificial intelligence. *Nature* 521(7553): 452.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *ICML*.

Harutyunyan, H.; Khachatrian, H.; Kale, D. C.; Ver Steeg, G.; and Galstyan, A. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific Data* 6(1): 96. ISSN 2052-4463. doi:10.1038/s41597-019-0103-9. URL https://doi.org/10.1038/s41597-019-0103-9.

Hendrycks, D.; and Gimpel, K. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR* .

Hendrycks, D.; Mazeika, M.; and Dietterich, T. G. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606* .

Hodge, V.; and Austin, J. 2004. A survey of outlier detection methodologies. *Artificial intelligence review* 22(2): 85–126.

Hofmann, H. 1994. Statlog (German Credit Data) Data Set. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

Horvitz, E.; Heckerman, D.; Nathwani, B.; and Fagan, L. 1986. The use of a heuristic problem-solving hierarchy to facilitate the explanation of hypothesis-directed reasoning. In *Proceedings of Medinfo, Washington, DC*, 27–31.

Huang, C.; Zhai, S.; Talbott, W.; Bautista, M. A.; Sun, S.-Y.; Guestrin, C.; and Susskind, J. 2019. Addressing the loss-metric mismatch with adaptive loss alignment. *arXiv preprint arXiv:1905.05895* .

Jung, C.; Kearns, M.; Neel, S.; Roth, A.; Stapleton, L.; and Wu, Z. S. 2019. Eliciting and Enforcing Subjective Individual Fairness. *arXiv preprint arXiv:1905.10660* .

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS*.

Lee, K.; Lee, H.; Lee, K.; and Shin, J. 2017. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325* .

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *NeurIPS*, 4765–4774.

Madras, D.; Pitassi, T.; and Zemel, R. 2018. Predict responsibly: improving fairness and accuracy by learning to defer. In *NeurIPS*.

Morgenstern, O.; and Von Neumann, J. 1953. *Theory of games and economic behavior*. Princeton university press.

Mozannar, H.; and Sontag, D. 2020. Consistent Estimators for Learning to Defer to an Expert. In *ICML*.

Nagar, Y.; and Malone, T. 2011. Making business predictions by combining human and machine intelligence in prediction markets. In *International Conference on Information Systems*. Association for Information Systems.

Nickelsburg, M. 2020. Washington state passes landmark facial recognition bill, reining in government use of AI. https://www.geekwire.com/2020/washington-state-passes-landmark-facial-recognition-bill-reining-government-use-ai/.

Niculescu-Mizil, A.; and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *ICML*.

Patel, B. N.; Rosenberg, L.; Willcox, G.; Baltaxe, D.; Lyons, M.; Irvin, J.; Rajpurkar, P.; Amrhein, T.; Gupta, R.; Halabi, S.; et al. 2019. Human–machine partnership with artificial intelligence for chest radiograph diagnosis. *NPJ digital medicine* 2(1): 1–10.

Platt, J. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* 10(3): 61–74.

ProPublica. 2016. How We Analyzed the COMPAS Recidivism Algorithm. https://github.com/propublica/compas-analysis.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *KDD*.

Weld, D. S.; and Bansal, G. 2019. The Challenge of Crafting Intelligible Intelligence. *CACM* 62: 70–79.

Wilder, B.; Horvitz, E.; and Kamar, E. 2020. Learning to Complement Humans. In *IJCAI*.

Wu, M.; Parbhoo, S.; Hughes, M.; Kindle, R.; Celi, L.; Zazzi, M.; Roth, V.; and Doshi-Velez, F. 2020. Regional Tree Regularization for Interpretability in Black Box Models. In *AAAI*.

Zadrozny, B.; and Elkan, C. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *ICML*, volume 1, 609–616. Citeseer.

Zadrozny, B.; Langford, J.; and Abe, N. 2003. Cost-Sensitive Learning by Cost-Proportionate Example Weighting. In *ICDM*, volume 3, 435.

Zafar, M. B.; Valera, I.; Rodriguez, M. G.; and Gummadi, K. P. 2017. Fairness constraints: Mechanisms for fair classification. In *AISTATS*.