Misaligned by Design: Incentive Failures in Machine Learning*

David Autor†, Andrew Caplin‡, Daniel Martin§, Philip Marx¶

November 11, 2025

Abstract

The cost of error in many high-stakes settings is asymmetric: misdiagnosing pneumonia when absent is an inconvenience, but failing to detect it when present can be life-threatening. Because of this, artificial intelligence (AI) models used to assist such decisions are frequently trained with asymmetric loss functions that incorporate human decision-makers' trade-offs between false positives and false negatives. In two focal applications, we show that this standard alignment practice can backfire. In both cases, it would be better to train the machine learning model with a loss function that ignores the human's objective and then adjust predictions ex post according to that objective. We rationalize this result using an economic model of incentive design with endogenous information acquisition. The key insight from our theoretical framework is that machine classifiers perform not one but two incentivized tasks: *choosing* how to classify and *learning* how to classify. We show that while the adjustments engineers use correctly incentivize choosing, they can simultaneously reduce the incentives to learn. Our formal treatment of the problem reveals that methods embraced for their intuitive appeal can in fact misalign human and machine objectives in predictable ways.

*We thank the Sloan Foundation for its support under the "Cognitive Economics at Work" grant. We thank Annie Liang, Preston McAfee, Ashesh Rambachan, Larry Samuelson, Jakub Steiner, Eva Tardos, and participants in the USC Conference on AI, Economics, and Business and the SMART symposium at the University of Zurich for valuable comments.

†Massachusetts Institute of Technology, Google Technology and Society Fellows program, and NBER.

‡New York University and NBER.

§University of California, Santa Barbara.

¶Louisiana State University.

1 Introduction

As AI-based systems are increasingly deployed to make high-stakes decisions and execute consequential tasks autonomously, it is critical that they are 'aligned' with the intentions of their human deployers (Bostrom 2014; Bengio et al. 2023; Ji et al. 2023; Anwar et al. 2024). If we take alignment at its simplest level to mean that systems are trying to behave as intended by their human deployers[1], alignment seems readily achievable in straightforward cases. Consider, for example, using a machine learner to detect potential cases of pneumonia from chest X-rays. While even the best human or machine classifier will misclassify some cases, the cost of errors is asymmetric: misdiagnosing pneumonia when it is absent (a false positive) inconveniences the patient; failing to diagnose pneumonia when it is present (a false negative) potentially threatens the patient's life. Accordingly, an aligned AI should behave like an ethical physician by erring on the side of overdiagnosing pneumonia. If medical ethics dictate that false negatives are 99 times as costly as false positives, it stands to reason that providing a machine learner with a loss function that penalizes false negatives 99 times as heavily as false positives should align the machine's actions with the deployer's intentions.

Following this logic, machine learning models are frequently trained with asymmetric loss functions that codify experts' assessed costs of false positives relative to false negatives.[2] Implicit in these adjustments is what we term the *aligned learning premise* (ALP): using the human's objective to train a machine learning model produces better performance in terms of that objective because it allows the human's objective to inform what the machine learns.

We show empirically that the ALP is false in two focal applications. In both cases, one would do better to first train the machine learning model using a standard loss function that ignores the human's objective and then adjust predictions ex post according to the human's objective, rather than to train with a *utility-weighted* loss function that accounts for the human's objective, even though both loss functions are smooth and convex, which allows for optimization procedures to work effectively. In other words, trying to bake utility weights into training makes predictions worse — even when judged by the utility-weighted objective itself.

These applications cover two standard prediction problems and algorithm architectures: medical diagnosis from chest X-rays using deep neural networks (Rajpurkar, Irvin, Zhu, et al. 2017) and image classification in the CIFAR benchmark dataset using transformers (Dosovitskiy et al. 2021). Previewing our results for pneumonia detection from chest X-

[1] This is the definition of *intent alignment* (Christiano 2018).

[2] For example, such asymmetric training for classifiers may be implemented through class weighting in the loss function or weighted resampling of the training data.

rays, Figure 1 plots the weighted loss for the same model under two possible loss functions: an unweighted one that does not account for the human's objective and one that weights instances of pneumonia relative to other classes at a ratio of 99 to 1.[3] In red are the weighted losses from training with the weights that reflect the human's objective (*Weighted Training*).[4] The ALP would suggest that this approach should produce the best performance one could achieve for the human's objective. However, in blue are the average weighted losses if we do not account for the human's objective when training the machine learning model, but instead account for utility by transforming predictions according to the desired objective (*Ex Post Weighting*). Comparing these series shows that the machine minimizes utility-weighted losses not when trained with the utility-weighted loss function, but rather when trained without such weighting. In this pneumonia detection task, this dominance holds at every training epoch and across every run.

We rationalize these ALP failures using an economic model of incentive design with endogenous information acquisition. In our theoretical framework, alignment is achieved if the AI system makes predictions that maximize the human's expected utility. If the AI can learn perfectly (acquire a perfectly informative information structure), then the human only needs to adjust AI predictions ex post to achieve maximal expected utility. However, if the AI cannot learn perfectly, achieving alignment can depend on how the human designs the AI system.

One core design decision is which loss function to use when training the AI, and the ALP suggests that the optimal course of action is to base that decision on the human's own utility function. Viewing this as an incentive design problem, we ask if the human's utility function provides the correct incentives to the machine learner. The key insight provided by our model is that machine learners are performing not one but two incentivized tasks: choosing how to classify and learning how to classify.[5] When the machine is *choosing* how to classify a given X-ray, its loss function should guide it to output false positives 99 times as often as false negatives. Asymmetric weighting accomplishes this goal. But what incentives should the machine be given when *learning* to classify X-rays? Intuition might suggest that learning is not an incentive problem: the machine should simply learn as effectively as possible. But the mathematics of machine learning dictate otherwise. Conventional machine

[3]As we discuss in Section A.2, this weighting could be motivated by the importance of avoiding false negatives, issues of class imbalance in training data, or both.

[4]The dark dots in each series correspond to the lowest utility-weighted loss across training epochs for the average run. In each case, the best results are obtained within the first ten epochs, suggesting the result is not attributable to underfitting and would be robust to variation in stopping rules.

[5]Prominent classifier architectures that are separable in this way include regression trees (Breiman et al. 1984), traditional neural networks (e.g., LeCun et al. 1998), and recent transformers (Vaswani et al. 2017, Dosovitskiy et al. 2021)

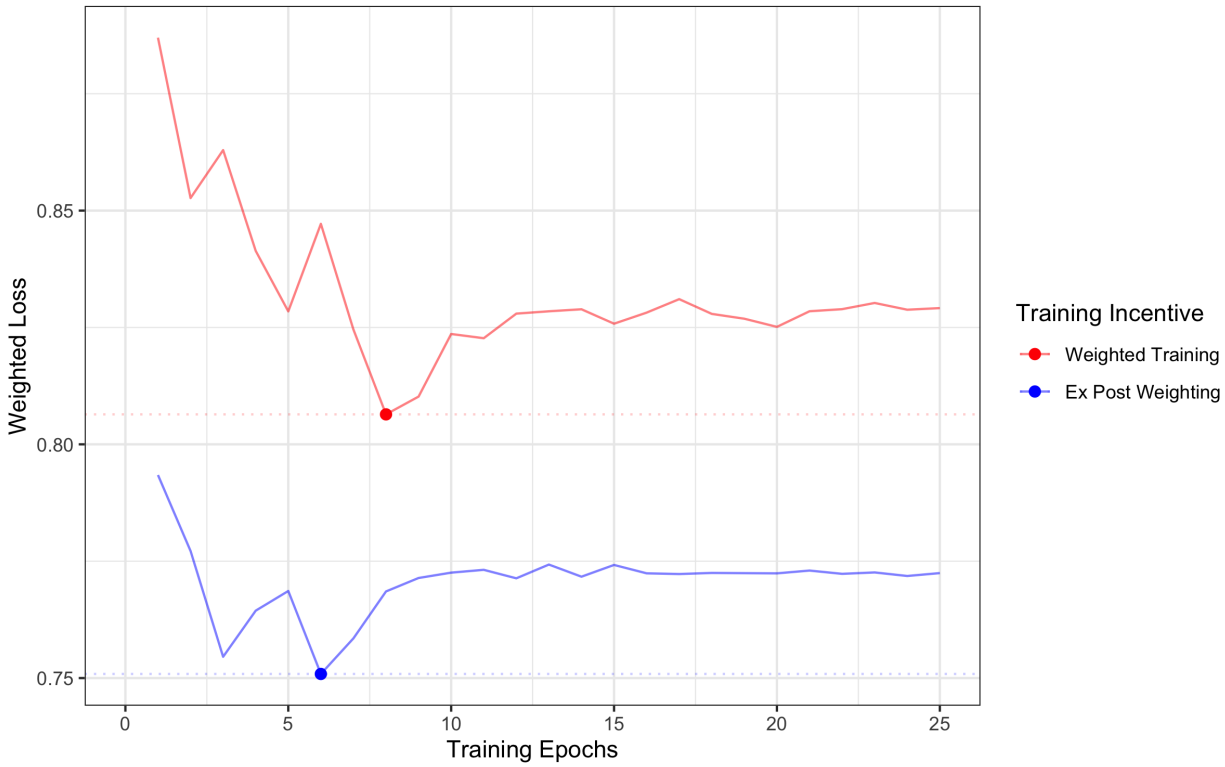

Figure 1: Weighted loss in the test sample across training incentives by training epoch, averaged across five runs. In red are the average weighted losses from training with the weights that reflect the human's objective, and in blue are the average weighted losses from training without weights, but accounting for the weights by transforming predictions ex post. Dots represent the minimum training epoch and dotted lines the corresponding weighted loss.

learning algorithms learn to map features into classes through the process of gradient descent. Because the machine's loss function dictates the shape of the gradient, it necessarily shapes the machine's incentives for learning. The learning problem is therefore an incentive problem.

Why does the human's objective not correctly incentivize the machine's learning problem? Formally speaking, why would the human's objective incentivize the machine to choose a poorly fitting information structure? We show theoretically that making a loss function asymmetric to account for the human's objective can backfire by weakening the machine learner's payoff to substantive learning. Accounting for optimal ex-post adjustments in our theoretical and empirical results allows us to neutralize the impact of incentives for choosing and focus attention on the incentives for learning.

Figure 2 illustrates the forces behind our main theoretical results using the case of binary classification, such as classifying a patient with pneumonia. The left panel shows the machine's optimal prediction for each posterior probability of the "positive" class (e.g., pneumonia).

Clearly, a machine that is incentivized to weight instances of pneumonia relative to other classes at a ratio of 99 to 1 would inflate predictions of pneumonia. The middle panel then shows how inflating predictions impacts the marginal incentives to learn: that is, the incentives to improve the posterior probability of the positive class. By inflating all predictions, the value of additional learning at all intermediate levels of learning is strongly dampened. Dampening the marginal incentives then lowers the overall incentive for learning—shown in the right panel—which is lower for all probabilities of the positive class. In summary, utility weighting provides correct incentives for choosing (left panel) but reduces the implied value of learning (middle and right panels), thus unintentionally misaligning human and machine objectives.

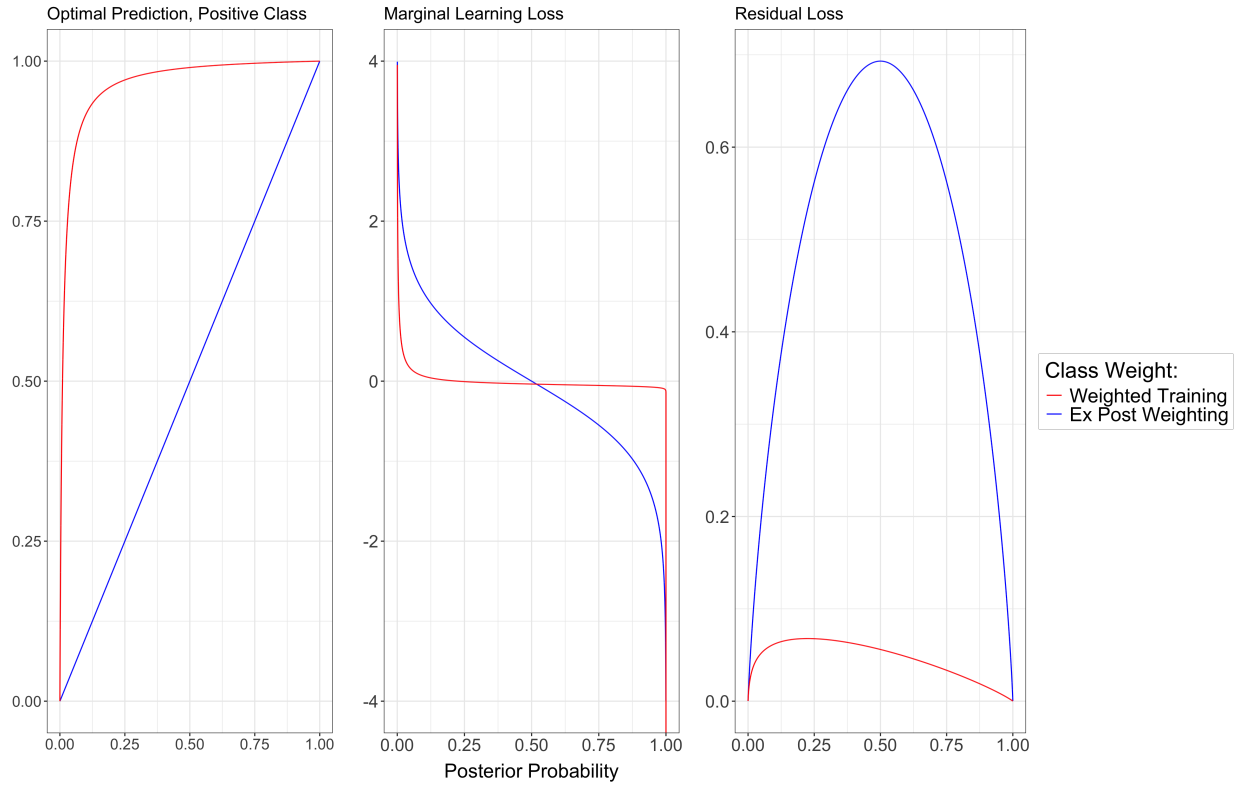

Figure 2: Incentives to choose and learn in the case of unweighted and weighted binary classification. Class weighting incentivizes distorting predictions as a function of posterior probabilities (left panel). This distorts the marginal benefit of learning such probabilities (center panel), lowering and distorting the overall incentives for learning (right panel).

The widespread use of utility-weighted loss functions reflects both precedent and plausibility: they have delivered performance gains on benchmark datasets, and they mirror the trade-offs faced by decision-makers in domains like medicine or finance. Yet despite their popularity, these methods rest on an incomplete theoretical foundation. In particular, there has been no systematic treatment of how such practices interact with the incentives underlying machine learning, nor how they may alter the value of information acquisition. By providing an

economic framework, we show that methods embraced for their intuitive appeal can in fact misalign human and machine objectives in predictable ways.

Our primary contribution is to illuminate the centrality of incentives in aligning machine behavior with human intentions. Using tools of economics, researchers have begun to consider how best to align machine outputs to human objectives, a process referred to as "algorithmic design" (Liang et al. 2021) and "welfare-aware machine learning" (Rolf et al. 2020). Liang et al. (2021) consider the role of data inputs in achieving human fairness, and Rolf et al. (2020) consider how to account for competing human objectives in machine learning. We add to this literature by showing how incentives play a central role in aligning machine outputs to human objectives. In this, we relate to previous work by Hummel and McAfee (2017), who consider the design of the training loss function based on downstream economic incentives.

Our paper also contributes to a literature that connects economic theory and machine learning (see Liang 2025 for a review). For instance, Samuelson and Steiner (2024) and Aridor, Grechi, and Woodford (2020) use variational encoders to model how humans with cognitive constraints learn from the world. Our paper inverts this lens to investigate what models of human learning can tell us about how machines learn. That is, we explicitly model the machine's incentives and information acquisition as we would for a human. This approach links to a recent incentive design literature that considers how to best incentivize *human* learners with positive learning costs (Lambert 2019; Camara, Immorlica, and Lucier 2025).

By taking a Bayesian learning approach, our paper also connects machine learning with the information design literature (Kamenica and Gentzkow 2011; Kamenica 2019). Specifically, we model the machine learner as first forming a probability distribution over classes and then selecting a prediction based on that distribution. We emphasize that this modeling approach is *as if* because the machine learner does not necessarily follow such a process. Nevertheless, this abstraction allows us to connect what a machine learns with the classical definition and ordering of Blackwell (1953).

Finally, the separation we make between AI predictions and final decisions appears often in the literature on fairness in AI (e.g., Kleinberg, Ludwig, Mullainathan, and Sunstein 2018). While the problem studied in that literature is different (balancing outcomes across groups), our result is spiritually related to a robust finding in that literature, which is that it is better in terms of fairness concerns to train an unconstrained predictor and then post-process those decisions to make decisions that balance outcomes across protected categories (e.g., Corbett-Davies et al. 2017; Menon and Williamson 2018; Lipton, McAuley, and Chouldechova 2018; Kleinberg, Ludwig, Mullainathan, and Rambachan 2018; Rambachan et al. 2020).

The rest of the paper proceeds as follows. Section 2 provides our theoretical framework and results. Section 3 provides the empirical results from our applications to chest X-ray diagnosis using deep neural networks and CIFAR image detection using transformers. Section 4 discusses how our methods and findings relate to machine learning literatures on cost-sensitive learning and alignment. Section 5 concludes.

2 Theoretical Framework and Results

This section formalizes the relationship between human objectives and machine learning. By framing the design of a machine's loss function as an incentive design problem, we show why intuitive practices like utility-weighted loss functions can systematically backfire and, more fundamentally, uncover the two incentivized tasks of the machine: choosing how to classify a set of inputs given what it has learned; and learning from the inputs that it is given. While asymmetric weighting correctly shapes incentives for choice, we show that it can inadvertently weaken incentives for learning, leading to worse outcomes on the objectives that weighting was meant to serve. The framework that follows establishes this result formally and provides the foundation for the empirical applications that follow.

2.1 Illustrative Example

Imagine that a machine learning engineer (ME) needs to develop an *AI system* to provide advice (an action a) based on an X-ray (a set of features x) that shows pneumonia or not (a class y). The ME wants the AI system to give "good" advice; that is, advice that is appropriate given the disease state. The ME's preferences for advice could reflect the preferences of the users of advice (doctors), the purchaser of the AI system (hospital), and/or the ME's employer (software company).

As illustrated in Figure 2.1, the AI system the ME develops can be summarized as a function d from X-rays to advice. The ME trains an *AI model* to be the key element in this AI system. The AI model outputs a probability of pneumonia based on an X-ray, and the AI model is summarized by a prediction function f. As a final step, the ME adds post processing δ to map the AI model output to specific advice a.

In practice, the ME has to make many choices when training the AI model, but we focus on a particularly important one, which is the loss function ℓ used when training the AI model. In this setting, alignment is achieved if the ME's choice of the loss function ℓ produces an AI model f, which when combined with the optimal post processing δ, generates an AI system

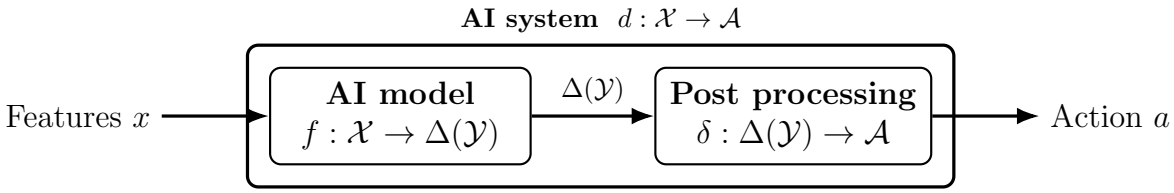

Figure 3: The components of the AI system.

d that maximizes the ME's expected utility over advice and disease states given the set of possible X-rays.

2.2 Human Decisions and Machine Predictions

Formalizing this problem, human preferences are based on a finite set of actions $\mathcal{A} \equiv \{1, \ldots, n\}$ (e.g., types of advice) and a finite set of *classes* $\mathcal{Y} \equiv \{1, \ldots, m\}$ (e.g., one class might be pneumonia and another might be no disease). These preferences are summarized by a nonnegative utility function $u : \mathcal{A} \times \mathcal{Y} \to \mathbb{R}_+$. Our formal results require that the utility function u is *nondegenerate*, meaning that for every class there is some action that yields positive utility:[6]

$$\forall y \in \mathcal{Y} : \exists a \in \mathcal{A} : u(a, y) > 0 \tag{1}$$

Human decisions are based on a set of features $\mathcal{X}$ (e.g., image pixels). The human decision-maker's problem is to develop an AI system $d : \mathcal{X} \to \mathcal{A}$ (shown in Figure 2.1) to maximize expected utility $\mathbb{E}[u(d(X), Y)]$.[7] The expectation operator is defined with respect to a probability distribution $\mathbb{P}$, which is a random vector of features and classes (X, Y) with realizations (x, y) and finite support $\mathcal{X} \times \mathcal{Y}$. We interpret this probability distribution as an idealized infinite dataset with irreducible error.

To help develop this AI system, the human decision-maker employs machine learning to generate an AI model, which is a probabilistic prediction model. In the machine learning literature, this stage in developing an AI system is referred to as "training." We focus on a particular aspect of training, which is the human decision-maker's choice of the machine's *loss function* $\ell : \Delta(\mathcal{Y}) \times \mathcal{Y} \to \mathbb{R}$ over the set of classes $\mathcal{Y}$ and the set of probability distributions $\Delta(\mathcal{Y})$ over those classes. The human's ability to dictate the machine's loss function has

[6]This assumption is innocuous because affine transformations of the utility function preserve the same expected utility preferences, and so a utility function can always be transformed by addition of a positive constant to satisfy nondegeneracy.

[7]Preferences can equally be summarized by a utility function to be maximized or a cost function to be minimized, and we consider the cost-based approach in Appendix A to facilitate comparison with the existing machine learning literature.

motivated a variety of approaches in the machine learning literature on cost-sensitive learning and classification, which we discuss in Section 4 and detail in Appendix A.

Our economic approach considers the human's choice of loss function as an incentive design problem for a downstream agent, the machine, which learns imperfectly. Based on the incentives provided by this loss function, the machine's problem is to find an AI model $f : \mathcal{X} \to \Delta(\mathcal{Y})$ to minimize expected loss $\mathbb{E}[\ell(f(X), Y)]$.

Finally, the human decision-maker combines the machine's AI model f with post processing $\delta : \Delta(\mathcal{Y}) \to \mathcal{A}$ to create the AI system $d : \mathcal{X} \to \mathcal{A}$. In the machine learning literature, this stage in developing the AI system is referred to as "inference." For an AI system that generates advice, post processing could involve simple recalibration of machine outputs (Guo et al. 2017; Caplin, Martin, and Marx 2022a) or a calibrating coarsening of the machine outputs (Hoong and Dreyfuss 2025). For an AI system that is a final decision-maker, post processing could be a threshold rule that produces a treatment decision.

It is worth noting that the post-processing rule is only based on the probability distribution over classes. For many algorithms, this distribution is generated by applying the soft-max function to raw outputs. In principle, the designer could base the post-processing rule on other outputs from the machine, and it would be possible to extend our framework to allow for post processing to use an arbitrary space of latent representations.[8]

2.3 Alignment

With this framework in place, we can formally define two forms of alignment. *External alignment* is achieved if the loss function produces an AI model, which in combination with a decision rule, yields an AI system that maximizes the human's expected utility. *Internal alignment* is achieved if the loss function provided to the machine produces an AI model which maximizes that loss function.

We focus our alignment framework on the common case of classification, such as diagnosing whether a patient has pneumonia. In this setting, the action is a class $\mathcal{A} = \mathcal{Y}$ and post processing is a classification rule $\delta : \Delta(\mathcal{Y}) \to \mathcal{Y}$. We focus on this case because it allows the connection between human preferences and machine learning to be stated as directly as possible, and it allows for simple expressions of theoretically optimal post processing. Nevertheless, our approach and results generalize.

[8]We feel this is an interesting avenue for future work, and we thank Jakub Steiner for raising this point.

2.4 Optimal Post Processing

We first use this framework to formally define optimal post processing for standard machine learning loss functions. Given a probability distribution $q \in \Delta(\mathcal{Y})$ over classes, the two key objects in this approach are the expected loss $\bar{\ell}$ of machine prediction $p \in \Delta(\mathcal{Y})$ and the expected utility $\bar{u}$ of human decision $y \in \mathcal{Y}$:

$$\bar{\ell}(p, q) \equiv \langle \ell(p, \cdot), q \rangle \tag{2}$$

$$\bar{u}(y, q) \equiv \langle u(y, \cdot), q \rangle \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and $\ell(p, \cdot)$, $u(a, \cdot)$ denote vectors over their second argument. For example, letting p_y and q_y denote the y-th elements of p and q, $\ell(p, y) = -\log p_y$ is the logistic loss underlying cross-entropy loss $\bar{\ell}(p, q) = -\sum_{y \in \mathcal{Y}} q_y \log p_y$.

For now, we consider a machine that has been given a loss function ℓ that is *strictly proper* (Buja, Stuetzle, and Shen 2005; Gneiting and Raftery 2007), as with most standard loss functions (i.e., cross entropy and mean squared error).[9] For such a loss function, the optimal prediction is to output the true probability of each class:

$$q = \underset{p \in \Delta(\mathcal{Y})}{\operatorname{argmin}} \ \bar{\ell}(p, q) \quad \text{for all } q \in \Delta(\mathcal{Y}). \tag{4}$$

As a result, optimal predictions will be *calibrated*: when the model assigns a given probability to a class, that probability matches the observed frequency of the class. When predictions are calibrated, the post processing that accounts for the human's preferences is the one that maximizes expected utility given the true probability of each class:

$$\delta^u(q) \equiv \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \ \bar{u}(y, q) \tag{5}$$

2.5 Changing the Machine's Loss Function

Next we show how the machine's loss function can be designed to account for the human decision-maker's preferences. For this, we adopt an *as-if* approach where we decompose the machine learner's prediction problem into two steps: (i) generating a probability distribution q, and (ii) converting those probabilities into optimal utility-weighted predictions $p^u(q)$. This approach is "as if" in the sense that the machine learner need not follow such a two-step

[9] We do not consider the problem of how to select among proper scoring rules for prediction; for such results, see Buja, Stuetzle, and Shen (2005) and also Hummel and McAfee (2017) for an economically motivated approach in the context of online advertising auctions.

procedure — it only needs to generate predictions to optimize its objective.

We formalize *utility-weighted loss* $\ell^u : \Delta(\mathcal{Y}) \times \mathcal{Y} \to \mathbb{R}$ as:

$$\ell^u(p, y) \equiv \langle \ell(p, \cdot), u(\cdot, y) \rangle \tag{6}$$

Conditional on a probability distribution q, the utility-weighted prediction p^u minimizes expected utility-weighted loss:

$$p^u(q) \equiv \underset{p \in \Delta(\mathcal{Y})}{\operatorname{argmin}} \ \ell^u(p, q) \tag{7}$$

Our main theoretical result establishes that utility-weighted loss provides the "correct" incentives for the machine when combined with optimal post processing, which is a simple argmax classification rule.

Theorem 1 (Optimal Utility-Weighted Prediction)**.** *Suppose that u is nonnegative and nondegenerate and ℓ is a strictly proper loss function. For any class distribution $q \in \Delta(\mathcal{Y})$, the unique optimal utility-weighted prediction (7) for each class y is:*

$$p^u_y(q) = \frac{\bar{u}(y, q)}{\sum_{y' \in \mathcal{Y}} \bar{u}(y', q)} \tag{8}$$

Theorem 1 clarifies how to modify the machine's loss function to obtain a normalized expected utility (3) irrespective of the probability distribution $q \in \Delta(\mathcal{Y})$. Consequently, preference-aligned decisions are obtained by combining a utility-weighted loss with a classification rule that selects the class with the highest normalized utility:

$$\delta(p^u) \equiv \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \ p^u_y \tag{9}$$

2.6 Learning Identity

The preceding analysis establishes how human preferences can be embedded into the machine prediction problem via the utility-weighted loss function (6) for utility-weighted predictions given a probability distribution over classes (Theorem 1). However, our analysis is thus far silent about what the machine is incentivized to learn.

We now consider how the utility-weighted loss function may *implicitly* misalign incentives to learn. To do so, we adopt the classical approach of Blackwell (1953) that operationalizes the value of learning about classes through the distribution over probability distributions over

classes.

Treating predictor $f : \mathcal{X} \to \mathcal{Y}$ as fixed, let $P \equiv f(X)$ denote the random vector of resulting predictions and (P, Y) the corresponding prediction evaluation data given predictor f and raw data (X, Y). To make precise the problem decomposition and the meaning of "what the machine learns," we introduce a simple yet useful identity following from the law of iterated expectations.

Proposition 1 (Learning Identity)**.** *Given a prediction loss function ℓ and evaluation data (P, Y), the empirical risk can be expressed as:*

$$\mathbb{E}[\ell(P, Y)] = \mathbb{E}[\bar{\ell}(P, Q)] \tag{10}$$

where:

$$Q_y \equiv \mathbb{P}(Y = y|P) \quad \textit{for all } y \in \mathcal{Y} \tag{11}$$

is a random vector summarizing the information contained in the predictions P.

The novel object of Proposition 1 is Q, which is a random vector of posterior class distributions that summarizes the information about classes inherent in the predictions P given the actual outcomes Y (on the same probability space). To endow this information with meaning, recall that we implicitly assume the existence of some irreducible error.[10] In that case we refer to Q as a summary of *what is learned*; such a distribution of posterior probability distributions is a well-known classical means of conveying information content, with close ties to information value (Blackwell 1953).

Thus, Proposition 1 decomposes the machine's prediction objective (i.e., finding a predictor function f to minimize empirical risk) into an "as if" sequence of problems. The learning problem is to find an informative Q. Then, as a function of any realization q of what is probabilistically learned Q, the conditional prediction problem is to choose a prediction $p(q)$ to minimize the expected prediction loss $\bar{\ell}(p, q)$. Again, note that for calibrated predictions, what is learned is immediately given by the predictions themselves, $Q = P$. However, the value of expression (11) is that Q has meaning even when the predictor is systematically miscalibrated and $Q \neq P$, as arises by design in Theorem 1 when prediction losses are weighted by preferences.

This approach allows us to model and understand the (dis)incentives for learning through the (marginal) incentives for learning different distributions q over the class space.[11]

[10] In other words, we think of this as being evaluated in an uncountable test set, rather than a perfect fit in training with low external validity.

[11] Another potential comparative static is changes in distributions over probability distributions rather than

2.7 Implicit Learning Incentives

To understand how learning incentives are affected by human preferences u, we define the *(utility-weighted) residual learning loss* as the expected utility-weighted loss of probability distribution q, once it is optimally post processed according to (8):

$$\tilde{\ell}^u(q) \equiv \bar{\ell}^u(p^u(q), q) \tag{12}$$

The next result derives a simple expression for the *marginal* residual learning loss at probability distribution q as the vector of utility-weighted losses at the optimal prediction across classes.

Proposition 2 (Incentives Underlying Learning)**.** *As in Theorem 1, suppose that u is nonnegative and nondegenerate and ℓ is a strictly proper loss function. Then residual learning loss $\tilde{\ell}^u$ is a concave and differentiable function, and its gradient at each q is the vector of utility-weighted prediction losses evaluated at the optimal utility-weighted prediction. Expressed component-wise,*

$$\frac{\partial \tilde{\ell}^u}{\partial q_y}(q) = \ell^u(p^u(q), y) \tag{13}$$

2.8 Special Case: Class-Weighted Cross Entropy

The common case of class-weighted cross entropy is a specialization in two ways. First, human preferences are utility-weighted indicators for correct classification:

$$u^w(y', y) \equiv w_y 1\{y' = y\} \tag{14}$$

with $w \in \mathbb{R}^m_{++}$ ensuring nondegeneracy (1). This yields *class-weighted loss* $\ell^w(p, y) \equiv w_y \ell(p, y)$, which is a special case of utility-weighted loss. Second, the base loss function is taken to be the standard one for classification: logistic loss, $\ell(p, y) = -\log p_y$. Combining these specializations, weighted loss (6) becomes:

$$\ell^w(p, y) = -w_y \log p_y \tag{15}$$

The optimal class-weighted prediction (8) of class y given class distribution q becomes:

$$p^w_y(q) = \frac{w_y q_y}{\langle w, q \rangle} \tag{16}$$

changes in the probabilities themselves. However, this higher-order object also entails a much richer set of directional derivatives, and it would still be composed of these simpler constituent parts.

The gradient of learning loss (13) at probability distribution q becomes:

$$\frac{\partial \tilde{\ell}^w}{\partial q_y}(q) = -w_y \log p_y^w(q) \tag{17}$$

The gradient of learning losses (17) captures the direct and indirect effects of weighting on learning, assuming that the machine makes optimal predictions given what it has learned. Directly, a higher weight w_y on a class y increases the value of shifting probability distributions (i.e., learning) on that class. Indirectly, however, a higher weight w_y decreases the remaining term $-\log p_y^w(q)$ as the prediction $p_y^w(q)$ increases toward 1. Intuitively, as a class y becomes more heavily weighted, the corresponding prediction $p^w(q)$ becomes more detached from the probability distribution q. Namely, it becomes relatively more beneficial to make a high prediction on class y regardless of underlying information. In the limit case where w is only nonzero in its y-th element, it is optimal to output $p_y^w(q) = 1$ regardless of probability distribution q, so that learning is not valuable at all.

To develop further intuition for how the incentives for learning are affected by the choice of weights w, consider the binary-class case ($m = 2$). In that case, the probability distribution q, weights w, and optimal predictions $p^w(q)$ can be parameterized respectively by the scalar probability q_1, the scalar weight $w_1 = 1 - w_2$, and the scalar prediction $p_1^{w_1}(q_1)$ on class 1 (the positive class), with the analogs on class 2 being the complementary probabilities.[12] Then we can express the marginal learning loss $\tilde{\ell}^{w_1}(q_1)$ as:

$$\frac{d\tilde{\ell}^{w_1}}{dq_1}(q_1) = -w_1 \log(p_1^{w_1}(q_1)) + (1 - w_1) \log(1 - p_1^{w_1}(q_1)) \tag{18}$$

Returning to Figure 2 from the Introduction, we plot the optimal class-weighted prediction $p_1^{w_1}(q_1)$, the marginal learning loss $\frac{d}{dq_1}\tilde{\ell}^{w_1}(q_1)$, and the normalized learning loss $\tilde{\ell}^{w_1}(q_1)$ as a function of probability q_1 for two class weights: the "Ex Post Weighting" case $w_1 = w_2$ and a "Weighted Training" case that emphasizes the positive class $w_1 = 99w_2$. We call the former "Ex Post Weighting" because optimal post processing accounts for the human's utility weights.

The left-hand subfigure shows that emphasizing the positive class extremizes and thus flattens the optimal prediction as a function of the probability on much of the domain; in particular,

[12] The class weight normalization $w_1 + w_2 = 1$ in this theoretical subsection is scaled down by a factor of two relative to our normalization (21) in the experimental section 3. In the former case, intuition is simplified with complementary probabilities; in the latter case, rescaling relative to unweighted cross entropy (where weights sum to the number of classes m) disentangles the effect of reweighting on the shape of the objective function from effects on its magnitude.

it becomes optimal to output a high prediction on the weighted class except in cases of very low probabilities. The middle subfigure shows that when learning about the class has less bearing on the predictions, the marginal value of learning is decreased and even negligible on much of the domain. The right-hand subfigure shows that this flattens the learning loss.

This intuition extends beyond the binary-class case. The derivative (18) is a difference in the partial derivatives in the gradient (17), and this derivative is small because its constituent terms are small. More generally, suppose class 1 is (heavily) over-weighted. Then the partial derivative of $\tilde{\ell}^w(q)$ with respect to q_1 is low on most of the support because $p_1^w(q)$ is close to 1 regardless of the probability q, so that $\log p_1^w(q)$ is close to 0 and the marginal value of learning for losses is still small even though the weight w_1 is large; however, the partial derivatives on other classes are also low because the weights w_{-1} are small. In the extreme case where the loss function puts all weight on the predictions of class 1, there is no incentive to learn at all.

It is worth noting that while the incentive for learning appears suppressed for weighted learning in Figure 2 relative to unweighted learning, it turns out that marginal learning loss in Figure 2 is *higher* for extreme probabilities. Thus, it is conceivable that for settings where initial learning is very easy, weighted learning would provide optimal learning incentives.

It is also worth noting that it is not possible to overcome the dampened incentives for learning by uniformly increasing the size of the weights. In practice, a constant rescaling of the machine loss function would be offset by a corresponding change in the learning rate to avoid issues around numerical precision. Unlike human learners, the machine simply follows a path of stochastic gradient descent over a high dimensional surface toward a locally optimal solution. Thus, we interpret our analysis as illustrating distortions in this learning surface, which we would only expect to increase in more complex multi-class settings.

2.9 Establishing Misalignment

Let P and P^u denote the random vectors of predictions obtained through training with unweighted and utility-weighted loss functions, respectively. We now formalize both external and internal alignment within our framework.

Then we would conclude that incorporating human preference misaligns the machine output (according to internal alignment) if the optimal predictions from unweighted training outperform the optimal predictions from weighted training in terms of weighted losses:

$$\mathbb{E}[\ell^u(p^u(P), Y)] < \mathbb{E}[\ell^u(P^u, Y)] \tag{19}$$

Thus, we measure the strength of internal alignment as the gain (reduction) in weighted loss from using unweighted loss in training. Note that this test does not modify the utility-weighted predictions; rather it constructs a potential improvement from the unweighted predictions by transforming those according to the optimal utility-weighted prediction (8).

Analogously, we would conclude that incorporating human preferences misaligned the machine output (according to external alignment) if the corrected predictions from unweighted training outperform the weighted predictions in terms of human utility (here "classification utility"):

$$\mathbb{E}[u(\delta(p^u(P)), Y)] > \mathbb{E}[u(\delta(P^u), Y)] \tag{20}$$

Following the above, we measure the strength of external alignment as the gain (increase) in expected utility from using unweighted loss in training.

Next, we establish the presence of such misalignment in a variety of applications and show that it has consequences for the downstream classification objective.

3 Experiments

We document machine misalignment with human objectives by revisiting two prominent multi-class classification applications and architectures: chest X-ray diagnosis with deep neural networks (Rajpurkar, Irvin, Zhu, et al. 2017, Wang et al. 2017) and CIFAR image classification with vision transformers (Dosovitskiy et al. 2021, Krizhevsky and Hinton 2009). In each application, we focus on human utility functions (14) corresponding to class-weighted loss emphasizing one class y with a ratio $w_y = 99w_{-y}$ relative to any other class $-y$. In Appendix A, we explain how this can be motivated by either an enhanced importance of avoiding false negatives on that class (e.g., failing to detect pneumonia) or as a way of addressing class imbalance (e.g., pneumonia cases constitute approximately 1 percent of the training data in the application of Rajpurkar, Irvin, Ball, et al. 2018).[13] Across our applications, we normalize the weight vector w so that the expected loss of agnostically outputting the prior probability $\mu \in \Delta(\mathcal{Y})$ is fixed across training schemes and relative to unweighted loss:[14]

$$\bar{\ell}^w(\mu, \mu) = \bar{\ell}(\mu, \mu) \tag{21}$$

[13] In the imbalanced case of chest X-rays, we also consider training according to inverse probability-weighted cross entropy $w_y\mathbb{P}(y) = w_{y'}\mathbb{P}(y')$ for any classes y, y', where the probability weights are computed in the training data to avoid referencing test or validation data.

[14] In the balanced CIFAR data, such weights sum to the number of classes m; that is, their average value is one. The average value is not as easily interpretable in the imbalanced chest X-ray data.

This normalization is intended to hold fixed the initial expected loss — before the machine learns to discern images — and thus to ensure that our results are not driven by systematic differences in the magnitude of the gradient across weighting schemes or hyperparameters such as the learning rate. Besides our manipulation of the loss function, we follow standard parameter values and protocols. Each model is trained using an NVIDIA Tesla A100 80GB GPU on the Google Colab platform. We discuss further details and preexisting implementations in Appendix D.

We compare two training regimes: first, training according to class-weighted cross entropy (15) [*Weighted Training*], and second, training according to unweighted cross entropy and adjusting the predictions ex post according to (16) [*Ex Post Weighting*]. We compare these predictions according to the machine's own class-weighted loss in a test dataset to evaluate objective misalignment (19). In addition, we compare performance according to the human's classification utility (14). However, we would expect smaller differences in classification utility across regimes since categorical decisions will vary across training regimes less often than the underlying probabilistic predictions.

We also compare the training regimes across a variety of emphasized classes. In the chest X-ray task that uses the data from Wang et al. (2017), we separately study four classes with varying levels of occurrence in our training data (Pneumonia = 0.003, Cardiomegaly = 0.012, Pneumothorax = 0.024, and Infiltration = 0.105). In the image classification task using CIFAR data (Krizhevsky and Hinton 2009), we focus on emphasizing the most difficult class (cat images in CIFAR-10 and maple tree images in CIFAR-100) since the baseline predictor and classifier achieve near-perfect performance already on other classes, leaving little scope for potential performance improvements through aligned learning. In each variant, we repeat the training procedure five times to separate substantive (mis)alignment from the inherent stochasticity of the training procedures. In the chest X-ray tasks, we evaluate the trained models in the test data at the end of every training epoch, and in the CIFAR applications, we evaluate the trained model in the test data at every 100 training steps. Henceforth, we refer to the training epochs and training steps as *training intervals*.

3.1 Results

Table 1 summarizes the performance results across training procedures and applications in terms of weighted loss (used to measure internal misalignment) and classification utility (used to measure external misalignment). For each application, the table presents basic summary statistics (mean/min/max) of the optimal performance (over training intervals) in the test sample across five training runs. In every application, we find that the *Weighted Training*

	Weighted Loss			Classification Utility		
	Weighted Training	*Ex Post Weighting*	*% Gain*	*Weighted Training*	*Ex Post Weighting*	*% Gain*
Weight: Pneumonia (Chest X-ray)						
Mean	0.805	0.749	6.96	0.292	0.309	5.82
Min	0.796	0.743	6.66	0.284	0.296	4.23
Max	0.812	0.755	7.02	0.296	0.318	7.43
Weight: Cardiomegaly (Chest X-ray)						
Mean	0.401	0.383	4.49	0.333	0.333	0.00
Min	0.398	0.377	5.28	0.332	0.328	-1.20
Max	0.408	0.391	4.17	0.337	0.336	-0.30
Weight: Infiltration (Chest X-ray)						
Mean	0.214	0.211	1.40	0.597	0.597	0.00
Min	0.212	0.210	0.94	0.597	0.596	-0.17
Max	0.215	0.213	0.93	0.597	0.597	0.00
Weight: Pneumothorax (Chest X-ray)						
Mean	0.370	0.347	6.22	0.355	0.363	2.25
Min	0.360	0.343	4.72	0.353	0.360	1.98
Max	0.377	0.352	6.63	0.361	0.366	1.39
Weight: Inverse Probability (Chest X-Ray)						
Mean	0.723	0.696	3.73	0.125	0.130	4.00
Min	0.719	0.688	4.31	0.123	0.129	4.88
Max	0.729	0.703	3.57	0.127	0.131	3.15
Weight: Cat (CIFAR-10)						
Mean	0.033	0.024	27.3	0.988	0.989	0.10
Min	0.032	0.023	28.1	0.986	0.989	0.30
Max	0.035	0.025	28.6	0.989	0.990	0.10
Weight: Maple Tree (CIFAR-100)						
Mean	0.188	0.159	15.4	0.960	0.969	0.94
Min	0.181	0.156	13.8	0.955	0.968	1.36
Max	0.195	0.161	17.4	0.964	0.970	0.62

Table 1: Comparison of prediction and evaluation methods under different weighting schemes. Each row reports mean, min, and max losses across five runs. The column "% Gain" reports the percentage improvement of Ex Post Weighting relative to Weighted Training for each row. The gain is positive for improvement: lower weighted loss or higher classification utility.

procedure is outperformed on average by the *Ex Post Weighting* procedure of training without weights. Indeed, in nearly every application, the worst performance on the weighted training objective of *Ex Post Weighting* exceeds the best performance of *Weighted Training*. The one exception is Infiltration diagnoses in chest X-rays, which is also the case where the mean performance improvements of *Ex Post Weighting* are smallest.

Consider our leading application of pneumonia: across five runs, we consistently achieve better performance on the weighted training objective under the *Ex Post Weighting* procedure than under *Weighted Training*, with a mean reduction in loss of 6.96%.[15] Importantly for our alignment interpretation, our measure of outperformance is benchmarked to the *Weighted Training* objective.

These results also largely translate to external alignment, which relates to the human's utility (the downstream objective of maximizing the underlying classification utility), although as expected the differences are smaller and noisier given the discontinuous nature of the objective. Classification performance according to the *Ex Post Weighting* procedure performs either better or approximately the same on average (up to three decimal points, in the cases of Cardiomegaly and Infiltration diagnoses). In Appendix C we illustrate this classification performance improvement across training intervals in pneumonia diagnosis.

3.2 Discussion

In summary, we document consistent performance improvements on both the machine's own objective and the human's downstream classification utility when the machine is *not* trained according to the weighted objective but instead when its unweighted predictions are corrected ex post to accord with the performance objective. In Section 2, we presented a theoretical framework and arguments attributing this objective misalignment to implicit incentives that distort or even stifle the value of learning substantive information.

Our main findings are consistent with prior Gauss-Markov style results showing that, in the absence of heteroskedasticity, an unweighted objective and estimator are better than weighted alternatives (Greene 2012). This alternative explanation has some important shortcomings, however. Most critically, it would predict that unweighted learning should outperform weighted learning without ex post corrections. In Appendix C we also compare weighted loss and classification utility from *Weighted Training* to an unweighted training regime without modifying predictions (*Unweighted Raw*), and we find that *Weighted Training* significantly outperforms *Unweighted Raw* according to the weighted objectives.

[15] Computed as $((0.749 - 0.805)/0.805) \times 100\%$ in the first panel of Table 1.

4 Related Machine Learning Literature

Our paper seeks to bridge a longstanding literature on cost-sensitive learning and class imbalance with an active literature on the alignment of machine learning to human preferences. Our overarching objective is to illuminate the centrality of *incentives* in aligning machine actions with human intentions. By embedding choice incentives into learning, many common methods conflate the objectives of choosing and learning, thereby distorting one while aligning the other.

Our incentive-based approach is based on a novel cost-weighted prediction loss function and generalizes and unifies many of the existing solutions in the literature, including (partial or binary-class) solutions based on thresholding, reweighting, resampling, and base rate adjustments. Additionally, our incentive-based approach yields a general multi-class formula for analytical recalibration of cost-weighted predictions.

4.1 Cost-Sensitive Learning

Motivated by the suboptimal performance of standard classifiers in cases of cost-sensitive classification and class imbalance, the literature on cost-sensitive learning built on a series of workshops at the turn of the century, including a workshop on cost-sensitive learning at the 2000 International Conference on Machine Learning (ICML, Dietterich et al. 2000), a workshop on learning from imbalanced data at the 2000 Association for the Advancement of Artificial Intelligence (AAAI) meetings (Japkowicz 2000, Provost 2000, Japkowicz and Holte 2001), and a second workshop on learning from imbalanced data at the 2003 ICML (Chawla 2003; Drummond and Holte 2003; Maloof 2003; Chawla, Japkowicz, and Kołcz 2003). Fernández et al. 2018 provides a recent review. The challenge of cost-sensitive classification and its resolution are embodied in Elkan 2001: "the essence of cost-sensitive decision-making is that it can be optimal to act as if one class is true even when some other class is more probable."

Our multi-class incentive-based method grounded in (6) and the theory of proper scoring[16] generalizes and clarifies existing approaches in the literature, including reweighting training data (e.g., Breiman et al. 1984; Domingos 1999; Drummond and Holte 2000; Elkan 2001; Ting 2002; Zhou and Liu 2010), resampling training data (e.g., Kubat and Matwin 1997; Elkan

[16] Our approach departs from this literature by aggregating proper scoring evaluations according to misclassification costs to generate *im*proper scoring rules. In contrast, Buja, Stuetzle, and Shen 2005 *decomposes* proper scoring rules as weighted sums of cost-weighted misclassification errors to select among *proper* scoring rules. Their approach is in turn based on the proper scoring decompositions of Shuford, Albert, and Massengill (1966), Savage (1971), and Schervish (1989).

2001; Drummond and Holte 2003; Zadrozny, Langford, and Abe 2003; Abe, Zadrozny, and Langford 2004; Xia et al. 2009), and ex post prediction adjustments based on modified base rates (Elkan 2001; Saerens, Latinne, and Decaestecker 2002) or thresholding and conditional risk minimization (Domingos 1999; Elkan 2001; Margineantu 2002).[17] Our simple framework combines the generality of existing data-based resampling solutions with the efficiency of algorithm-based solutions. It also generalizes existing analytical recalibration approaches beyond the well-studied binary resampling case (Dal Pozzolo et al. 2015), and it makes precise the connections between cost-sensitive learning (i.e., incentives) and class imbalance (i.e., the distribution of classes and features in training data) in a general multi-class framework. We detail these relations in Appendix A.

Empirically, the literature has debated the utility of introducing costs into the machine's loss functions. For example, Elkan (2001) conjectures that introducing costs into the training objective may not significantly affect performance relative to introducing costs into the classification incentive, and Vanderschueren et al. (2022) conclude that it is more important that costs are included in the decision-making strategy than whether they are included in the training or classification stage. Our work is, to the best of our knowledge, the first to demonstrate that introducing such costs into the training objective may be counterproductive with respect to the terminal goal because it is counterproductive for the machine's own learning objective. Our theoretical framework and results provide the foundation to develop improvements by framing and decomposing the problem as one of incentive design (compared to, e.g., resampling strategies).

4.2 Alignment

As machine learning models have become more powerful and foundational, researchers have raised substantial concerns about the risks and possibilities of misalignment between human values and machine objectives (e.g., Bostrom 2014; Russell 2016; Amodei et al. 2016). As famously presaged by Wiener (1960): "If we use, to achieve our purposes, a mechanical agency with whose operation we cannot interfere effectively... we had better be quite sure that the purpose put into the machine is the purpose which we really desire." Seemingly the central challenge of alignment, then, is correctly and precisely communicating complex objectives about which we ourselves may not yet be sure. However, while aligned objectives are almost certainly necessary for human-machine alignment, another question is whether they are sufficient. Our results suggest that even a "correctly" specified human objective

[17] Similar strategies for designing cost-sensitive binary classifiers have also been considered in the econometric literature, e.g., Lieli and White (2010) and Elliott and Lieli (2013) and references therein.

may encode other, possibly pathological, implicit incentives into machine learning.

The goal of our method is to align the machine's prediction incentives with the human decision-maker's classification incentives. Our as-if modeling approach generates a natural separation into an *inner* and *outer* alignment problem, analogous to Hubinger et al. (2019). In the inner alignment problem, the model is incentivized to transform posterior class distributions into predictions that align with misclassification costs. In the outer alignment problem, the machine is incentivized to learn a mapping from features to (posterior distributions over) classes, given the transformation inherent in the inner alignment problem.

Our empirical and theoretical finding on calibration and learning incentives can be rephrased using this inner/outer alignment dichotomy: Given outer alignment, cost-weighting may still affect the shape of the inner objective in a way that softens incentives toward inner alignment, even though it does not distort those incentives. Conversely, alignment of the inner objective (cost-sensitive outputs) may induce pathological incentives in the outer objective (actual machine learning), such that the machine incentives that theoretically align with human objectives are *not* optimal for training the machine. In our case, the paradox is that even the machine would prefer — according to its objective function — to be trained according to other incentives, conditional on adjusting its outputs to reflect its incentives to choose. While this may be rationalized as a consequence of modifying the shape of the loss function in counterproductive ways, we reiterate that our loss function transformations at least preserve the smoothness and convexity properties that motivate surrogate loss. Furthermore, we would expect the possibility of inadvertent incentive spillover to become more severe in settings that are more complex or where even the inner objective is misaligned. Thus, we caution that not all human objectives may be productively encoded into the machine learning algorithm, even when they can be precisely articulated.

Analogously to Hubinger et al. (2019), our approach to the inner problem studies the model as if it were an optimizer, given posterior probabilities over classes; we do not, however, claim that the model in the inner problem is a *mesa* optimizer, which seems reserved for reinforcement learning, large language, or general foundation models with richer input and output spaces introducing the possibility of in-context learning and optimization. Still, cost-sensitive classification is perhaps a minimal extension of standard classification for which the task is complex enough for the two-stage analogy to be non-tautological,[18] yet simple enough where we can articulate the correct incentives for the machine, as well as the procedure for recovering the machine's behavioral objective or implicit preferences (analogous to inverse

[18]Tautological in their sense that any object's objective can always be defined as being and behaving like what it is.

reinforcement learning; Ng and Russell 2000). Since we analytically derive the inner as-if objective, we can thus conclude that the inner objective is essentially (albeit more softly) aligned upon using our approach, and show why this still expectedly creates challenges for aligning the outer objective. We are also able to use our analytical approach to define and recover latent but important attributes such as "what the machine has learned." This suggests the potential for as-if modeling approaches to complement deepened mechanistic intuition (e.g., Olsson et al. 2022; Oswald, Niklasson, et al. 2023; Oswald, Schlegel, et al. 2024) and pre-formal analysis (e.g., Ngo, Chan, and Mindermann 2024) for understanding, interpreting, and aligning ever-more-complex machine learning models with human preferences.

While our results are broadly consistent with the empirical findings of Vanderschueren et al. (2022) and Caplin, Martin, and Marx (2022a) in the case of binary classifiers, we document strict and significant suboptimality in the multi-class case, suggesting that implicit misalignment increases with the complexity of the prediction problem.

5 Conclusion

This paper examines a simple premise that underpins much current practice: if human decision-makers value some errors more than others, then training machine learners on a utility-weighted loss should better align model behavior with human objectives. We show empirically that this aligned learning premise (ALP) can fail in systematic ways. We explain this failure theoretically using economic principles of incentive design. Machine learners simultaneously face two incentivized tasks: *choosing* how to classify given what they know and *learning* what is worth knowing in the first place. While utility weighting can correctly incentivize the choice, it inadvertently weakens incentives for learning by flattening the value of additional information.

Our theoretical framework formalizes this separation. Modeling prediction as a two-step process, forming posterior probabilities and then mapping them to predictions, we derive the optimal way to embed human preferences into the prediction step (Theorem 1). We then show how this transformation changes the shape of the learning objective. Weighting inflates preferred classes, which then reduces the marginal value of moving posteriors in most regions of the state space (Proposition 2). In short, the same adjustment that aligns incentives for what to choose can misalign incentives for what to learn.

Across two standard applications — chest X-ray diagnosis with deep neural networks and CIFAR image classification with transformers — we find consistent evidence in favor of a simple alternative: train with a strictly proper, unweighted loss to learn calibrated probabilities, and

then impose human objectives ex post based on those objectives. This *Ex Post Weighting* approach dominates training directly on the utility-weighted objective (*Weighted Training*) when evaluated on that very objective, and it typically yields equal or better downstream classification utility.

Our analysis focuses on multi-class classification under proper losses and nondegenerate human utilities. It abstracts from dynamic, interactive, or sequential settings, and from additional issues such as fairness or robustness to distribution shift. Extending incentive design to these domains (e.g., by constructing training objectives that both preserve information incentives and respect complex human goals) is an important direction for future work.

The high-level message from this paper is that alignment should not focus exclusively on determining the human's objective; it must also provide the machine learner with the right incentives to acquire information. Recognizing both alignment objectives may prove crucial to building AI systems that are not, in effect, misaligned by design.

References

Abe, Naoki, Bianca Zadrozny, and John Langford (2004). "An iterative method for multi-class cost-sensitive learning". In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Alonso, Ricardo and Odilon Câmara (2016). "Bayesian persuasion with heterogeneous priors". In: *Journal of Economic Theory* 165, pp. 672–706.

Amodei, Dario et al. (2016). *Concrete problems in AI safety*. arXiv:1606.06565.

Anwar, Usman et al. (2024). "Foundational Challenges in Assuring Alignment and Safety of Large Language Models". In: *Transactions on Machine Learning Research*.

Aridor, Guy, Francesco Grechi, and Michael Woodford (2020). "Adaptive efficient coding: A variational auto-encoder approach". In: *bioRxiv*.

Bengio, Yoshua et al. (2023). "Managing AI risks in an era of rapid progress". In: *arXiv preprint arXiv:2310.17688*.

Blackwell, David (1953). "Equivalent comparisons of experiments". In: *Annals of Mathematical Statistics*, pp. 265–272.

Bostrom, Nick (2014). *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.

Breiman, Leo et al. (1984). *Classification and Regression Trees*. Wadsworth.

Brier, Glenn W. (1950). "Verification of forecasts expressed in terms of probability". In: *Monthly Weather Review* 78.1, pp. 1–3.

Buja, Andreas, Werner Stuetzle, and Yi Shen (2005). "Loss functions for binary class probability estimation and classification: Structure and applications". Working draft.

Camara, Modibo K., Nicole Immorlica, and Brendan Lucier (2025). "Eliciting informed preferences". In: *arXiv preprint arXiv:2505.19570*.

Caplin, Andrew, Daniel Martin, and Philip Marx (2022a). *Calibrating for class weights by modeling machine learning*. arXiv:2205.04613.

– (2022b). *Modeling Machine Learning*. Tech. rep. National Bureau of Economic Research.

Chawla, Nitesh V. (2003). "C4.5 and imbalanced datasets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure". In: *International Conference on Machine Learning (ICML)*.

Chawla, Nitesh V., Nathalie Japkowicz, and Aleksander Kołcz, eds. (2003). *Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Datasets*.

– (2004). "Editorial: Special issue on learning from imbalanced data sets". In: *SIGKDD Explorations* 6.1, pp. 1–6.

Christiano, Paul (2018). *Clarifying "AI alignment"*. Blog post, AI Alignment Forum.

Corbett-Davies, Sam et al. (2017). "Algorithmic decision making and the cost of fairness". In: *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pp. 797–806.

Dal Pozzolo, Andrea et al. (2015). "Calibrating probability with undersampling for unbalanced classification". In: *IEEE Symposium Series on Computational Intelligence*, pp. 159–166.

DeGroot, Morris H. and Stephen E. Fienberg (1983). "The comparison and evaluation of forecasters". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 32.1–2, pp. 12–22.

Deng, Jia et al. (2009). "ImageNet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255.

Dietterich, Thomas et al., eds. (2000). *Proceedings of the ICML 2000 Workshop on Cost-Sensitive Learning*.

Domingos, Pedro (1999). "Metacost: A general method for making classifiers cost-sensitive". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Dosovitskiy, Alexey et al. (2021). "An image is worth 16×16 words: Transformers for image recognition at scale". In: *International Conference on Learning Representations (ICLR)*.

Drummond, Chris and Robert C. Holte (2000). "Exploiting the cost (in)sensitivity of decision tree splitting criteria". In: *International Conference on Machine Learning (ICML)*.

– (2003). "C4.5, class imbalance and cost sensitivity: Why under-sampling beats over-sampling". In: *International Conference on Machine Learning (ICML)*.

Elkan, Charles (2001). "The foundations of cost-sensitive learning". In: *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 973–978.

Elliott, Graham and Robert P Lieli (2013). "Predicting Binary Outcomes". In: *Journal of Econometrics* 174.1, pp. 15–26.

Fan, Wei et al. (1999). "AdaCost: Misclassification cost-sensitive boosting". In: *International Conference on Machine Learning (ICML)*.

Fernández, Alberto et al. (2018). *Learning from Imbalanced Datasets*. Springer.

Gneiting, Tilmann and Adrian E. Raftery (2007). "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American Statistical Association* 102.477, pp. 359–378.

Greene, William H. (2012). *Econometric Analysis*. 7th. Upper Saddle River, NJ: Pearson Education.

Guo, Chuan et al. (2017). "On calibration of modern neural networks". In: *International Conference on Machine Learning (ICML)*, pp. 1321–1330.

Hiriart-Urruty, Jean-Baptiste and Claude Lemaréchal (2004). *Fundamentals of convex analysis*. Springer Science & Business Media.

Hoong, Ruru and Bnaya Dreyfuss (2025). "Improving AI-Assisted Decision-Making Through Calibrated Coarsening". In: *Available at SSRN 5286198*.

Huang, Gao et al. (2017). "Densely connected convolutional networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hubinger, Evan et al. (2019). *Risks from learned optimization in advanced machine learning systems*. arXiv:1906.01820.

Hummel, Patrick and R Preston McAfee (2017). "Loss Functions for Predicted Click-Through rates in Auctions for Online Advertising". In: *Journal of Applied Econometrics* 32.7, pp. 1314–1328.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International Conference on Machine Learning (ICML)*, pp. 448–456.

Japkowicz, Nathalie (2000). "The class imbalance problem: Significance and strategies". In: *Proceedings of the International Conference on Artificial Intelligence*.

Japkowicz, Nathalie and Robert C. Holte (2001). "Workshop report: AAAI 2000 workshop on learning from imbalanced datasets". In: *AI Magazine* 22.1, pp. 81–83.

Ji, Jiaming et al. (2023). "AI alignment: A comprehensive survey". In: *arXiv preprint arXiv:2310.19852*.

Kamenica, Emir (2019). "Bayesian persuasion and information design". In: *Annual Review of Economics* 11, pp. 249–272.

Kamenica, Emir and Matthew Gentzkow (2011). "Bayesian persuasion". In: *American Economic Review* 101.6, pp. 2590–2615.

Kingma, Diederik and Jimmy Ba (2015). "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*.

Kleinberg, Jon, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan (2018). "Algorithmic fairness". In: *Aea papers and proceedings*. Vol. 108. American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203, pp. 22–27.

Kleinberg, Jon, Jens Ludwig, Sendhil Mullainathan, and Cass R Sunstein (2018). "Discrimination in the Age of Algorithms". In: *Journal of legal analysis* 10, pp. 113–174.

Krizhevsky, Alex and Geoffrey Hinton (2009). "Learning multiple layers of features from tiny images". Technical Report, University of Toronto.

Kubat, Miroslav and Stan Matwin (1997). "Addressing the curse of imbalanced training sets: One-sided selection". In: *International Conference on Machine Learning (ICML)*.

Lambert, Nicolas S. (June 2019). "Elicitation and Evaluation of Statistical Forecasts". Working paper. Stanford, CA.

LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Leike, Jan et al. (2018). *Scalable agent alignment via reward modeling: A research direction.* arXiv:1811.07871.

Liang, Annie (2025). *Using machine learning to generate, clarify, and improve economic models.* arXiv:2508.19136.

Liang, Annie et al. (2021). "Algorithm design: A fairness-accuracy frontier". In: *arXiv preprint arXiv:2112.09975.*

Lieli, Robert P and Halbert White (2010). "The Construction of Empirical Credit Scoring Rules Based on Maximization Principles". In: *Journal of Econometrics* 157.1, pp. 110–119.

Lipton, Zachary, Julian McAuley, and Alexandra Chouldechova (2018). "Does mitigating ML's impact disparity require treatment disparity?" In: *Advances in neural information processing systems* 31.

Maloof, Marcus A. (2003). "Learning when data sets are imbalanced and when costs are unequal and unknown". In: *International Conference on Machine Learning (ICML).*

Margineantu, Dragos D. (2002). "Class probability estimation and cost-sensitive classification decisions". In: *European Conference on Machine Learning (ECML).*

Menon, Aditya Krishna and Robert C Williamson (2018). "The cost of fairness in binary classification". In: *Conference on Fairness, accountability and transparency.* PMLR, pp. 107–118.

Minderer, Matthias et al. (2021). "Revisiting the calibration of modern neural networks". In: *Advances in Neural Information Processing Systems.*

Naeini, Mahdi P., Gregory Cooper, and Milos Hauskrecht (2015). "Obtaining well-calibrated probabilities using Bayesian binning". In: *AAAI Conference on Artificial Intelligence.*

Ng, Andrew Y. and Stuart Russell (2000). "Algorithms for inverse reinforcement learning". In: *International Conference on Machine Learning (ICML).*

Ngo, Richard, Lawrence Chan, and Sören Mindermann (2024). "The alignment problem from a deep learning perspective". In: *International Conference on Learning Representations (ICLR).*

Niculescu-Mizil, Alexandru and Rich Caruana (2005). "Predicting good probabilities with supervised learning". In: *International Conference on Machine Learning (ICML)*, pp. 625–632.

Nixon, Jeremy et al. (2019). "Measuring calibration in deep learning". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).*

Olsson, Catherine et al. (2022). *In-context learning and induction heads.* Transformer Circuits Thread.

Oswald, Johannes von, Eyvind Niklasson, et al. (2023). "Transformers learn in-context by gradient descent". In: *International Conference on Machine Learning (ICML)*.

Oswald, Johannes von, Maximilian Schlegel, et al. (2024). *Uncovering mesa-optimization algorithms in transformers*. arXiv:2309.05858.

Platt, John (1999). "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in Large Margin Classifiers*. MIT Press, pp. 61–74.

Provost, Foster (2000). "Machine learning from imbalanced datasets 101". In: *AAAI 2000 Workshop on Imbalanced Datasets*, pp. 1–3.

Rajpurkar, Pranav, Jeremy Irvin, Robyn L. Ball, et al. (2018). "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists". In: *PLOS Medicine* 15.11, e1002686.

Rajpurkar, Pranav, Jeremy Irvin, Kaylie Zhu, et al. (2017). "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning". In: *arXiv preprint arXiv:1711.05225*.

Rambachan, Ashesh et al. (2020). *An economic approach to regulating algorithms*. Tech. rep. National Bureau of Economic Research.

Rockafellar, R. Tyrrell (1970). *Convex Analysis*. Princeton University Press.

Rolf, Esther et al. (2020). "Balancing competing objectives with noisy data: Score-based classifiers for welfare-aware machine learning". In: *International Conference on Machine Learning (ICML)*, pp. 8158–8168.

Russell, Stuart (2016). "Should we fear supersmart robots?" In: *Scientific American* 314.6, pp. 44–49.

Saerens, Marco, Patrice Latinne, and Christine Decaestecker (2002). "Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure". In: *Neural Computation* 14.1, pp. 21–41.

Samuelson, Larry and Jakub Steiner (2024). *Constrained data-fitters*. Tech. rep. 460. Department of Economics Working Paper Series, University of Zurich.

Savage, Leonard J. (1971). "Elicitation of personal probabilities and expectations". In: *Journal of the American Statistical Association* 66.336, pp. 783–801.

Schervish, Mark J. (1989). "A general method for comparing probability assessors". In: *The Annals of Statistics* 17, pp. 1856–1879.

Shuford, Emir H., Arthur Albert, and H. Edward Massengill (1966). "Admissible probability measurement procedures". In: *Psychometrika* 31.2, pp. 125–145.

Sun, Yanmin, Mohamed S. Kamel, and Yang Wang (2006). "Boosting for learning multiple classes with imbalanced class distribution". In: *IEEE International Conference on Data Mining (ICDM)*.

Ting, Kai Ming (2002). "An instance-weighting method to induce cost-sensitive trees". In: *IEEE Transactions on Knowledge and Data Engineering* 14.3, pp. 659–665.

Vanderschueren, Toon et al. (2022). "Predict-then-optimize or predict-and-optimize? An empirical evaluation of cost-sensitive learning strategies". In: *Information Sciences* 594, pp. 400–415.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems*.

Wang et al. (2017). "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2097–2106.

Wiener, Norbert (1960). "Some moral and technical consequences of automation". In: *Science* 131, pp. 1355–1358.

Xia, Fen et al. (2009). "A closed-form reduction of multi-class cost-sensitive learning to weighted multi-class learning". In: *Pattern Recognition*.

Zadrozny, Bianca and Charles Elkan (2001). "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers". In: *International Conference on Machine Learning (ICML)*, pp. 609–616.

– (2002). "Transforming classifier scores into accurate multiclass probability estimates". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 694–699.

Zadrozny, Bianca, John Langford, and Naoki Abe (2003). "Cost-sensitive learning by cost-proportionate example weighting". In: *IEEE International Conference on Data Mining (ICDM)*, pp. 435–442.

Zhou and Xu-Ying Liu (2010). "On multi-class cost-sensitive learning". In: *Computational Intelligence*.

A Relations to Cost-Sensitive and Imbalanced Learning

In this section, we use our general incentive-based approach to compare, contrast, and unify four classes of solution to cost-sensitive learning and class imbalance: (i) thresholding, (ii) reweighting, (iii) resampling, and (iv) base rate adjustments. We then use our approach to (iv) derive a novel and general multi-class formula for analytical recalibration across these contexts.

Contrasting with Section 2, we mainly represent downstream preferences and objectives as nonnegative and nondegenerate costs c rather than utilities u to facilitate comparison with the literature. It is straightforward to show that our results and approach hold upon replacing expected utility maximization with conditional cost (i.e., risk) minimization.

A.1 Thresholding and Conditional Risk Minimization

Thresholding refers to the practice of changing the decision cutoffs for classification, and conditional risk minimization to its appropriate multi-class generalization. In its simplest form, this means to train a prediction model without reference to cost, and then classify instances based not on their immediate predictions, but on their cost-derived conditional risk.

In the binary-class case $\mathcal{Y} = \{0, 1\}$ where class distributions q are summarized by the positive-class probability q_1, Elkan (2001) derives a closed-form solution for the positive-class probability $q_1^c \in [0, 1]$ where the conditional risks are equal, $\bar{c}(1, q^c) = \bar{c}(0, q^c)$. This probability q_1^c in turn serves as a threshold for classifying positive instances:

$$y^c(q_1) \overset{(1)}{=} 1\{q_1 \geq q_1^c\} \overset{(2)}{=} 1\{\bar{c}(1, q_1) \leq \bar{c}(0, q_1)\}.$$

In the multi-class case, there is no simple threshold determining classification as in the first equality, yet the conditional risk minimization inherent in the second equality still applies and is generalized by the conditional risk minimizer (in our leading case of utility, the expected utility maximizer (5)). This conditional risk minimizer formula appears previously in Margineantu (2002), who also notes that such ex post conditional risk minimization allows general application of a single trained model across misclassification cost functions. The Metacost algorithm of Domingos (1999) applies the same conditional risk minimization formula (5) in an interim training stage in order to reclassify the model *training* data, before training the final algorithm on the reclassified data. Thus, Metacost departs from unweighted learning by applying conditional risk minimization at the training rather than at the classification stage, which requires retraining an algorithm for any change in misclassification costs.

A.2 Reweighting

Reweighting training instances changes their relative importance in the loss function. By implicitly changing training incentives, reweighting is spiritually similar to our incentive-based approach; nevertheless, it is only a special case confined to having a single weight per instance. Consider, for example, the common case of *class weighting* discussed in Section 2.8. The corresponding *class-weighted prediction loss* is:

$$\ell^w(p, y) \equiv w_y \ell(p, y) \tag{22}$$

which is evidently a special case of our preference-weighted loss. Again, the optimal class-weighted prediction of class y given class distribution q becomes:

$$p_y^w(q) = \frac{w_y q_y}{\langle w, q \rangle} \tag{23}$$

We now show how our novel approach relates to and extends existing class-reweighting approaches. We discuss the connections to problems of class (im)balance in Section A.4.

In the binary classification case, Theorem 1 of Elkan (2001) considers how to make a given target probability threshold on the positive class p_1^* correspond to a given probability threshold p_1^o for an accuracy-maximizing classifier. In our notation, Elkan's problem is to choose weights on the negative class $w = (w_0, 1)$ such that:

$$q_1 \geq p_1^* \iff p_1^w(q_1) \geq p_1^o$$

By monotonicity of $p_1^w(q_1)$ in its argument q_1, this is equivalent to solving:

$$p_1^w(p_1^*) = p_1^o$$

which yields Elkan's formula:

$$w_0 = \frac{p^*}{1 - p^*} \frac{1 - p_0}{p_0}$$

The threshold approach is not generalizable to the multi-class case. In contrast, our approach of beginning with (mis)classification preferences c or u and incorporating them into training incentives via the preference-weighted prediction loss is generally applicable, with even multi-class weighting (22) as a special case.

In the multi-class case with general misclassification costs c, Breiman et al. (1984), Domingos (1999), Ting (2002), and Drummond and Holte (2003) propose reducing the cost matrix

$c(y', y)$ to a cost vector $c(y) = \sum_{y'} c(y', y)$ that can be used for class weighting. The resulting machine incentives are significantly different from those of our incentive-based approach because:

$$\ell(p, y)c(y) = \ell(p, y) \sum_{y'} c(y', y) \neq \sum_{y'} \ell(p, y')c(y', y) \equiv \ell^c(p, y)$$

Indeed, in the same multi-class case with general misclassification costs c, Zhou and Liu (2010) propose an alternative class-weighting scheme w to address the issues of the preceding cost-reduction proposal, which instead solves:

$$\frac{w_y}{w_{y'}} = \frac{c(y', y)}{c(y, y')} \quad \text{for all } y, y' \in \mathcal{Y}.$$

If such a weighting scheme does not exist, they decompose the multi-class problem into a set of binary-class problems and apply the preceding constraints separately. Even if such a weighting scheme does exist, it typically disagrees with our approach since the ratio of weights between classes is determined only by a pair of misclassification costs, rather than a pair of conditional risks which may depend non-trivially on the entire vector of losses given a true label. In contrast to their approach, our solution (i) generally exists, (ii) yields closed-form prediction formulas, and (iii) is rigorously grounded in prediction and classification incentives via Theorem 1.

More generally, it is evident from comparing the set of cost-weighted and class-weighted predictions (16) that our incentive-based approach is not recovered by class weighting alone. Indeed, our approach naturally generalizes the existing practice of weighting observations only by their true class. For example, evaluating weighted loss for the logistic loss function yields a new family of weighted loss functions we term *doubly-weighted cross entropy*:[19]

$$\ell^c(p, y) = -\langle \log p, c(\cdot, y) \rangle \tag{24}$$

This allows for differentially costly misclassifications across labels, given a true class y. For example, some medical misdiagnoses may be less costly if they lead to similar courses of action, or image misclassifications may be less costly if the mislabels are still similar in some characteristic space.

Finally, another noteworthy but not directly comparable family of reweighting procedures for cost-sensitive learning involve boosting, where instance weights are chosen dynamically during training (Fan et al. 1999; Sun, Kamel, and Y. Wang 2006).

[19] This weighting procedure is equally applicable to other proper loss functions, such as the Brier score (Brier 1950).

A.3 Resampling

Resampling is intuitively similar to the preceding case of reweighting, except that it manipulates the training data rather than the training incentives (i.e., the algorithm). In the binary classification case, the problem is well-studied (e.g., Breiman et al. 1984; Kubat and Matwin 1997; Japkowicz 2000; Elkan 2001), particularly in the context of class imbalance to which we return in Section A.4.

Existing resampling approaches in the multi-class case reveal a particular advantage of resampling relative to reweighting: a single training instance can be repeatedly resampled, whereas it can be reweighted only once (unless it is also resampled). Most closely related to our approach, Xia et al. (2009) provide a theoretically grounded data expansion technique based on Abe, Zadrozny, and Langford (2004), in which a single training instance may be repeatedly resampled in proportion to its various cost entries. More specifically, given a distribution of features and classes $\mathbb{P}(x, y)$, they derive a resampled distribution $\hat{\mathbb{P}}(x, y)$ such that the minimizer of cost-sensitive classification on distribution $\mathbb{P}$ is theoretically equivalent to the minimizer of zero-one classification on distribution $\hat{\mathbb{P}}$. Thus, their solution based on manipulating training data is similar to our solution based on manipulating incentives. Still, our incentive-based approach confers several relative advantages.

As noted previously, an advantage of resampling relative to reweighting is the possibility of repeatedly resampling the same training instance, which allows resampling to convey more complex incentives than those permitted by reweighting. At the same time, a disadvantage of resampling is its inefficiency from both a data and computational standpoint, due to data loss from not sampling all observations, alongside redundancy, stochasticity, and increased complexity from sampling observations repeatedly. Our incentive-based approach captures the relative advantages of both reweighting and resampling — allowing for complex (mis)classification preferences without any training data manipulation.

In addition, our incentive-based approach yields a simple implementation and intuition via the cost-weighted prediction loss and closed-form solutions for predictions, which can also be used for analytical recalibration or belief recovery (Section A.5) as well as understanding the resulting implicit incentives to learn (Section 2.7). Additionally, our approach is simply extended to *example-based costs* (Zadrozny, Langford, and Abe 2003, Abe, Zadrozny, and Langford 2004) by allowing the cost function in our incentive-based approach to also depend on features, $c(x, y, y')$, in which case the conditional risk and the adjustment formula will also depend on features x. On the other hand, our approach is inherently based in proper surrogate loss functions, which may limit its use to certain algorithm classes; in contrast,

resampling can be generally implemented, even if its theoretical justifications are equally limited.

A.4 Base Rate Adjustments and Class Imbalance

The resampling and reweighting approaches discussed in the preceding Section A.2 and Section A.3 are often motivated by questions of class imbalance (for overviews, see Chawla, Japkowicz, and Kołcz 2004 and Fernández et al. 2018). In turn, class imbalance and its resolution relate closely to the idea of modifying class base rates. For example, in settings of binary classification, a common intuition is that standard algorithms are biased toward the majority class, which can be addressed by either upweighting or upsampling the majority class, and/or downweighting or downsampling the minority class. We have already resolved the question of correcting asymmetric human classification incentives in the machine prediction problem (Theorem 1), and our incentive-based approach was notably independent of the actual class distribution.

Still, the incentive-based approach is also productive for the goal of effectively changing base rates: namely, given overall class base rates $\bar{q} \in \Delta(\mathcal{Y}) \cap \mathbb{R}^m_{++}$ in training, how do we modify prediction incentives such that the machine is incentivized to choose *as if* the base rates were instead $\bar{p} \in \Delta(\mathcal{Y})$? Saerens, Latinne, and Decaestecker (2002) provide an adjustment formula for posterior probabilities when only base rates are modified, which is rooted in Bayes' rule. We recall this in our framework and language below.[20]

Proposition 3 (Saerens, Latinne, and Decaestecker (2002))**.** *For a fixed set of conditional feature probabilities* $\mathbb{P}(X = x|Y = y)$*, the conditional class distributions* $p(x), q(x) \in \Delta(\mathcal{Y})$ *of feature vector* x *under respective target base rates* $\bar{p} \in \Delta(\mathcal{Y})$ *and source base rates* $\bar{q} \in \Delta(\mathcal{Y}) \cap \mathbb{R}^m_{++}$ *are related by formula:*

$$p_y(x) = \frac{\bar{w}_y q_y(x)}{\langle \bar{w}, q(x) \rangle} \text{ for all } y \in \mathcal{Y}, \tag{25}$$

where $\bar{w}_y = \bar{p}_y / \bar{q}_y$.

It is immediate by comparison of (25) to class-weighted adjustments (16) and the underlying class-weighted prediction loss (22) that the base rate adjustment formula is implementable by our incentive-based approach with class weights $\bar{w}_y = k\bar{p}_y/\bar{q}_y$ for any positive constant k.

A special case is the practice of inverse class weighting $\bar{w}_y = 1/\bar{q}_y$, which normalizes the base

[20] In the binary case, a similar formula appears in Theorem 2 of Elkan (2001). The formula also appears previously in the economic literature on Bayesian persuasion (Alonso and Câmara 2016).

rates to be uniform across classes. Whether this provides the correct classification incentives depends on the classification preferences and the distribution of test data. Indeed, our incentive-based approach provides a useful means of simultaneously unifying and disentangling the related problems of cost-sensitive misclassification and base rate adjustment strategies to address class imbalance.

A.5 Inversion and Calibration

When predictions are probabilistically correct, they are said to be *calibrated*. Probabilistic model calibration has clear advantages, as evidenced by the large literature devoted to its study (e.g., Platt 1999; Zadrozny and Elkan 2001; Zadrozny and Elkan 2002; Guo et al. 2017; Minderer et al. 2021). By design, however, the predictions generated by incentivizing the machine according to (mis)classification preferences are typically *mis*calibrated because preference-weighted predictions distort latent probabilities:

$$p^u(q) \neq q$$

This raises a question about when it is possible to analytically recover calibrated probability estimates from observed, cost-weighted predictions. Theorem 2 provides a simple yet general analytical solution when the utility function $u : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ can be represented as an invertible matrix U with entries $U_{y',y} = u(y', y)$. In this case, let U^{-1} denote the inverse matrix of U, and let $\bar{u}^{-1}(y, p)$ denote the expected utility of y given prediction p and (inverse) utility function u^{-1} defined by matrix U^{-1}. Then:

Theorem 2 (Analytical Recalibration)**.** *For a nonnegative, nondegenerate classification utility u represented as an invertible matrix U and a strictly proper loss function ℓ, the optimal cost-weighted prediction function (8) is invertible with closed-form:*

$$[p^u_y]^{-1}(p) = \frac{\bar{u}^{-1}(y, p)}{\sum_{y' \in \mathcal{Y}} \bar{u}^{-1}(y', p)} \tag{26}$$

where u^{-1} is the utility function defined by inverse matrix U^{-1}.

Theorem 2 provides a simple means to recover latent probabilities q underlying observed predictions $p^u(q)$ as long as the utility matrix is invertible: compute the inverse utility matrix, and then compute the normalized risks of the observed predictions according to the inverse matrix. In the special case of class weighting with a diagonal matrix defined by weights $w \in \mathbb{R}^m_{++}$, the matrix inverse U^{-1} is just the inverse of every diagonal entry, and so (26)

reduces to:

$$[p_y^w]^{-1}(p) = \frac{(p_y/w_y)}{\sum_{y' \in \mathcal{Y}}(p_{y'}/w_{y'})} \tag{27}$$

Even this special multi-class case generalizes existing binary analytical calibration methods such as Dal Pozzolo et al. (2015).

When costs are not invertible, multiple conditional probability distributions can generate the same prediction, resulting in a non-recoverable loss of probabilistic information. Still, given the observed prediction data $(P \equiv f(X), Y)$, it remains possible to empirically recalibrate predictions by fitting a model for the empirical class probabilities:

$$\hat{q}_y(p) \equiv \mathbb{P}(Y = y | P = p),$$

e.g., Platt (1999), Zadrozny and Elkan (2001), Zadrozny and Elkan (2002), and Guo et al. (2017)). Even when underlying probabilities are analytically recoverable from observed predictions, whether the resulting probabilities are indeed well-calibrated is an application- and algorithm-specific empirical question.

A related but subtly different question is whether predictions are well-calibrated to their incentives, i.e., *loss-calibrated* (Caplin, Martin, and Marx 2022b). When the cost matrix is invertible, this can be checked by seeing whether the analytically recalibrated predictions (26) are empirically well-calibrated in the traditional sense, according to standard methods (e.g., DeGroot and Fienberg 1983; Niculescu-Mizil and Caruana 2005; Naeini, Cooper, and Hauskrecht 2015; Nixon et al. 2019). A more general approach that does not require analytical invertibility of the adjustment formula is the following. The predictions are *loss-calibrated* if the empirical predictions P coincide with the analytical prediction $p^u(\cdot)$ evaluated at the empirically recalibrated beliefs $\hat{q}(P)$:

$$p^u(\hat{q}(P)) = P \tag{28}$$

In the case of a proper loss function (4) where there are no incentives to misreport, i.e., $p^u(q) = q$ for all $q \in \Delta(\mathcal{Y})$, loss calibration collapses to standard multi-class calibration (e.g., Minderer et al. 2021):

$$\hat{q}(P) = P$$

Definition (28) of loss calibration does not rely on invertibility of the adjustment formula $p^u(\cdot)$, only that there exists an accurate empirical procedure or model for recovering recalibrated beliefs $\hat{q}(p)$. In particular, loss calibration permits multiple latent "subjective" posterior probabilities to map to the same observed prediction. Since the observed $\hat{q}(p)$ will be a convex

combination of such latent posteriors at every realization $P = p$, validity of this definition requires that level sets of the adjustment formula $p^u(\cdot)$ be convex. We confirm this in the following lemma.

Lemma 1. *For any nondegenerate and nonnegative classification utility u, level sets of the analytical adjustment formula (8) are convex. That is, for any $q, q' \in \Delta(\mathcal{Y})$ and $\alpha \in [0, 1]$,*

$$p^u(q) = p^u(q') = p \implies p^u(\alpha q + (1 - \alpha)q') = p$$

Thus, (28) is a valid definition of loss calibration, in the sense that if it is violated, there exists some latent subjective posterior q for which the observed prediction is not well-calibrated to its incentives. Loss calibration is testable similarly to calibration, since $p^u(\cdot)$ is a known, deterministic function that can be applied ex post to the recovered probabilities P.

Finally, we note that determining whether a prediction model is loss-calibrated is essentially the same as determining whether its actual objective — as summarized by a utility function u — coincides with its behavioral objective — as summarized by a utility function $\hat{u}$ *estimated* from the prediction data (P, Y), e.g., by multivariate Platt (1999) scaling, and analogously to the inverse reinforcement learning exercise (Ng and Russell 2000). In turn, this is analogous to establishing a zero reward-result gap (Hubinger et al. 2019; Leike et al. 2018) in the machine's prediction problem.

B Proofs

Proof of Theorem 1. For ease of reference with the statement in Theorem 1, we restrict to the classification problem $\mathcal{A} = \mathcal{Y}$. However, note that the same result and argument hold for downstream choices and preferences in any finite set $\mathcal{A}$.

Nondegeneracy (1) of u implies that, for any class distribution $q \in \Delta(\mathcal{Y})$, the expected utility is positive for some label y', so that the sum across labels is positive:

$$\sum_{y' \in \mathcal{Y}} \bar{u}(y', q) > 0$$

Therefore we can also define the normalized vector of expected utilities $\bar{h}^u(q) \in \Delta(\mathcal{Y})$ as:

$$\bar{h}^u_y(q) \equiv \frac{\bar{u}(y, q)}{\sum_{y' \in \mathcal{Y}} \bar{u}(y', q)} \tag{29}$$

Next, recall the prediction problem (7). Decomposing cost-weighted prediction loss and rearranging terms yields:

$$\begin{aligned}
\bar{\ell}^u(p, q) &= \sum_{y \in \mathcal{Y}} \ell^u(p, y) q_y \\
&= \sum_{y \in \mathcal{Y}} \left[\sum_{y' \in \mathcal{Y}} \ell(p, y') u(y', y) \right] q_y \\
&= \sum_{y' \in \mathcal{Y}} \ell(p, y') \sum_{y \in \mathcal{Y}} u(y', y) q_y \\
&= \sum_{y' \in \mathcal{Y}} \ell(p, y') \bar{u}(y', q) = \bar{\ell}(p, \bar{u}(\cdot, q))
\end{aligned}$$

Therefore the optimal prediction given a latent posterior q solves:

$$\operatorname*{argmin}_{p \in \Delta(\mathcal{Y})} \bar{\ell}(p, \bar{u}(\cdot, q)) \quad \text{or equivalently,} \quad \operatorname*{argmin}_{p \in \Delta(\mathcal{Y})} \bar{\ell}(p, \bar{h}^u(q))$$

since the latter objective just entails division of the former objective by a positive constant $\sum_{y' \in \mathcal{Y}} \bar{u}(y', q)$. Since the unweighted loss function ℓ is strictly proper (4), it follows that $\bar{h}^u(q)$ is the unique optimal solution to the prediction problem (7). The closed-form solution (8) follows by definition (29) of $\bar{h}^u(q)$. □

Proof of Proposition 2. By definition of indirect learning (12) and the optimal prediction (7),

$$\tilde{\ell}^u(q) = \min_{p \in \Delta(\mathcal{Y})} \langle \ell^u(p, \cdot), q \rangle \tag{30}$$

As a pointwise minimum of linear functions in q, $\tilde{\ell}^u$ is concave. By Theorem 1, there is a unique minimizer $p^u(q)$ for each q, and thus a unique subgradient $\ell^u(p^u(q), \cdot)$ to $\tilde{\ell}^u$ at q (e.g., Corollary 4.4.4 of Hiriart-Urruty and Lemaréchal (2004)). By Theorem 25.1 of Rockafellar (1970), $\tilde{\ell}^u$ is differentiable at q, and its derivative is the unique subgradient. Proposition 2 simply expresses this derivative component-wise. □

Proof of Theorem 2. Throughout this proof, we alternatively represent the utility function $u : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ as an $m \times m$ matrix U, with rows and columns corresponding to labels and classes, respectively, and let U^{-1} denote its inverse. Similarly, denote probability distributions $p, q \in \Delta(\mathcal{Y})$ as column vectors, and let superscript T denote the transpose operator. The optimal utility-weighted prediction from Theorem 1 expressed in matrix form is:

$$p^u(q) = \frac{1}{\iota^T U q} U q \tag{31}$$

where ι is defined as a column vector of ones and the term $\iota^T U q$ is a real-valued function of q.

Fix $q \in \Delta(\mathcal{Y})$. Since U is nonnegative and nondegenerate, $Uq \in \mathbb{R}^m_+ \setminus \{0\}$, and so there exists a unique scalar multiplier $\alpha > 0$ such that $\alpha U q \in \Delta(\mathcal{Y})$. By (31), $\alpha = (\iota^T U q)^{-1}$ and $p^u(q) = \alpha U q$. Therefore:

$$U^{-1}[p^u(q)] = U^{-1}[\alpha U q] = \alpha U^{-1} U q = \alpha q$$

Dividing the first and last terms of the preceding equalities by $\alpha > 0$,

$$\alpha^{-1} U^{-1}[p^u(q)] = q$$

Since $\alpha > 0$ and $q \in \Delta(\mathcal{Y})$, we have $U^{-1}[p^u(q)] \in \mathbb{R}^n_+ \setminus \{0\}$, and it must be that $\alpha = \iota^T U^{-1} p^u(q)$. Thus, the desired inverse construction:

$$[p^u]^{-1}(p) = \frac{U^{-1} p}{\iota^T U^{-1} p}$$

follows. Again, equivalence with (26) is easily verified. □

Proof of Lemma 1. For what follows, recall our matrix notation in (31) from Theorem 2:

$$p^u(q) = \frac{1}{\iota^T U q} U q$$

Our first claim is that for any $q, q' \in \Delta(\mathcal{Y})$ such that $Uq = \beta Uq'$ for some $\beta > 0$, we have $p^u(q) = p^u(q')$. This follows because:

$$p^u(q) = \frac{1}{\iota^T U q} U q = \frac{1}{\iota^T \beta U q'} \beta U q' = \frac{1}{\iota^T U q'} U q' = p^u(q')$$

Our second, converse claim is that for any two posteriors $q, q' \in \Delta(\mathcal{Y})$ that generate the same analytical prediction $p^u(q) = p^u(q') = p$, it must be that $Uq = \beta Uq'$ for some $\beta > 0$. Suppose not. Then there exist realizations y, y' and $\gamma > 0$ such that $(Uq)_y = \gamma (Uq)_{y'}$, but $(Uq')_y \neq \gamma (Uq')_{y'}$. Yet, the respective equality and inequality are preserved under any scalar division, implying that $p^u(q) \neq p^u(q')$, a contradiction to our premise.

Having established by the second claim that $Uq = \beta Uq'$ for some $\beta > 0$, it follows for any $\alpha \in [0, 1]$ that:

$$U[\alpha q + (1 - \alpha) q'] = \alpha U q + (1 - \alpha) U q' = [\alpha \beta + (1 - \alpha)] U q'$$

The desired result then follows by the first claim. □

Proof of Proposition 1. The proof makes use of an indicator function (i.e., a one-hot encoder) $\delta : \mathcal{Y} \to \{0, 1\}^m$ defined by $\delta_y(y') \equiv I\{y' = y\}$.

$$\begin{aligned}
\mathbb{E}[\ell(P, Y)] &= \mathbb{E}\left[\langle \ell(P, \cdot), \delta(Y) \rangle\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\langle \ell(P, \cdot), \delta(Y) \rangle \middle| P\right]\right] \\
&= \mathbb{E}\left[\langle \ell(P, \cdot), \mathbb{E}\left[\delta(Y) | P\right] \rangle\right] \\
&= \mathbb{E}\left[\langle \ell(P, \cdot), Q \rangle\right] \\
&= \mathbb{E}[\bar{\ell}(P, Q)]
\end{aligned}$$

□

C Figures

In the figures that follow, we disaggregate the loss in the test sample across five training runs per training incentive. We conduct multiple training runs in order to disentangle systematic

changes in performance from stochasticity in the training procedure. For each run, the point denotes the optimal performance and the training step at which it is achieved.

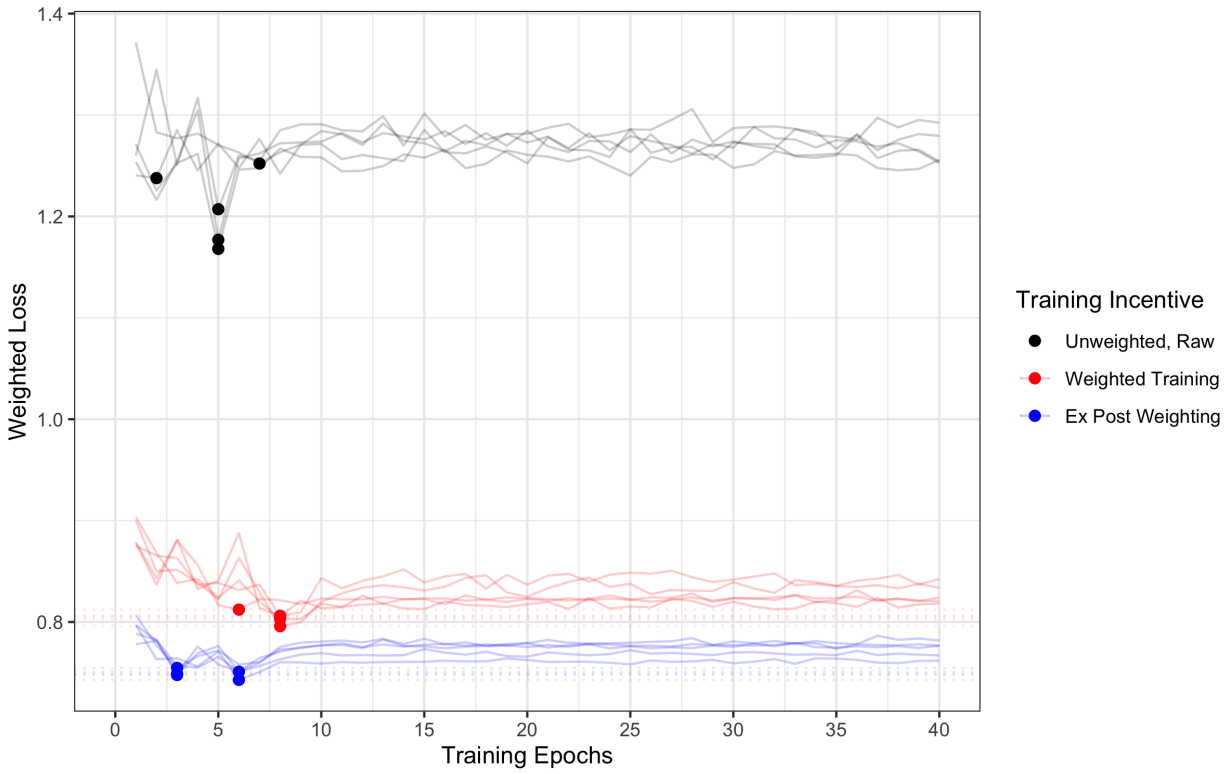

Figure 4: Weighted loss when emphasizing pneumonia, evaluated in test sample. The black lines represent the weighted loss (across training runs) from unweighted training without adjusting predictions. The red lines represent the weighted loss from weighted training. Finally, the blue lines represent the weighted loss from unweighted training after analytically adjusting predictions. For each run, the point denotes the minimal weighted loss and the training epoch at which it is achieved. Consistently across training runs, we outperform the machine on its own objective by not training according to downstream incentives, but rather analytically adjusting for them ex post.

For our leading case of pneumonia instances in chest X-rays, we present figures extended in two further ways. First, when plotting performance according to weighted loss (Figure 4), we include the *uncorrected* predictions from unweighted training [*Unweighted, Raw*, in black]. As is to be expected, these perform worse according to weighted loss than training according to that objective itself. However, this conflates an issue of misspecified predictions with their underlying information content. By analytically correcting the unweighted predictions [*Ex Post Weighting*, in blue], we isolate the information channel and achieve better performance on weighted loss than training on that objective [*Weighted Training*, in red]. Across five runs, we achieve better performance on the weighted objective under the *Ex Post Weighting*

procedure than under *Weighted Training*, with a mean improvement of 6.9%.

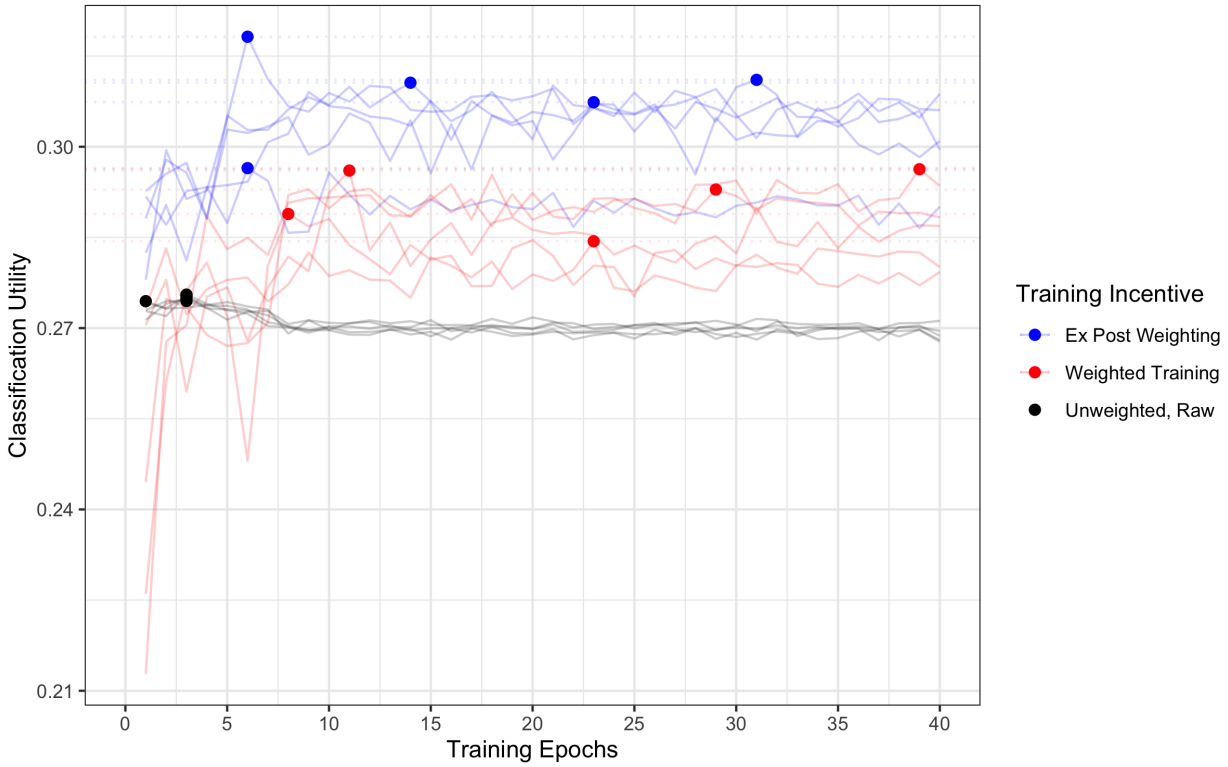

Figure 5: Classification utility when emphasizing pneumonia, evaluated in test sample. The black lines represent the achieved utility (across training runs) from unweighted training without adjusting predictions. The red lines represent utility from weighted training. Finally, the blue lines represent utility from unweighted training after analytically adjusting predictions. For each run, the point denotes the maximal classification utility and the training epoch at which it is achieved. Consistently across training runs, we outperform the machine on the downstream utility objective by not training according to downstream incentives, but rather analytically adjusting for them ex post. Based on the preceding Figure 4, we attribute this to underperformance and suppressed learning when the machine is trained according to utility-weighted cross entropy.

Second, we include an analogous figure of classification utility (Figure 5). Perhaps surprisingly, the training epoch that minimizes weighted loss is not necessarily the one that also maximizes classification utility. We attribute this to the discontinuity of classification utility in predictions, combined with a heavy dependence on single observations of a rare but heavily weighted class. Nevertheless, our main point is robust: training under the unweighted objective and analytically correcting predictions consistently outperforms training under the weighted objective, even according to that objective. Across five runs, the worst performance of the unweighted procedure is approximately equal to the best performance of the weighted

procedure (achieving in each case a classification utility of 0.296).

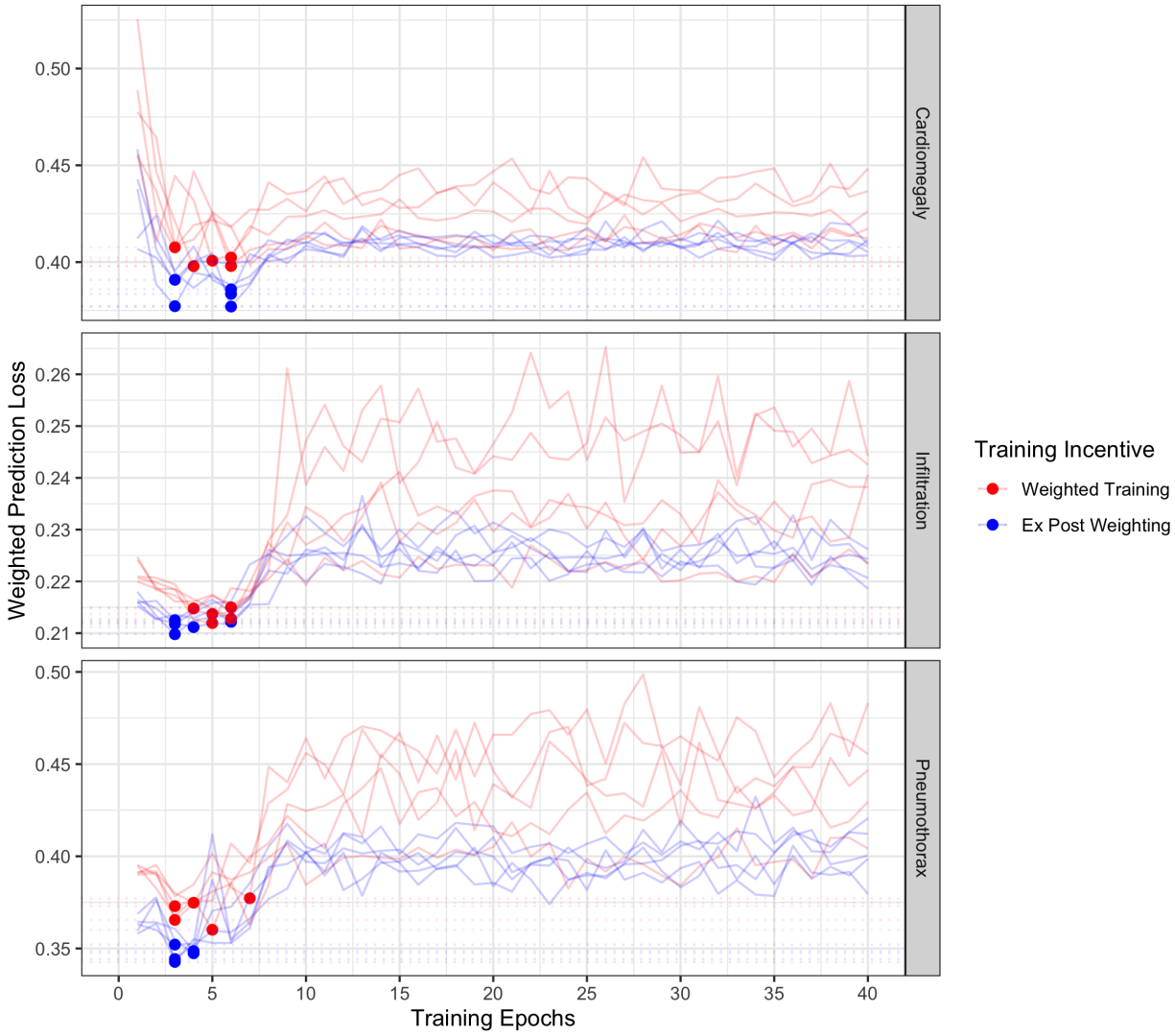

Figure 6: Weighted loss when respectively emphasizing cardiomegaly, infiltration, and pneumothorax, evaluated in test sample. The red lines represent the weighted loss from weighted training across five training runs. The blue lines represent the weighted loss from unweighted training after analytically adjusting predictions. For each run, the point denotes the minimal weighted loss and the training epoch at which it is achieved. On average and (with the exception of infiltration) consistently across training runs, we outperform the machine on its own objective by not training according to downstream incentives, but rather analytically adjusting for them ex post.

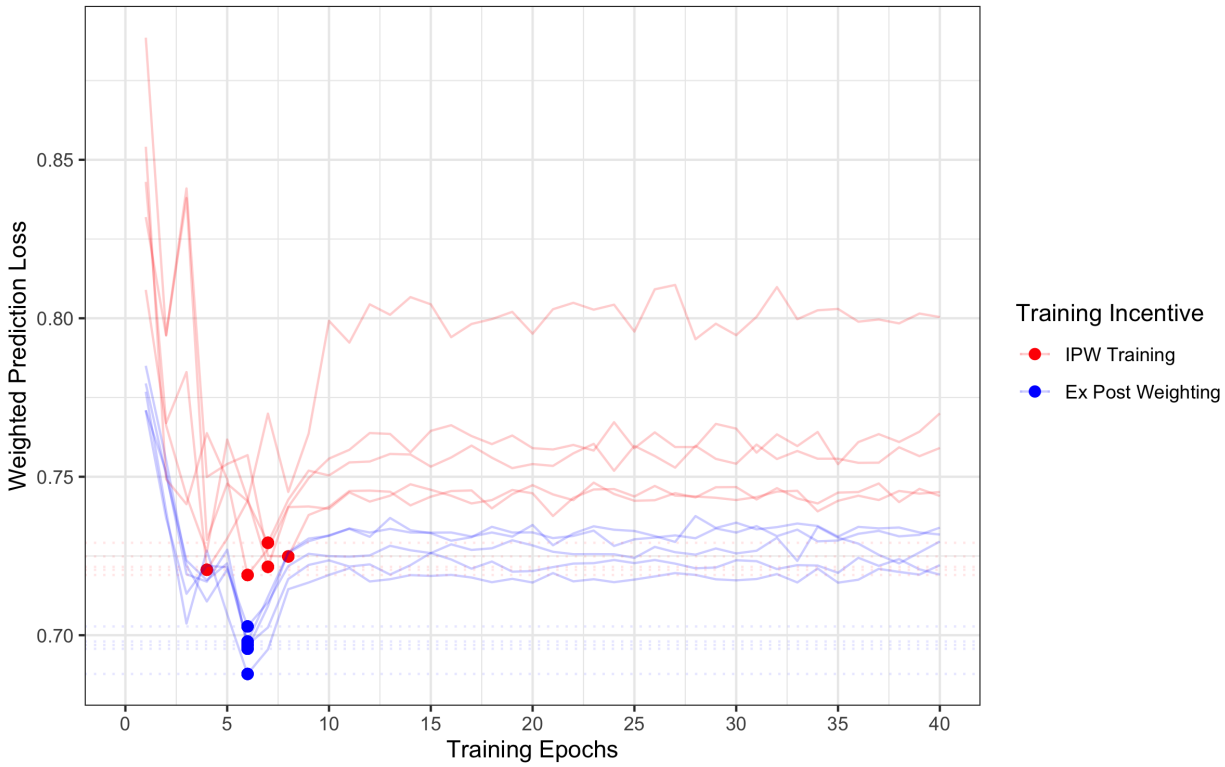

Figure 7: Weighted loss when balancing the data through inverse probability weighting. The red lines represent the weighted loss from weighted training across five training runs. The blue lines represent the weighted loss from unweighted training after analytically adjusting predictions. For each run, the point denotes the maximal classification utility and the training epoch at which it is achieved. Consistently across training runs, we outperform the machine on its own objective by not training with inverse probability weights, but rather analytically adjusting predictions to account for the weights ex post.

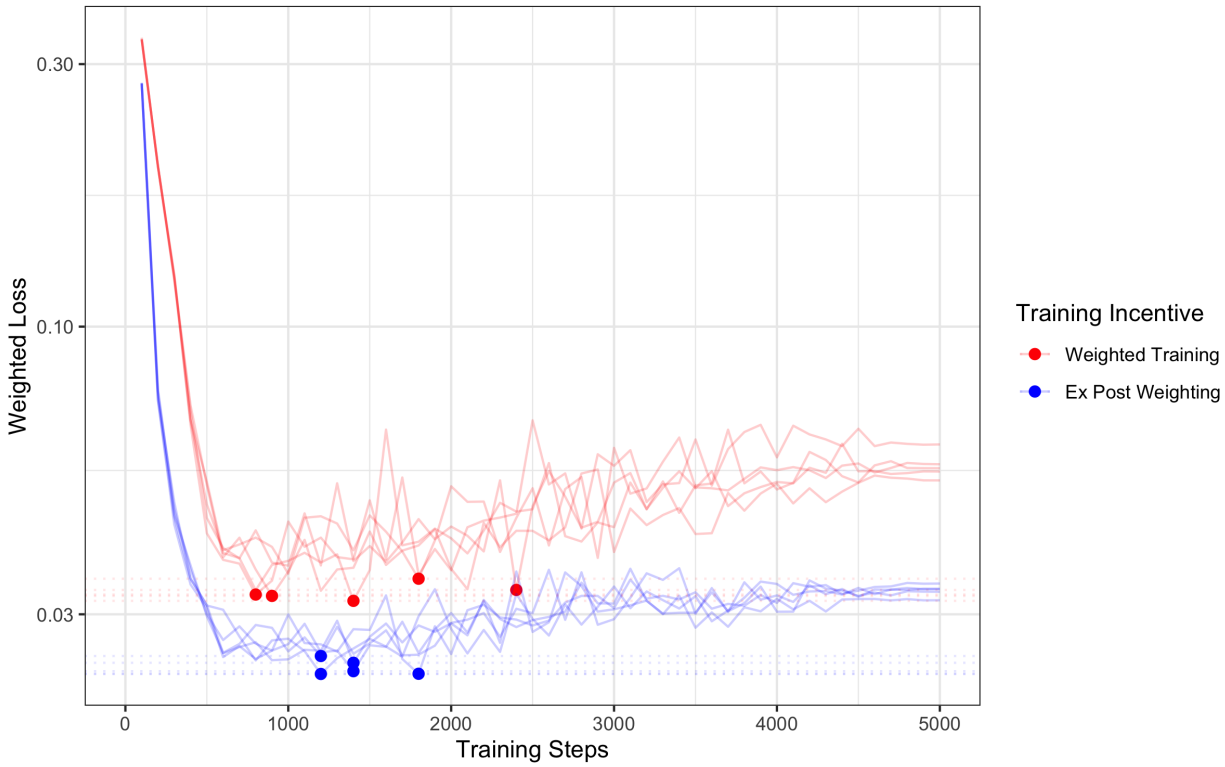

Figure 8: Weighted loss when emphasizing the (most difficult) class "cat" in CIFAR-10 data, evaluated in the test sample. The red lines represent the weighted loss from weighted training across five training runs. The blue lines represent the weighted loss from unweighted training after analytically adjusting predictions. For each run, the point denotes the minimal weighted loss and the training step at which it is achieved. Consistently across training runs, we outperform the machine on its own objective by not training according to downstream incentives, but rather analytically adjusting for them ex post.

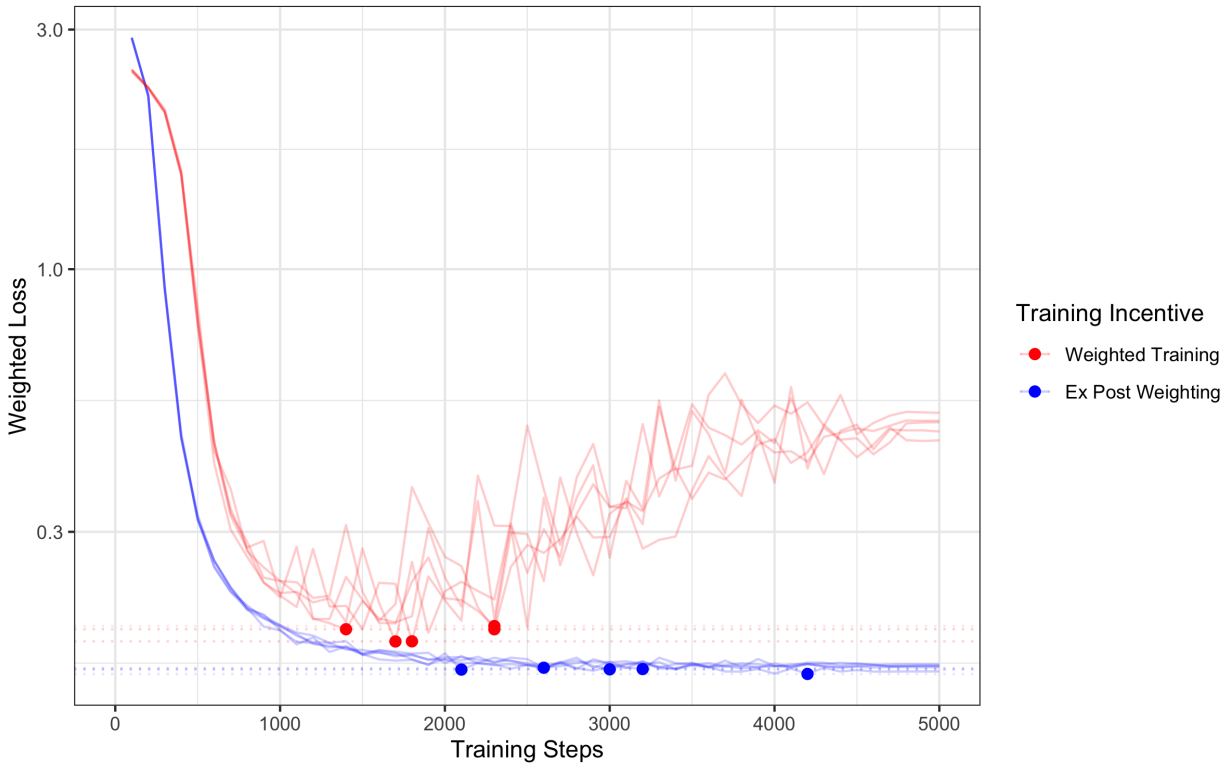

Figure 9: Weighted loss when emphasizing the (most difficult) class "maple tree" in CIFAR-100 data, evaluated in the test sample. The red lines represent the weighted loss from weighted training across five training runs. The blue lines represent the weighted loss from unweighted training after analytically adjusting predictions. For each run, the point denotes the minimal weighted loss and the training step at which it is achieved. Consistently across training runs, we outperform the machine on its own objective by not training according to downstream incentives, but rather analytically adjusting for them ex post.

D Data and Training Procedures

D.1 Chest X-Rays

Our model training procedure follows that of the CheXNeXt algorithm (Rajpurkar, Irvin, Ball, et al. 2018), in which a deep neural network was trained using the ChestX-ray14 dataset of Wang et al. (2017). The ChestX-ray14 dataset consists of 112,120 frontal chest X-rays that were synthetically labeled with up to fourteen thoracic diseases. Our code for model training is adapted from the publicly available CheXNeXt codebase of Rajpurkar, Irvin, Ball, et al. (2018).

As in their work, we adopt random horizontal flipping and normalize based on the mean and standard deviation of images in the ImageNet dataset (Deng et al. 2009). For each model,

we train a 121-layer dense convolutional neural network (DenseNet, Huang et al. 2017) with network weights initialized to those pretrained on ImageNet, using Adam with standard parameters 0.9 and 0.999 (Kingma and Ba 2015), and using batch normalization (Ioffe and Szegedy 2015). We use an initial learning rate of 0.0001 that is decayed by a factor of 10 each time the validation loss plateaus after an epoch.

Besides our class-weighting modification to the loss function, we modify their implementation in three ways. First, we restrict attention to the subset of 91,324 images with a single label (including "No Finding" of disease) and train multi-class classifiers using a 70-20-10 training-test-validation split. In related work, Caplin, Martin, and Marx (2022b) focus on binary classification of pneumonia and find statistically significant but economically small effects consistent with our results; this suggests that the extent of misalignment increases with the complexity of the problem and prediction space. Second, rather than conduct early stopping based on validation loss, we run each instance for 40 training epochs to compare the evolution of loss in the test sample and show that it is pointwise ordered across our training regimes. (The validation loss is still used implicitly by the algorithm to update the learning rate.) Third, we trade off a higher batch size of 16 at the expense of a slightly smaller imaging scaling size of 224 by 224 pixels (instead of a batch size of 8 and an image rescaling of 512 by 512 pixels, respectively).

D.2 CIFAR

We fine-tune and evaluate the Base variant of the Vision Transformer model with 16x16 pixel patch size (ViT-B16, Dosovitskiy et al. 2021), pre-trained on the ImageNet-21k dataset (Deng et al. 2009), on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky and Hinton 2009). Summarizing their training implementation, the Base model contains 12 layers, hidden size 768, MLP size 3072, 12 heads, and 86 million parameters. The model is trained using Adam (Kingma and Ba 2015) with $\beta_1 = 0.9, \beta_2 = 0.999$, a batch size of 4,096, and a high weight decay of 0.1. In the fine-tuning stage, we follow their implementation with a batch size of 512, linear learning rate warm-up and cosine decay with a base learning rate of 0.03, stochastic gradient descent with momentum of 0.9, and gradient clipping at global norm of 1. The only modification we make from their implementation is to vary class weighting in the loss function.