Uncertainty-Aware Systems for Human-AI Collaboration

Anonymous authors
Paper under double-blind review

Abstract

Learning to defer (**L2D**) algorithms improve human-AI collaboration (**HAIC**) by deferring decisions to human experts when they are more likely to be correct than the AI model. This framework hinges on machine learning (**ML**) models' ability to assess their own certainty and that of human experts. L2D struggles in dynamic environments, where distribution shifts impair deferral. We present two uncertainty-aware approaches to HAIC. First, we enhance L2D by combining ML outputs with density functions to improve uncertainty estimation and robustness. Second, we use density-based conformal prediction to assess epistemic uncertainty, dynamically balancing the assignment strategy by either employing L2D or deferring high-uncertainty instances directly to human experts. Both methods are the first uncertainty-aware approaches for HAIC that also address limitations of L2D systems including cost-sensitive scenarios, limited human predictions, and capacity constraints. Empirical evaluation in fraud detection shows both approaches outperform state-of-the-art baselines while improving calibration and supporting real-world adoption.

1 Introduction

While artificial intelligence (**AI**) and machine learning (**ML**) models rival or surpass human expert performance and efficiency in various domains, including finance (Khandani et al., 2010; Awoyemi et al., 2017), criminal justice (Goel et al., 2021), and healthcare (Gulshan et al., 2016; Somanchi et al., 2015; Esteva et al., 2017), their use in high-stakes settings still faces challenges. Their reliance on training data constrains their scope and they often struggle to generalize under dynamic conditions, such as in adversarial settings in finance, where fraudsters adapt to new models (Perdomo et al., 2020; Lunghi et al., 2023), or under concept drift (Gama et al., 2014). Furthermore, their predictions often lack transparency, posing difficulties to interpretability and trust (Saeed & Omlin, 2023; Gohel et al., 2021). As a result, their applicability in fully automated decision-making systems remains constrained.

In contrast, humans can access external information, adapt to changing environments, and provide causal reasoning and explanations for decisions (Gopnik & Wellman, 2012). These attributes make human experts indispensable in scenarios where interpretability, adaptability, and broader contextual awareness are required. Consequently, combining the complementary strengths of humans and ML models through human-AI collaboration (**HAIC**) systems has emerged as a promising approach to address the limitations of fully automated systems (Dellermann et al., 2019; De-Arteaga et al., 2020).

Optimizing the allocation of decisions to humans or AI is a major challenge in HAIC. The simplest approach, rejection learning (**ReL**) (Chow, 1970; Hellman, 1970; Cortes et al., 2016; Geifman & El-Yaniv, 2017), addresses this by deferring to human experts instances with high model uncertainty. However, an optimal solution would involve taking human certainty into consideration, assigning instances to humans only when they are likely to outperform the model. Learning to defer (**L2D**) aims to achieve this by estimating the performance of both the model and the human decision-maker, thereby enabling a better allocation of tasks to the decision-maker most likely to succeed (Madras et al., 2018; Mozannar & Sontag, 2020; Mozannar et al., 2023; Verma & Nalisnick, 2022; Verma et al., 2023; Hemmer et al., 2022; 2023; Charusaie et al., 2022).

Although L2D improves decision-making performance, it struggles in dynamic environments. L2D methods rely on ML models to estimate the correctness of each decision-maker, whether human or AI. However,

when faced with distributional changes, these models often become unreliable (Gama et al., 2014). Beyond performance degradation, a critical issue is that they can exhibit high confidence in their predictions on out-of-distribution (**OOD**) samples (Hein et al., 2019), which in an L2D system leads to suboptimal assignments. In contrast, ReL, which leverages uncertainty measures (Hendrickx et al., 2024), can detect such changes at inference time, deferring novel instances to human experts, thus ensuring more robust handling of OOD data.

Unlike current ML models, humans can adapt to changing environments, a crucial advantage that HAIC systems should be designed to leverage. Ideally, one would rely on the models whenever they reliably perform well, which requires that model uncertainty be estimated in a way that allows humans to intervene when necessary. However, current L2D approaches remain limited in this respect, as they focus on obtaining calibrated performance prediction, without systematically incorporating uncertainty estimation into the deferral process. Integrating uncertainty estimation techniques with optimal decision allocation in HAIC is nontrivial, since this combination introduces new theoretical and practical challenges and tradeoffs, which are not present when each is studied in isolation. To address this gap, we propose two uncertainty-aware methods for HAIC and provide an empirical study of their advantages and limitations. Our results demonstrate the feasibility of introducing uncertainty into HAIC and show that different contexts may require different strategies, highlighting the importance of studying uncertainty estimation within the HAIC setting.

In addition to challenges posed by dynamic environments, L2D systems have failed to address other practical constraints (Leitão et al., 2022; Alves et al., 2024), including human work capacity limitations, reliance on extensive expert predictions for training, and limited consideration of cost-sensitive scenarios. While prior work (Hemmer et al., 2023; Tailor et al., 2024; Alves et al., 2024) has addressed some of these issues, our approach systematically accounts for all of the aforementioned constraints, enabling robust and practical deployment.

In summary, this work makes three key contributions to HAIC:

- An uncertainty-aware L2D system incorporating distance-aware models to enhance calibration and robustness against OOD data, with the ability to compute optimal assignments under misclassification costs and work capacity constraints.
- A hybrid system that uses density-based conformal prediction to balance ReL and L2D for adaptive deferral to human experts under distribution shifts, while also accommodating cost and capacity constraints.
- Extensive experimental evaluation in a realistic cost-sensitive fraud detection setting, using a benchmark dataset (Alves et al., 2025a) containing synthetic fraud analyst decisions modeled after real experts and a diverse range of testing conditions. This study shows improved performance over baseline methods across variations in noise, data availability, and work capacity constraints.

2 Related Work

ReL and uncertainty estimation. The simplest HAIC approach in the literature is ReL (Chow, 1970; Hellman, 1970; Cortes et al., 2016; Geifman & El-Yaniv, 2017), which often involves having the model abstain from predicting in high uncertainty cases (Hendrycks & Gimpel, 2017). Hendrickx et al. (2024) categorize rejection into two types: *ambiguity rejection*, which enables the model to abstain in scenarios where the target values are inherently ambiguous, and *novelty rejection*, where the model refrains from predicting for instances that deviate significantly from the training data.

These types of rejection align with different types of uncertainty: *aleatoric uncertainty*, which stems from irreducible randomness within the data (such as class overlap), thus leading to ambiguity rejection; *epistemic uncertainty*, which arises from incomplete knowledge, whether due to distribution shifts or uncertainty about the model fit to the data, leading to novelty rejection (Hüllermeier & Waegeman, 2021). Several studies have proposed methods to distinguish these types of uncertainty for applications like ReL (Senge et al., 2014) and active learning (Nguyen et al., 2019). Generative models have been proposed as an intuitive way to

quantify epistemic and aleatoric uncertainty (Hechtlinger et al., 2018; Sun et al., 2023; Postels et al., 2020), by leveraging estimates of the density function $p(\mathbf{x})$, where $\mathbf{x}$ is a feature vector, to decide whether input points are located in regions of high or low density, in which the latter serves as a proxy for high epistemic uncertainty.

Learning to defer. The L2D framework was introduced by Madras et al. (2018) to address the shortcomings of confidence-based ReL, which does not take into account the human decision-maker's performance. Leveraging not only class labels but also predictions from the downstream human decision-maker, a classifier and a deferral mechanism are trained, meaning L2D considers both model and human errors when optimizing decisions. Mozannar & Sontag (2020) highlight that the loss function proposed by Madras et al. (2018) is not consistent, proposing a consistent surrogate loss that includes a separate class for deferral. Verma & Nalisnick (2022) critique the L2D surrogate loss developed by Mozannar & Sontag (2020) for miscalibration issues, introducing a one-vs-all (**OvA**) approach that improves calibration by training the classifier and deferral model independently. In the multi-expert setting, Keswani et al. (2021) extend L2D to assign instances to multiple experts, while Verma et al. (2023) propose a new consistent and calibrated loss, generalizing the loss of Verma & Nalisnick (2022) to handle multi-expert scenarios.

However, L2D methods face several limitations in practical deployment (Leitão et al., 2022). They lack adaptability in dynamic environments, where data distributions can evolve over time due to concept drift (Gama et al., 2014), or in performative prediction (Perdomo et al., 2020). Changes in distribution degrade the performance of the ML models used in L2D to estimate the correctness of the decision-makers, which leads to sub-optimal assignments. Moreover, these frameworks rely heavily on labeled data sourced from all experts involved, rendering them impractical in real-world scenarios due to high labeling costs. Most existing L2D methods also do not consider cost-sensitive scenarios where misclassification costs can vary. Finally, these methods often neglect human work capacity constraints at inference time.

Recent advancements in L2D research have aimed to address some of these practical constraints, particularly the challenge of learning with limited expert data. Charusaie et al. (2022), Hemmer et al. (2023), and Tailor et al. (2024) explore active learning, semi-supervised learning, and meta-learning, respectively, to reduce the need for extensive human annotations. Alves et al. (2024) focus on cost and capacity constraints, which are crucial for real-world applications. Despite these advances, L2D research has not focused on it's limitations in dynamic environments. Since human experts adapt to changing conditions, HAIC systems can mitigate this, but doing so requires reliable uncertainty estimates to defer cases where models may fail.

Synthetic human expert predictions. Datasets with real human predictions are scarce and collecting expert annotations is often prohibitively expensive. Prior work resorts to simulating human decision-makers by injecting label noise. For example, Verma & Nalisnick (2022), Mozannar & Sontag (2020), and Charusaie et al. (2022) use CIFAR-10 (Krizhevsky, 2012) to create experts with high accuracy on some classes while deciding randomly on others, while Keswani et al. (2021) simulate racially biased experts by varying error rates across groups.

While these methods provide a simple way to introduce heterogeneity across experts, they have a major limitation: their accuracy is determined solely by class labels or a single feature, ignoring the broader set of input features that influence real expert decision-making. To the best of our knowledge, the most principled approach to synthetic expert simulation in L2D is the OpenL2D framework (Alves et al., 2025a). This framework addresses the aforementioned limitations by generating synthetic expert predictions through instance-dependent label noise, where the probability of error is a function of instance features and additional information available to decision-makers (e.g., a risk score provided by a separate ML model (De-Arteaga et al., 2020)). This provides a more realistic approximation of human decision-making behavior. By applying this framework to a public fraud detection dataset, Alves et al. (2025b) produce the Financial Fraud Alert Review (**FiFAR**) dataset, which contains predictions from synthetic fraud analysts over 30K instances. Crucially, this dataset was constructed with the explicit goal of realistically simulating the behavioural properties (*e.g.,* interrater agreement, consistency, performance distribution) of a team of highly skilled experts. These synthetic experts are supported by decision-making literature and were modeled after real financial fraud analysts. By adopting this dataset, our empirical evaluation of L2D methods is, to the best of our knowledge, based on the most realistic set of expert decisions to date.

3 Uncertainty-Aware L2D

To address the limitations of L2D systems in dynamic environments, we incorporate density-aware modeling (Bui & Liu, 2024) into our L2D system (Figure 1). This enables the development of an L2D system that is more robust against distribution shifts, ensuring improved calibration, reduced overconfidence on OOD data, and more adequate expert modeling in scenarios with limited data.

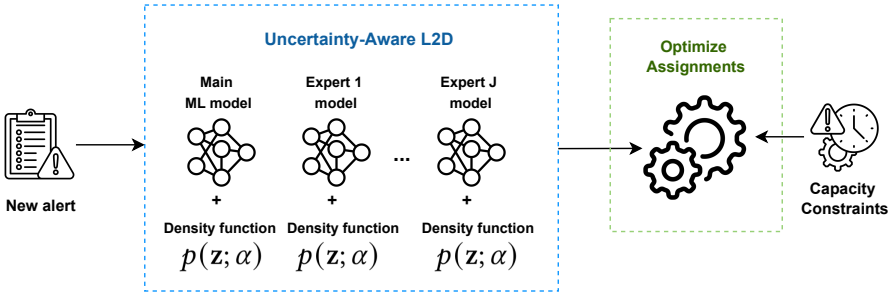

Figure 1: Uncertainty-aware L2D.

Density-softmax. The density-softmax model (Bui & Liu, 2024) consists of a feature extractor, a density estimator based on normalizing flows (**NF**) (Dinh et al., 2017), and a classifier. The feature extractor f maps an input $\mathbf{x}$ into a latent space representation $\mathbf{z}$, where the NF model estimates the log-likelihood $\log p(\mathbf{z};\alpha)$ under the training distribution, with α representing the model parameters. The NF model is optimized via maximum likelihood estimation (**MLE**) and provides exact log-likelihoods. Importantly, Bui & Liu (2024) show that the density-softmax approach achieves lower test-time latency than state-of-the-art uncertainty estimation methods, making it a lightweight addition to an L2D system with minimal computational overhead. To avoid numerical issues from unbounded log-likelihoods, we scale them following Bui & Liu (2024): at test time, log-likelihoods are divided by the maximum training log-likelihood and then min–max normalized to $(0, 1]$. The scaled likelihood $p(\mathbf{z};\alpha)$ adjusts the classifier g's logits, modifying the predictive probability as follows:

$$p(y = i|\mathbf{x}) = \frac{\exp(p(\mathbf{z};\alpha)\ g_i(\mathbf{x}))}{\sum_{j=1}^{K} \exp(p(\mathbf{z};\alpha)\ g_j(\mathbf{x}))}, \tag{1}$$

where g_i is the logit of classifier g for class i. For OOD instances in low-density regions, this adjustment scales down the logits, resulting in predictive probabilities that reflect higher uncertainty and improve calibration. In the binary classification setting, a sigmoid function replaces the softmax in Equation 1. These calibrated probability estimates are then used in the optimization step described in Section 5.

L2D Formulation. Our L2D system is built using the OvA framework introduced by Verma et al. (2023), which aims to minimize the 0-1 loss defined as

$$L_{0-1}(h, r) = \mathbb{E}_{\mathbf{x},y,\{m_1,\ldots,m_J\}}\left[\mathbb{I}[r(\mathbf{x}) = 0]\ \mathbb{I}[h(\mathbf{x}) \neq y] + \sum_{j=1}^{J} \mathbb{I}[r(\mathbf{x}) = j]\ \mathbb{I}[m_j(\mathbf{x}) \neq y]\right],$$

where $h : \mathcal{X} \to \mathcal{Y}$ is the classifier, $r : \mathcal{X} \to \{0, 1, ..., J\}$ is the rejector, and m_j is the prediction from expert $j \in \{1, ..., J\}$. When $r(\mathbf{x}) = 0$, the classifier h makes the decision, and when $r(\mathbf{x}) = j$, the decision is deferred to the j-th human expert. Verma et al. (2023) show that minimizing the 0-1 loss leads to the following Bayes optimal classifier and rejector:

$$h^*(\mathbf{x}) = \arg\max_{y\in\mathcal{Y}} \mathbb{P}(y \mid \mathbf{x}), \tag{2}$$

$$r^*(\mathbf{x}) = \begin{cases} 0 \text{ if } \max\limits_{y\in\mathcal{Y}} \mathbb{P}(h^*(\mathbf{x}) = y \mid \mathbf{x}) \geq \max\limits_{j\in\{1,\ldots,J\}} \mathbb{P}(m_j(\mathbf{x}) = y \mid \mathbf{x}) \\ \arg\max\limits_{j\in\{1,\ldots,J\}} \mathbb{P}(m_j(\mathbf{x}) = y \mid \mathbf{x}) \text{ otherwise.} \end{cases} \tag{3}$$

To approximate this solution, the OvA framework constructs the classifier h and rejector r using $|\mathcal{Y}|$ functions $g_k : \mathcal{X} \to \mathbb{R}$, where each function g_k is related to the probability of the instance belonging to class k, and J functions $g_{\perp,j} : \mathcal{X} \to \mathbb{R}$ related to the likelihood of each expert $j \in J$ making the correct decision. These $|\mathcal{Y}|+J$ functions are combined in the OvA surrogate loss, defined as

$$\Psi^J_{\text{OvA}} = \Phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \Phi[-g_{y'}(\mathbf{x})] + \sum_{j=1}^{J} \Phi[-g_{\perp,j}(\mathbf{x})] + \sum_{j=1}^{J} \mathbb{I}[m_j = y](\Phi[g_{\perp,j}(\mathbf{x})] - \Phi[-g_{\perp,j}(\mathbf{x})]), \tag{4}$$

where m_j represents the j-th expert's decision, and $\Phi : \{\pm 1\} \times \mathbb{R} \to \mathbb{R}_+$ is a strictly proper binary surrogate loss function.

Verma et al. (2023) prove that the minimizer of the pointwise inner Ψ^J_{OvA}-risk is composed of the minimizers of the inner Φ-risk for each binary classification problem, meaning each g_y can be trained independently on its corresponding OvA binary classification task by minimizing a proper binary loss (such as the log-loss), and calibrated estimates for a given instance's probability of belonging to each class k are given by $\psi^{-1}(g_y)$, where ψ^{-1} is the inverse link function for the proper binary surrogate loss Φ (in the case of the log-loss, ψ^{-1} is given by the sigmoid function σ). Similarly, each $g_{\perp,j}$ can be trained independently on the subset of data where predictions from expert j are available, and calibrated estimates for the probability that expert j will predict correctly on instance i are given by $\psi^{-1}(g_{\perp,j})$.

Aleatoric and Epistemic Uncertainty. By using the density-softmax approach, each of the $|\mathcal{Y}|+J$ binary classifiers is paired with a density function to improve uncertainty estimation. The allocation of instances to decision-makers relies on these correctness estimates, which must be calibrated and reflect instance-specific uncertainty. Equation 1 contains both aleatoric and epistemic uncertainty. To disentangle these, the likelihood value from the density model must be separated from that of the main classifier, allowing it to represent epistemic uncertainty. In regions of high epistemic uncertainty, it is reasonable to downweigh the classifier's predictions, as the classifier relies solely on training data and may not generalize well to OOD instances. Human experts, however, can adapt, learn, and access additional information not available to machine learning models, meaning we can expect them to outperform the classifier on OOD instances. Consequently, we use the human expert's average correctness on the training data as a proxy for their expected performance. Specifically, for instances with high epistemic uncertainty, the model blends the predicted logits of the binary classifier for expert j ($\hat{g}_{\perp,j}$) with the expert's average probability of correctness on the training data ($\hat{\text{p}}_{j,\text{avg}}$), using the density score $p(\mathbf{z};\alpha)$ from the normalizing flows model ($p(\mathbf{z};\alpha) \in (0,1]$). The adjusted logits are defined as

$$p(\mathbf{z};\alpha)\hat{g}_{\perp,j} + (1 - p(\mathbf{z};\alpha))\sigma^{-1}(\hat{\text{p}}_{j,\text{avg}}), \tag{5}$$

where σ represents the sigmoid function, and are fed into the softmax function in Equation 1. This adjustment ensures that the model reflects a more realistic estimate of expert performance in high epistemic uncertainty instances, balancing the model's own estimates with the historical expert performance.

4 Conformal Prediction for HAIC

In this section, we introduce a different approach to enhancing HAIC systems by integrating uncertainty estimation. We hypothesize that, while L2D methods perform well on familiar, in-distribution cases, these should avoid handling instances lying outside the training distribution. For such OOD cases, the models' probability estimates can be poorly calibrated, resulting in suboptimal decisions. In these scenarios, human experts are better suited to make predictions, as they are able to generalize and adapt by leveraging broader contextual knowledge. Furthermore, in high-stakes environments, it may be preferable, or even required, that OOD instances be reviewed by human experts to ensure accountability and explainability. To meet these requirements, we propose a system that defers OOD instances to human experts through ReL, while relying on L2D for in-distribution cases (Figure 2).

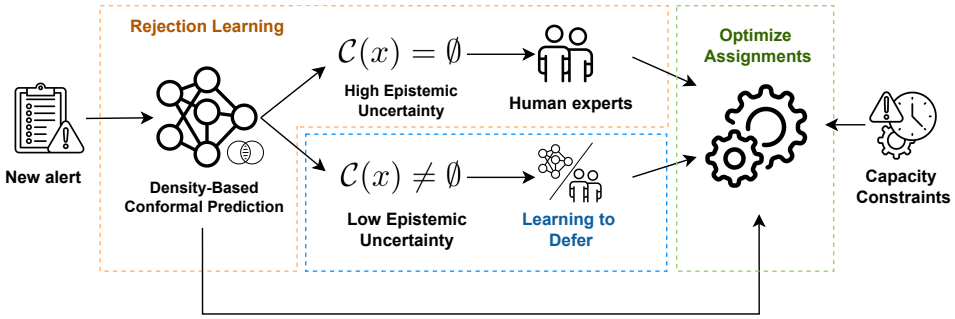

Figure 2: Conformal prediction for HAIC.

Density-based conformal prediction. To implement this system, we use density-based conformal prediction (Hechtlinger et al., 2018; Messoudi et al., 2020) to assess epistemic uncertainty on a per-instance basis, allowing to determine if an instance is likely to be OOD. Density-based conformal prediction uses a two-step training process involving a proper training set $D^{tr} = (X^{tr}, Y^{tr})$ and a calibration set $D^{cal} = (X^{cal}, Y^{cal})$. First, a class-conditional density estimator $\hat{p}(\mathbf{x}|y)$ is built using D^{tr}. The calibration set is then used to determine the empirical $1-\alpha$ quantile $\hat{q}_y$ of the density values for each class,

$$\hat{q}_y = \sup\left\{ t : \frac{1}{n_y} \sum_{(\mathbf{x}_i, y_i) \in D_y^{cal}} \mathbb{I}(\hat{p}(\mathbf{x}_i|y) \geq t) \geq 1-\alpha \right\}, \tag{6}$$

where $\mathbb{I}(\hat{p}(\mathbf{x}_i|y) \geq t)$ equals 1 if the estimated probability $\hat{p}(\mathbf{x}_i|y)$ is no less than t, and 0 otherwise, $D_y^{cal} = \{(\mathbf{x}_i, y_i) \in D^{cal} : y_i = y\}$ is the subset of calibration examples in class y, and n_y is the cardinality of D_y^{cal}. This quantile effectively acts as a threshold, allowing the model to define, for any new observation x_{n+1}, a prediction set

$$\mathcal{C}_\alpha(\mathbf{x}_{n+1}) = \{y \in \mathcal{Y} : \hat{p}(\mathbf{x}_{n+1}|y) \geq \hat{q}_y\}, \tag{7}$$

which includes all classes y for which the observed density is above the threshold. Hechtlinger et al. (2018) show that $|P(y \in \mathcal{C}_\alpha(\mathbf{x}_{n+1})) - (1-\alpha)| \to 0$ as $\min_y n_y \to \infty$, ensuring the asymptotic validity of the model. The training and prediction algorithms for density-based conformal prediction are described in Section B of the Appendix. At inference, the additional computation is limited to evaluating density scores and comparing them to pre-computed thresholds for each class. In binary settings, this adds negligible overhead, but in multi-class scenarios with many labels (e.g., image classification), the cost scales with the number of classes and can become more substantial.

Aleatoric and epistemic uncertainty. Figure 2 provides a schematic representation of our system. Instances with high epistemic uncertainty, which deviate significantly from the training distribution, are classified as empty sets and deferred to human experts. For non-empty predictions, which indicate in-distribution instances, the L2D system (as previously formulated in Section 3) assigns instances by using the correctness estimates for both the ML model and human experts. The assignments are then optimized taking into account human work capacity constraints (Section 5).

5 Cost and Capacity Constraints

Cost-sensitive learning. For completeness, we adapt our methods to cost-sensitive scenarios, assuming each instance i may incur a different error cost c_i, should it be misclassified. We follow the approach proposed by Zadrozny et al. (2003), Elkan (2001), and used by Alves et al. (2024) in an L2D setting. This approach redefines the training distribution by weighting each instance according to its misclassification cost.

In standard learning setups, we typically minimize the expected error rate, represented by $\mathbb{E}_{(\mathbf{x},y)\sim D}[\mathbb{I}_{h(\mathbf{x})\neq y}]$, where D is the data distribution and $h(\mathbf{x})$ is the predicted label. In cost-sensitive scenarios, the objective

shifts to minimizing the expected misclassification cost, defined as $\mathbb{E}_{x,y,c\sim D}[c\,\mathbb{I}_{h(\mathbf{x})\neq y}]$, where c is the instance-specific cost. To achieve this, Zadrozny et al. (2003) show that we can redefine the data distribution as

$$\tilde{D}(\mathbf{x}, y) = \frac{c}{\mathbb{E}_{c\sim D}[c]} D(\mathbf{x}, y, c), \tag{8}$$

and that, under this modified distribution, the expected error rate is

$$\mathbb{E}_{\mathbf{x},y\sim\tilde{D}}[\mathbb{I}_{h(\mathbf{x})\neq y}] = \frac{1}{\mathbb{E}_{c\sim D}[c]} \mathbb{E}_{\mathbf{x},y,c\sim D}[c\,\mathbb{I}_{h(\mathbf{x})\neq y}]. \tag{9}$$

Thus, minimizing the error rate under $\tilde{D}$ is equivalent to minimizing the expected misclassification cost under the original distribution D. In practical terms, this entails modifying the empirical loss function $\mathcal{L} = \sum_i^N \ell\big(h(\mathbf{x_i}), y_i\big)$, where ℓ is the pointwise loss, by weighing each instance by its associated cost (Zadrozny et al., 2003; Elkan, 2001), resulting in a loss function aligned with the cost-sensitive objective $\mathcal{L} = \sum_{i=1}^N c_i \ell\big(h(\mathbf{x_i}), y_i\big)$. To implement this approach in our OvA framework, we modify the standard log-loss functions of each of the $|\mathcal{Y}|+J$ binary classifiers. The weighted loss functions are described in detail in Equations 13 and 15 in the Appendix.

Capacity constraints. We address human work capacity constraints in both assignment systems. For the density-softmax approach, which is a pure L2D system, we apply the strategy proposed by Alves et al. (2024), optimizing assignments under capacity constraints. For the conformal prediction approach, we extend this framework by creating a method that dynamically balances L2D with ReL while respecting capacity limitations.

Rather than processing the entire dataset at once, we impose constraints over batches of instances. This approach aligns with practical scenarios where data is processed in batches and allows the system to adapt dynamically to varying workload demands. We define a batch vector $\mathbf{b}$ that assigns each instance i to a batch b containing n_b instances, and a human capacity matrix $\mathbf{H}$, where $H_{b,j}$ denotes the maximum number of cases that can be deferred to expert j within batch b.

Each instance $\mathbf{x_i}$ is subject to an action $a_i \in \{1, ..., J + |\mathcal{Y}|\}$ with an associated estimated probability of correctness $\hat{\mathbb{P}}(\text{correct}|\mathbf{x}_i, a_i)$, obtained via the OvA classifiers. If $a_i = y+1$, class y is automatically predicted with an estimated probability of correctness of $\hat{\mathbb{P}}(\text{correct}|\mathbf{x}_i, a_i) = \sigma(g_y)$; while $a_i = j+|\mathcal{Y}|$ indicates deferral to the jth human expert with an estimated probability of correctness of $\hat{\mathbb{P}}(\text{correct}|\mathbf{x}_i, a_i) = \sigma(g_{\perp,j})$. A set of n_b assignments can then be represented as a matrix $A \in \{0,1\}^{n_b\times(J+|\mathcal{Y}|)}$, where A_{i,a_i} denotes whether action a_i is taken for instance i. We can then frame the assignment optimization as

$$\boldsymbol{A}^* = \underset{A\in\{0,1\}^{n_b\times(J+|\mathcal{Y}|)}}{\operatorname{argmax}} \sum_{i=1}^{n_b} \sum_{a_i=1}^{J+|\mathcal{Y}|} \hat{\mathbb{P}}(\text{correct}|\mathbf{x_i}, a_i) A_{i,a_i}, \tag{10}$$

subject to two constraints: (1) $\sum_{i=1}^{n_b} A_{i,a_i} = H_{b,a_i}$ ensuring each expert meets his capacity, and (2) $\sum_{a_i=1}^{J+|\mathcal{Y}|} A_{i,a_i} = 1$ to ensure each instance has a unique assignment.

For the conformal prediction approach, balancing between L2D and ReL is necessary. The indicator function $\mathbb{I}_{\mathcal{C}_\alpha(\mathbf{x_i})\neq\emptyset}$ indicates whether an instance will be assigned through L2D or ReL, based on the coverage level α for the conformal prediction method. We then optimize this balance alongside the assignment matrix $\boldsymbol{A}$, with the following objective:

$$\begin{aligned}(\boldsymbol{A}^*,\, \alpha^*) = \underset{\mathbf{A}\in\{0,1\}^{n_b\times(J+|\mathcal{Y}|)},\alpha}{\operatorname{argmax}} \sum_{i=1}^{n_b} \sum_{a_i=1}^{J+|\mathcal{Y}|} \Bigg[&\mathbb{I}_{\mathcal{C}_\alpha(\mathbf{x_i})\neq\emptyset}\ \frac{1}{w_i}\ \hat{\mathbb{P}}(\text{correct} \mid \mathbf{x_i}, a_i) \\ &+ \mathbb{I}_{\mathcal{C}_\alpha(\mathbf{x_i})=\emptyset}\ \hat{\mathbb{P}}(\text{correct} \mid X_{\text{train}}, a_i) \Bigg] A_{i,a_i}.\end{aligned} \tag{11}$$

This optimization is subject to the same constraints mentioned above, and an additional constraint ensuring that when ReL is applied ($\mathcal{C}_\alpha(\mathbf{x_i}) = \emptyset$), the instance is deferred to a human expert, preventing OOD instances

from being handled by the model. Additionally, we downweigh model predictions for some instances (through w_i) based on the coverage level α at which they fall into the null set, and if the proportion of null set predictions exceeds the expected rate from training data. The downweighing is described in detail in Section C. This adaptation allows the system to respond to distribution shifts by prioritizing expert work capacity for OOD instances when required. We solve the assignment problems 10 and 11 using the CP-SAT solver from Google Research's OR-Tools (Perron & Didier).

6 Experimental Setup

Reproducibility. The code used for the experiments is available at https://anonymous.4open.science/r/Uncertainty-Aware-Systems-for-Human-AI-Collaboration-FB62/. The dataset is also publicly available (Alves et al., 2025a).

Dataset. As discussed in Section 1, datasets with real human predictions are scarce and collecting expert annotations is expensive. OpenL2D (Alves et al., 2025a) addresses this by providing an open-source framework for generating highly customizable synthetic experts with control over feature dependence, bias towards protected attributes, and performance levels. In this work, we use the FiFAR dataset (Alves et al., 2025a), generated with OpenL2D from the publicly available bank-account-fraud (**BAF**) dataset (Jesus et al., 2022). FiFAR simulates a team of fraud analysts whose decision-making properties were validated against the literature and compared to real analysts in terms of performance, inter- and intra-rater agreement, and decision processes. Our experiments use a team of five analysts, for which the dataset provides predictions on all 30K instances flagged as fraudulent by a fraud detection model, representing an "alert-review" scenario where analysts review high-risk applications. Each instance contains the original **BAF** features, expert predictions, and the fraud model's score. The **BAF** dataset spans 8 months: the first 3 were used in Alves et al. (2025a) to train the fraud detection model, while the flagged 30K instances come from months 4–8 and comprise the FiFAR dataset.

Misclassification costs. In account opening fraud detection, a positive prediction leads to application rejection, while a negative prediction results in account opening. Consequently, misclassification costs differ, as false positives (rejecting legitimate applications) lead to customer loss, whereas false negatives (accepting fraudulent applications) may result in financial losses for the bank. Consequently, fraud detection requires balancing the costs of false positives (c_{FP}) and false negatives (c_{FN}), making commonly used metrics like accuracy inadequate. In our cost-sensitive task, assuming no cost for correct classifications, the objective function to be minimized is the expected misclassification cost,

$$\frac{1}{N}\sum_{i=1}^{N}\left(\lambda\mathbb{I}[y_i = 0 \wedge \hat{y}_i = 1] + \mathbb{I}[y_i = 1 \wedge \hat{y}_i = 0]\right), \tag{12}$$

where $\lambda = c_{FP}/c_{FN}$. As only the ratio λ is relevant for minimization purposes, we choose to arbitrarily set $c_{FP} = \lambda$ and $c_{FN} = 1$ for our misclassification cost re-weighting approach. For the FiFAR dataset, λ is derived from the alert model threshold selected under the Neyman–Pearson criterion, which maximizes recall at a 5% false positive rate, yielding $\lambda_t = 0.057$ (Alves et al., 2025a).

Noise injection. The two foundational works in density-based conformal prediction (Hechtlinger et al., 2018; Messoudi et al., 2020) evaluate robustness by injecting Gaussian noise to simulate distribution shifts and OOD data. The work introducing density-softmax (Bui & Liu, 2024) similarly applies Gaussian noise among other perturbations in its empirical evaluation. Such noise-based stress tests are standard in uncertainty estimation, where the goal is to assess performance and calibration on borderline or clearly OOD instances (Hein et al., 2019; Hendrycks & Gimpel, 2017; Depeweg et al., 2018). In our case, the introduction of controlled Gaussian noise allows isolating the contributions of each system component, including calibration and performance from density models, and OOD detection and deferral from conformal prediction.

In our setup, we modify the test set (last month of data) by introducing noise to 20% of instances under three configurations: low, medium, and high noise. For each configuration, numerical features are standardized, perturbed with Gaussian noise (σ = 1.0, 1.5, 2.0), and mapped back to the original scale, with discrete values rounded and bounded features clipped. Categorical and binary features are randomly switched with

probabilities 0.3, 0.4, and 0.5, respectively. To account for statistical variation, each configuration is repeated five times with different seeds.

Data availability and capacity constraints. We simulate scenarios with variable amounts of expert labels across four data conditions: all instances labeled by all experts, a common but unrealistic assumption in prior L2D methods (Verma et al., 2023; Hemmer et al., 2022), and subsets where experts label only 1/5, 1/20, or 1/40 of the data. These subsets are randomly distributed across experts using five random seeds per condition. In both systems, the expert-specific models are trained on the subset of data with labels for that expert, and in the density-softmax approach, the same subsets train the density models. We impose uniform work capacity constraints on experts by varying deferral rates (10%-50%) to evaluate the system's ability to manage OOD instances across four experimental variables: noise (low, medium, high), number of experts (1-5), deferral rates, and data availability. This leads to 300 different test settings, so results are analyzed for a key subset of representative settings to assess the system's robustness under these constraints.

Baselines. We consider three baseline approaches. First, for the **L2D baseline**, we implement the OvA L2D algorithm by Verma et al. (2023) using LightGBM (Ke et al., 2017) models for each binary classification task, omitting the distance-awareness feature to isolate the impact of our proposed contributions. This setup also serves as a baseline for the conformal prediction approach, as the same L2D method handles instances not identified as OOD. Second, in the **ReL baseline**, density-based conformal prediction measures uncertainty to decide whether to defer instances to human experts, deferring randomly when predicted as the null set and assigning the remainder to the ML model, with coverage level α selected to match expert work capacity. This baseline therefore serves as a direct measure of the effectiveness of conformal prediction in detecting and deferring OOD instances, independent of any optimization in the expert assignment step. Third, a **random assignment baseline** allocates a randomly selected subset of test instances to experts, ensuring capacity constraints are not violated.

System Training. Both systems share a single MLP trained on the data from months 4 to 6 to minimize weighted log-loss, with month 7 used for validation. The MLP's penultimate dense layer is used as a feature extractor to map inputs into a continuous latent space for density estimation. In the density-based conformal prediction method, class-conditional densities are estimated using kernel density estimation (**KDE**) (Rosenblatt, 1956), and empirical $1 - \alpha$ quantiles are computed on calibration data from month 7. For the density-softmax approach, RealNVP (Dinh et al., 2017) models are trained on the extracted features to estimate likelihoods, using subsets of the data labeled by each expert to reflect specific data availability scenarios. Both systems use LightGBM (Ke et al., 2017) models for the binary classifiers in L2D: the main classifier is trained to predict the true labels, while expert-specific models estimate correctness. Because human experts often have access to the alert model's score, the human correctness models include this score as an additional input feature. All classifiers are trained to minimize the weighted log-loss, as described in Section A.1. Further details on the training process and hyper-parameter selection are available in Section A of the Appendix.

7 Results

Calibration improvements from the density-softmax approach. To assess how the density model interacts with predictions, we first extract the features from the penultimate layer of the MLP trained for feature extraction and project them in two dimensions with t-SNE. The first panel on the left of Figure 3 shows the resulting feature map in the high-noise test setting, where in-distribution and noisy (OOD) instances are clearly separated. The second panel displays density scores estimated by the normalizing flows density model, which assign high values to in-distribution points and low values to noisy data, confirming the model's ability to capture the data likelihood under the prior distribution. The last two panels shows the effect on classifier predictions: raw LightGBM probabilities (third panel) can be overconfident on OOD regions, but after adjustment with density scores (forth panel), probabilities in those regions are pulled closer to 0.5. This demonstrates how the density-softmax approach improves calibration by adjusting predictions on data unlikely to belong to the training distribution.

Incorporating density models into our L2D classifiers consistently enhances calibration across all settings. Figure 4 reports the mean expected calibration error (**ECE**) with 95% confidence intervals, accounting for

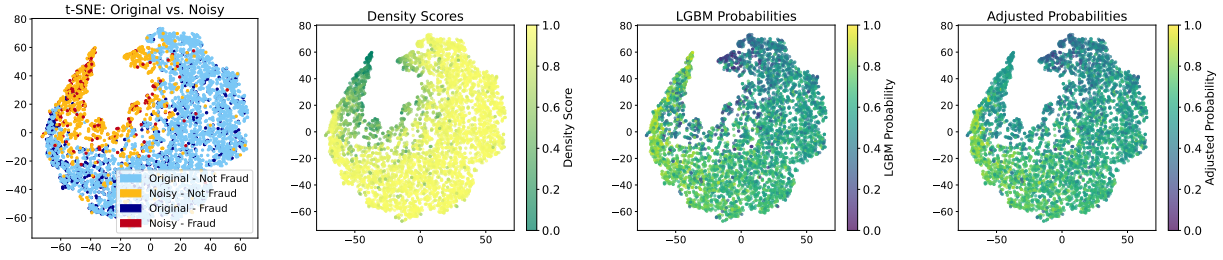

Figure 3: Left to right: t-SNE visualization of extracted features, density scores, raw LightGBM probabilities, and adjusted probabilities with density-softmax under the high-noise test setting.

the randomness in noise injection and training set selection. Here, data availability is the fraction of training data with a given expert's predictions: 1/1 means full coverage, while 1/40 means each expert labels only 1/40 of the data, and their models (both LightGBM and density) are trained on that subset. The figure shows ECE for the models that predict expert correctness, which are used in our system (Section 3) to assign instances to the most reliable expert. Calibration of these models is crucial, since we directly compare their predicted probabilities of correctness to decide which expert should handle a given instance.

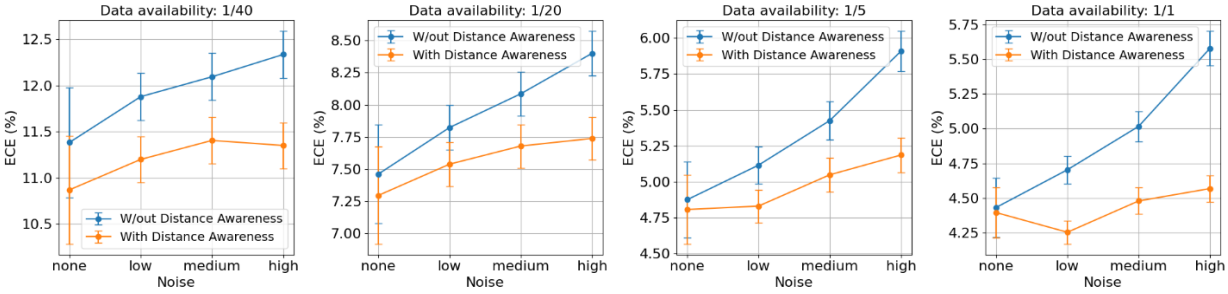

Figure 4: Comparison of the ECE of expert correctness models with and without distance awareness across different data availability and noise scenarios.

The results show that without density models, ECE increases sharply as noise grows. On the other hand, incorporating density models significantly reduces the growth of ECE, creating a widening gap in performance as noise increases. The benefit is especially pronounced under higher data availability, where having more expert predictions allows for more accurate modeling of human behavior and better estimation of density scores, which in turn improves discrimination between in-distribution and OOD data.

This improvement arises because density models balance the expert's predicted probability of correctness with their average correctness on the training data in regions of high epistemic uncertainty (as detailed in Section 3), yielding more reliable predictions even for instances that deviate from the training distribution.

Epistemic uncertainty and calibration in the conformal prediction approach. In our conformal prediction approach (Section 4), an empty set prediction indicates high epistemic uncertainty, controlled by the coverage level α: a higher α leads to more empty set predictions overall. Empty-set predictions are handled via a ReL strategy, while non-empty predictions are assigned through the L2D system. Ideally, we would expect the classifiers to have low ECE on non-empty predictions (in-distribution instances, where probability estimates should be reliable) and high ECE on empty-set predictions (OOD instances, which are deferred to human experts). To examine this, Figure 5 shows the ECE of instances predicted as empty sets across coverage levels α. The right plot averages results across expert models with 95% confidence intervals for the mean; ECE is initially high at small α values but decreases across all noise settings as α increases. This is expected: when α is near 0, only the most extreme OOD data is excluded, yielding few empty-set predictions with poor calibration. As α rises, the threshold for OOD detection becomes less

conservative, more instances are categorized as empty sets, and calibration improves as these instances more closely resemble the training distribution.

These results highlight the need to choose an optimal coverage level α to balance OOD detection against errors on in-distribution data. In practice, this means optimizing the use of L2D on in-distribution instances and deferring OOD cases to human experts. Consequently, jointly optimizing α and the assignment matrix (Equation 11) is necessary. Results on tuning α are presented in Section D of the Appendix.

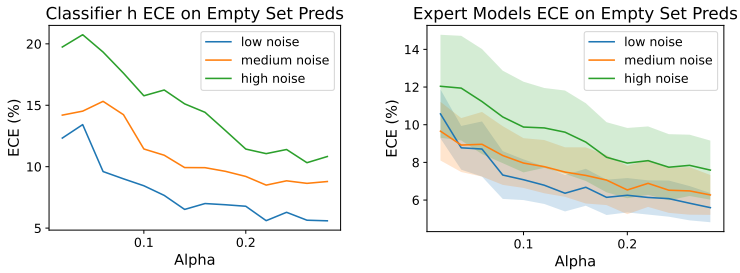

Figure 5: ECE of the models on empty set predictions.

Misclassification cost analysis. We evaluate the misclassification cost across different deferral strategies and experimental settings. Table 1 compares five strategies (Density-Softmax (**DS**), Conformal Prediction (**CP**), L2D, ReL, and random assignment) showing representative settings from the 300 test cases described in Section 6, covering three noise levels (low, medium, high), two deferral rates (20% or 40%), 1, 3 or 5 experts, and two data availability scenarios (full data and one-fifth). Each setting is run five times with distinct seeds for data sampling and noise generation, and results are reported with 95% confidence intervals for the mean. Tables with the remaining 300 settings are available at https://anonymous.4open.science/r/Uncertainty-Aware-Systems-for-Human-AI-Collaboration-FB62/results/results.ipynb.

Across all noise levels, both DS and CP outperform the baselines on average. In low-noise scenarios, DS consistently achieves the lowest misclassification cost, benefiting from its gradual adjustment of probability estimates: by slightly adjusting overconfident predictions in uncertain regions, DS produces more accurate estimates of expert correctness and thus better assignment decisions. Under medium and high noise, CP becomes the strongest strategy. Its hard cutoff mechanism—deferring all instances deemed OOD to experts—provides a clear advantage when noisy data is easier to identify, while DS may still allow some OOD instances to be handled by the classifier.

The most informative baseline is ReL. Both ReL and CP use density-based conformal prediction to flag empty-set (OOD) cases, but differ in how the coverage level α is set. ReL fixes α to match expert capacity, rejecting as many instances as possible up to that limit. CP instead optimizes α to balance assignments between L2D and ReL: more OOD data pushes assignments toward ReL, while more in-distribution data favors L2D. This optimization enables CP to interpolate between the two strategies depending on the data distribution, and the CP–ReL performance gap therefore quantifies the added value of assignment and coverage optimization.

The remaining baselines underperform: random assignment ignores expertise, and OvA L2D lacks density information to assign instances correctly, highlighting the importance of having calibrated estimates of correctness in L2D. The DS vs. L2D comparison shows that incorporating density scores improves expert assignment in the presence of OOD data.

8 Conclusions and Future Work

We presented two novel uncertainty-aware approaches to human-AI collaboration (HAIC) that address a key weakness of traditional learning-to-defer (L2D) methods: their reliance on machine learning models whose correctness estimates degrade under distribution shifts, leading to poor assignments. The proposed methods

Table 1: Misclassification cost for the representative settings. The best result is in bold and the second best is underlined.

Setting				Deferral Strategy				
Noise	NE	DR	DA	**CP**	**DS**	**L2D**	**ReL**	**Random**
low	1	20	1/1	4.81 ± 0.11	**4.81 ± 0.17**	5.09 ± 0.13	5.11 ± 0.07	5.20 ± 0.15
low	1	20	1/5	4.85 ± 0.16	**4.83 ± 0.22**	5.18 ± 0.16	5.11 ± 0.07	5.23 ± 0.11
low	1	40	1/1	**4.42 ± 0.09**	4.48 ± 0.11	5.00 ± 0.13	4.77 ± 0.07	4.97 ± 0.13
low	1	40	1/5	**4.40 ± 0.13**	4.41 ± 0.09	4.89 ± 0.15	4.77 ± 0.07	4.99 ± 0.09
low	3	20	1/1	4.78 ± 0.13	**4.72 ± 0.11**	5.20 ± 0.13	5.14 ± 0.10	5.25 ± 0.12
low	3	20	1/5	4.91 ± 0.15	**4.90 ± 0.18**	5.23 ± 0.11	5.14 ± 0.11	5.28 ± 0.18
low	3	40	1/1	4.40 ± 0.07	**4.35 ± 0.09**	5.11 ± 0.13	4.77 ± 0.07	5.12 ± 0.16
low	3	40	1/5	4.53 ± 0.26	**4.45 ± 0.26**	5.03 ± 0.16	4.76 ± 0.08	5.16 ± 0.15
low	5	20	1/1	4.76 ± 0.15	**4.70 ± 0.09**	5.20 ± 0.13	5.09 ± 0.06	5.30 ± 0.21
low	5	20	1/5	4.85 ± 0.10	**4.84 ± 0.16**	5.19 ± 0.14	5.12 ± 0.04	5.25 ± 0.12
low	5	40	1/1	4.46 ± 0.10	**4.44 ± 0.09**	4.71 ± 0.13	4.79 ± 0.09	5.11 ± 0.11
low	5	40	1/5	4.48 ± 0.16	**4.42 ± 0.16**	4.80 ± 0.10	4.79 ± 0.11	5.07 ± 0.18
medium	1	20	1/1	**4.82 ± 0.07**	4.96 ± 0.09	5.32 ± 0.08	4.85 ± 0.06	5.39 ± 0.03
medium	1	20	1/5	**4.82 ± 0.09**	4.97 ± 0.15	5.37 ± 0.06	4.85 ± 0.06	5.46 ± 0.15
medium	1	40	1/1	**4.41 ± 0.05**	4.61 ± 0.10	5.20 ± 0.08	4.49 ± 0.09	5.13 ± 0.09
medium	1	40	1/5	**4.46 ± 0.12**	4.60 ± 0.12	5.08 ± 0.11	4.49 ± 0.09	5.17 ± 0.24
medium	3	20	1/1	**4.85 ± 0.09**	4.91 ± 0.09	5.42 ± 0.08	5.01 ± 0.13	5.47 ± 0.12
medium	3	20	1/5	5.00 ± 0.11	5.08 ± 0.19	5.39 ± 0.08	**4.99 ± 0.14**	5.50 ± 0.14
medium	3	40	1/1	**4.41 ± 0.08**	4.52 ± 0.06	5.29 ± 0.08	4.55 ± 0.11	5.39 ± 0.13
medium	3	40	1/5	4.56 ± 0.19	4.70 ± 0.28	5.20 ± 0.15	**4.50 ± 0.09**	5.40 ± 0.12
medium	5	20	1/1	4.86 ± 0.07	4.94 ± 0.13	5.41 ± 0.08	**4.86 ± 0.05**	5.51 ± 0.08
medium	5	20	1/5	**4.86 ± 0.12**	5.00 ± 0.09	5.36 ± 0.14	4.87 ± 0.07	5.47 ± 0.11
medium	5	40	1/1	**4.37 ± 0.08**	4.51 ± 0.07	5.03 ± 0.08	4.54 ± 0.11	5.35 ± 0.14
medium	5	40	1/5	**4.47 ± 0.12**	4.60 ± 0.16	5.06 ± 0.16	4.53 ± 0.12	5.27 ± 0.13
high	1	20	1/1	**4.62 ± 0.07**	5.00 ± 0.09	5.44 ± 0.13	4.86 ± 0.07	5.50 ± 0.14
high	1	20	1/5	**4.65 ± 0.14**	4.97 ± 0.19	5.49 ± 0.13	4.86 ± 0.07	5.48 ± 0.10
high	1	40	1/1	**4.21 ± 0.08**	4.53 ± 0.08	5.23 ± 0.13	4.39 ± 0.04	5.25 ± 0.22
high	1	40	1/5	**4.22 ± 0.10**	4.56 ± 0.13	5.15 ± 0.15	4.39 ± 0.04	5.25 ± 0.17
high	3	20	1/1	**4.74 ± 0.13**	5.04 ± 0.15	5.50 ± 0.13	5.09 ± 0.19	5.55 ± 0.16
high	3	20	1/5	**4.81 ± 0.17**	5.15 ± 0.16	5.49 ± 0.16	5.09 ± 0.15	5.56 ± 0.15
high	3	40	1/1	**4.29 ± 0.09**	4.53 ± 0.09	5.41 ± 0.13	4.46 ± 0.09	5.40 ± 0.17
high	3	40	1/5	**4.29 ± 0.14**	4.65 ± 0.28	5.32 ± 0.23	4.47 ± 0.06	5.42 ± 0.09
high	5	20	1/1	**4.72 ± 0.15**	5.04 ± 0.13	5.54 ± 0.13	4.85 ± 0.09	5.61 ± 0.15
high	5	20	1/5	**4.75 ± 0.17**	5.04 ± 0.17	5.49 ± 0.11	4.86 ± 0.10	5.59 ± 0.17
high	5	40	1/1	**4.25 ± 0.10**	4.57 ± 0.14	5.01 ± 0.13	4.42 ± 0.07	5.37 ± 0.19
high	5	40	1/5	**4.25 ± 0.10**	4.56 ± 0.13	5.10 ± 0.17	4.44 ± 0.06	5.41 ± 0.17

use uncertainty estimation to make HAIC systems more robust in dynamic environments while accounting for cost sensitivity, limited data, and human-capacity constraints. The first method, *Uncertainty-Aware L2D*, incorporates density information via normalizing flows to improve calibration and assignments. The second, *Conformal Prediction for HAIC*, combines L2D with rejection learning (ReL) by using density-based conformal scores to flag high-uncertainty cases for direct deferral, while remaining in-distribution instances are handled by L2D using estimated correctness for both machine and human experts.

Empirical results on a fraud detection task show that both systems leverage complementary strengths of humans and models under distribution shift. The uncertainty-aware L2D approach improved calibration and reduced misclassification costs on OOD data, while the CP approach performed best in high-noise settings by deferring high-uncertainty instances and maintaining calibration on in-distribution data. The results indicate that the relative gains of each system depend on the evaluation setting, highlighting the need to study uncertainty estimation directly in HAIC—an additional contribution of this work.

Future work should focus on incorporating dynamic expert feedback, for example by retraining on new expert-labeled data to adapt to changing expert strengths, and on integrating active learning with epistemic-uncertainty sampling to reduce labeling demands and improve scalability.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 2623–2631. ACM, 2019. doi: 10.1145/3292500.3330701. URL https://doi.org/10.1145/3292500.3330701.

Jean Alves, Diogo Leitão, Sérgio Jesus, Marco Sampaio, Javier Liébana, Pedro Saleiro, Mário Figueiredo, and Pedro Bizarro. A benchmarking framework and dataset for learning to defer in human-ai decision-making. *Scientific Data*, 12, 04 2025a. doi: 10.1038/s41597-025-04664-y.

Jean V. Alves, Diogo Leitão, Sérgio M. Jesus, Marco O. P. Sampaio, Javier Liébana, Pedro Saleiro, Mário A. T. Figueiredo, and Pedro Bizarro. Cost-sensitive learning to defer to multiple experts with workload constraints. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=TAvGZm2Rqb.

Jean V. Alves, Diogo Leitão, Sérgio Jesus, Marco O. P. Sampaio, Javier Liébana, Pedro Saleiro, Mário A. T. Figueiredo, and Pedro Bizarro. Financial Fraud Alert Review Dataset. 4 2025b. doi: 10.6084/m9.figshare.28351172.v1. URL https://springernature.figshare.com/articles/dataset/Financial_Fraud_Alert_Review_Dataset/28351172.

John O. Awoyemi, Adebayo O. Adetunmbi, and Samuel A. Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pp. 1–9, 2017. doi: 10.1109/ICCNI.2017.8123782.

Ha Manh Bui and Anqi Liu. Density-softmax: Efficient test-time model for uncertainty estimation and robustness under distribution shifts. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=lon750Kf7n.

Mohammad-Amin Charusaie, Hussein Mozannar, David A. Sontag, and Samira Samadi. Sample efficient learning of predictors that complement humans. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2972–3005. PMLR, 2022. URL https://proceedings.mlr.press/v162/charusaie22a.html.

C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16 (1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles (eds.), *Algorithmic Learning Theory*, pp. 67–82, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46379-7.

Maria De-Arteaga, Riccardo Fogliato, and Alexandra Chouldechova. A case for humans-in-the-loop: Decisions in the presence of erroneous algorithmic scores. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20. ACM, April 2020. doi: 10.1145/3313831.3376638. URL http://dx.doi.org/10.1145/3313831.3376638.

Dominik Dellermann, Philipp Ebel, Matthias Söllner, and Jan Marco Leimeister. Hybrid intelligence. *Bus. Inf. Syst. Eng.*, 61(5):637–643, 2019. doi: 10.1007/S12599-019-00595-2. URL https://doi.org/10.1007/s12599-019-00595-2.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1192–1201. PMLR, 2018. URL http://proceedings.mlr.press/v80/depeweg18a.html.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Charles Elkan. The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Conference on Artificial Intelligence: 4-10 August 2001; Seattle*, 1, 05 2001.

Andre Esteva, Brett Kuprel, Roberto Novoa, Justin Ko, Susan Swetter, Helen Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542, 01 2017. doi: 10.1038/nature21056.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Hamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46, 04 2014. doi: 10.1145/2523813.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4878–4887, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Sharad Goel, Ravi Shroff, Jennifer Skeem, and Christopher Slobogin. *The accuracy, equity, and jurisprudence of criminal risk assessment*. 05 2021. ISBN 9781788972819. doi: 10.4337/9781788972826.00007.

Prashant Gohel, Priyanka Singh, and Manoranjan Mohanty. Explainable AI: current status and future directions. *CoRR*, abs/2107.07045, 2021. URL https://arxiv.org/abs/2107.07045.

Alison Gopnik and Henry M. Wellman. Reconstructing constructivism: causal models, bayesian learning mechanisms, and the theory theory. *Psychological bulletin*, 138 6:1085–108, 2012. URL https://api.semanticscholar.org/CorpusID:2496804.

Varun Gulshan, Lily Peng, Marc Coram, Martin Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip Nelson, Jessica Mega, and Dale Webster. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316, 11 2016. doi: 10.1001/jama.2016.17216.

Yotam Hechtlinger, Barnabás Póczos, and Larry A. Wasserman. Cautious deep learning. *CoRR*, abs/1805.09460, 2018. URL http://arxiv.org/abs/1805.09460.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 41–50. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00013. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Hein_Why_ReLU_Networks_Yield_High-Confidence_Predictions_Far_Away_From_the_CVPR_2019_paper.html.

Martin E. Hellman. The nearest neighbor classification rule with a reject option. *IEEE Trans. Syst. Sci. Cybern.*, 6(3):179–185, 1970. doi: 10.1109/TSSC.1970.300339. URL https://doi.org/10.1109/TSSC.1970.300339.

Patrick Hemmer, Sebastian Schellhammer, Michael Vössing, Johannes Jakubik, and Gerhard Satzger. Forming effective human-ai teams: Building machine learning models that complement the capabilities of multiple experts. In Luc De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 2478–2484. ijcai.org, 2022. doi: 10.24963/IJCAI.2022/344. URL https://doi.org/10.24963/ijcai.2022/344.

Patrick Hemmer, Lukas Thede, Michael Vössing, Johannes Jakubik, and Niklas Kühl. Learning to defer with limited expert predictions. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 6002–6011. AAAI Press, 2023. doi: 10.1609/AAAI.V37I5.25742. URL https://doi.org/10.1609/aaai.v37i5.25742.

Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: a survey. *Mach. Learn.*, 113(5):3073–3110, 2024. doi: 10.1007/S10994-024-06534-X. URL https://doi.org/10.1007/s10994-024-06534-x.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, March 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05946-3. URL http://dx.doi.org/10.1007/s10994-021-05946-3.

Sérgio M. Jesus, José Pombal, Duarte M. Alves, André Ferreira Cruz, Pedro Saleiro, Rita P. Ribeiro, João Gama, and Pedro Bizarro. Turning the tables: Biased, imbalanced, dynamic tabular datasets for ML evaluation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d9696563856bd350e4e7ac5e5812f23c-Abstract-Datasets_and_Benchmarks.html.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3146–3154, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Vijay Keswani, Matthew Lease, and Krishnaram Kenthapadi. Towards unbiased and accurate deferral to multiple experts. In Marion Fourcade, Benjamin Kuipers, Seth Lazar, and Deirdre K. Mulligan (eds.), *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, pp. 154–165. ACM, 2021. doi: 10.1145/3461702.3462516. URL https://doi.org/10.1145/3461702.3462516.

Amir E. Khandani, Adlar J. Kim, and Andrew W. Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking Finance*, 34(11):2767–2787, 2010. ISSN 0378-4266. doi: https://doi.org/10.1016/j.jbankfin.2010.06.001. URL https://www.sciencedirect.com/science/article/pii/S0378426610002372.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

Diogo Leitão, Pedro Saleiro, Mário A. T. Figueiredo, and Pedro Bizarro. Human-ai collaboration in decision-making: Beyond learning to defer, 2022. URL https://arxiv.org/abs/2206.13202.

Daniele Lunghi, Alkis Simitsis, Olivier Caelen, and Gianluca Bontempi. Adversarial learning in real-world fraud detection: Challenges and perspectives. In *Proceedings of the Second ACM Data Economy Workshop*, DEC '23, pp. 27–33, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400708466. doi: 10.1145/3600046.3600051. URL https://doi.org/10.1145/3600046.3600051.

David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. *Advances in neural information processing systems*, 31, 2018.

Soundouss Messoudi, Sylvain Rousseau, and Sébastien Destercke. Deep conformal prediction for robust models. In Marie-Jeanne Lesot, Susana M. Vieira, Marek Z. Reformat, João Paulo Carvalho, Anna Wilbik, Bernadette Bouchon-Meunier, and Ronald R. Yager (eds.), *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 18th International Conference, IPMU 2020, Lisbon, Portugal, June 15-19, 2020, Proceedings, Part I*, volume 1237 of *Communications in Computer and Information Science*, pp. 528–540. Springer, 2020. doi: 10.1007/978-3-030-50146-4_39. URL https://doi.org/10.1007/978-3-030-50146-4_39.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David A. Sontag. Who should predict? exact algorithms for learning to defer to humans. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent (eds.), *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10520–10545. PMLR, 2023. URL https://proceedings.mlr.press/v206/mozannar23a.html.

Vu-Linh Nguyen, Sébastien Destercke, and Eyke Hüllermeier. Epistemic uncertainty sampling. *CoRR*, abs/1909.00218, 2019. URL http://arxiv.org/abs/1909.00218.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7599–7609. PMLR, 2020. URL http://proceedings.mlr.press/v119/perdomo20a.html.

Laurent Perron and Frédéric Didier. Cp-sat. URL https://developers.google.com/optimization/cp/cp_solver/.

Janis Postels, Hermann Blum, Cesar Cadena, Roland Siegwart, Luc Van Gool, and Federico Tombari. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *CoRR*, abs/2012.03082, 2020. URL https://arxiv.org/abs/2012.03082.

Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956. ISSN 00034851, 21688990. URL http://www.jstor.org/stable/2237390.

Waddah Saeed and Christian Omlin. Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273, 2023. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2023.110273. URL https://www.sciencedirect.com/science/article/pii/S0950705123000230.

Robin Senge, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier. Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences*, 255:16–29, 2014. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2013.07.030. URL https://www.sciencedirect.com/science/article/pii/S0020025513005410.

Sriram Somanchi, Samrachana Adhikari, Allen Lin, Elena Eneva, and Rayid Ghani. Early prediction of cardiac arrest (code blue) using electronic medical records. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 2119–2126, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2788588. URL https://doi.org/10.1145/2783258.2788588.

Hao Sun, Boris van Breugel, Jonathan Crabbé, Nabeel Seedat, and Mihaela van der Schaar. What is flagged in uncertainty quantification? latent density models for uncertainty categorization. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023,*

NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/0f0c4f3d83c58df58380af3b0729354c-Abstract-Conference.html.

Dharmesh Tailor, Aditya Patra, Rajeev Verma, Putra Manggala, and Eric T. Nalisnick. Learning to defer to a population: A meta-learning approach. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3475–3483. PMLR, 2024. URL https://proceedings.mlr.press/v238/tailor24a.html.

Rajeev Verma and Eric T. Nalisnick. Calibrated learning to defer with one-vs-all classifiers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22184–22202. PMLR, 2022. URL https://proceedings.mlr.press/v162/verma22c.html.

Rajeev Verma, Daniel Barrejón, and Eric T. Nalisnick. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent (eds.), *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11415–11434. PMLR, 2023. URL https://proceedings.mlr.press/v206/verma23a.html.

B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE International Conference on Data Mining*, pp. 435–442, 2003. doi: 10.1109/ICDM.2003.1250950.

A Training details

A.1 Learning to Defer

For the ML classifier h, we employ a LightGBM classifier (Ke et al., 2017), trained on the alerts raised from months four to six and validated on month 7. We perform hyperparameter optimization using TPE (Akiba et al., 2019), running the optimization process for 300 trials with 200 start-up trials, using the hyperparameter search space in Table 2. As described in Section 5, the goal of this search is to minimize the weighted log-loss

$$\mathcal{L}'_h\left(\{x_i, y_i, c_i\}_{i=1}^N, g\right) = \frac{1}{N}\sum_{i=1}^{N} c_i\Big[-y_i \log\Big(\psi^{-1}\big(g(x_i)\big)\Big) - (1-y_i)\log\Big(1-\psi^{-1}\big(g(x_i)\big)\Big)\Big], \tag{13}$$

where g is the scoring function we are training to predict y_i, ψ^{-1} is a well-defined inverse link function, and c_i is the instance-specific cost. Note that although in our experiments c_i can only have two possible values (c_{FP} or c_{FN}), this formulation can be applied to any misclassification cost structure.

Since false negatives carry a higher cost, we explore whether introducing a bias towards predicting fraud is beneficial for our model. This is done by testing different initial probability estimates for the base predictor in the boosting model. During model training, we perform an independent hyperparameter search over a range of values for the initial prediction value $g_{initial}$, from g_d to $g_d + 2$, in increments of 0.2, where g_d is the default initial prediction value

$$g_d = logit\left(\frac{\sum_i c_i y_i}{\sum_i c_i}\right). \tag{14}$$

In the first hyperparameter search, LightGBM classifiers with a maximum tree depth of 2 achieved the best performance. Therefore, we conducted a second search consisting of a total of 1700 trials, with 1500 startup trials, where we used the same hyperparameter search space detailed in Table 2, but fixing the maximum depth parameter at 2.

Table 2: ML classifier: lightGBM hyperparameter search space.

Hyperparameter	**Values**	**Distribution**
boosting_type	"dart"	
enable_bundle	[False, True]]	
n_estimators	[50, 250]	Uniform
max_depth	[2, 5]	Uniform
num_leaves	[100, 1000]	Uniform
min_child_samples	[5, 200]	Uniform
learning_rate	[0.001, 1]	Uniform
reg_alpha	[0.0001, 2]	Uniform
reg_lambda	[0.0001, 2]	Uniform

For the expert correctness models, we also use LightGBM classifiers (Ke et al., 2017). These models are trained on alerts raised during months four to six and validated on alerts from month seven. The objective is to predict the correctness of the experts' predictions, using only data for which human labels are available. Consequently, the amount of training data depends on the data availability scenario under consideration. Hyperparameter optimization is performed using TPE (Akiba et al., 2019), with a total of 120 trials, including 100 start-up trials. The hyperparameter search space is detailed in Table 3. The objective is to minimize the weighted log-loss, adapted to predict expert correctness instead of fraud labels. The loss function is defined as

$$\mathcal{L}'_j\left(\{S_i\}_{i=1}^N, g_{\perp,j}\right) = \frac{1}{N}\sum_{i=1}^{N} c_i \left[-\mathbb{I}[m_{i,j}=y]\log\left(\psi^{-1}\big(g_{\perp,j}(x_i)\big)\right) - \mathbb{I}[m_{i,j}\neq y]\log\left(1-\psi^{-1}\big(g_{\perp,j}(x_i)\big)\right)\right], \tag{15}$$

where $S_i = \{x_i, y_i, c_i, m_{i,j}\}$, and $m_{i,j}$ represents a prediction from expert j on instance x_i.

Table 3: Expert models: lightGBM hyperparameter search space.

Hyperparameter	**Values**	**Distribution**
boosting_type	"dart"	
enable_bundle	[False, True]]	
n_estimators	[50, 250]	Uniform
max_depth	[2, 20]	Uniform
num_leaves	[100, 1000]	Log
min_child_samples	[5, 100]	Log
learning_rate	[0.005, 0.5]	Log
reg_alpha	[0.0001, 0.1]	Log
reg_lambda	[0.0001, 0.1]	Log

A.2 Density Estimation

Both our systems incorporate a density estimation step. For this, we train an MLP and use its penultimate dense layer as a feature extractor to map inputs into a continuous latent space. The MLP is trained on alerts from months 4 to 6 and validated on month 7. It is optimized to minimize the weighted log-loss described in Equation 13. Hyperparameter optimization is performed using TPE (Akiba et al., 2019), running for 100 trials with 10 startup trials. The hyperparameter search space and the selected parameters are shown in Table 4.

In our density-based conformal prediction method, we apply KDE (Rosenblatt, 1956) with a bandwidth of 0.2 to the feature vectors extracted from the MLP. These density scores are subsequently used to compute prediction sets, as outlined in Section B of the Appendix.

Table 4: Feature extraction MLP: hyperparameter search space.

Hyperparameter	Values	Distribution	Selected
optimizer	Adam	–	Adam
hidden_layer_sizes	{[128/200, 20/50/128, 0/20/50]}	–	[128, 50]
learning_rate	[1e-5, 1e-3]	Log	3.096e-5
weight_decay	[1e-5, 1e-3]	Log	3.84e-4
dropout_rate	[0.0, 0.5]	Uniform	0.056
num_epochs	[20, 200]	Uniform	73
batch_size	[10, 128]	Log	30

In the density-softmax approach, we use RealNVP normalizing flows (Dinh et al., 2017) to estimate data likelihoods from the extracted features. Hyperparameter optimization for RealNVP is performed using TPE (Akiba et al., 2019) over 100 trials (10 startup trials) to minimize the negative log-likelihood on validation data. Training data spans alerts from months 4 to 6, with validation on month 7. The search space for hyperparameters is detailed in Table 5. A separate RealNVP model is then trained for each expert and data availability scenario, producing unique density scores for each expert based on their labeled subsets.

Table 5: RealNVP: hyperparameter search space.

Hyperparameter	Values	Distribution
num_coupling_layers	[2, 10]	Uniform
hidden_dim	[64, 256]	Uniform (step 32)
learning_rate	[1e-5, 1e-3]	Log
num_epochs	[50, 300]	Uniform

B Density-Based Conformal Prediction

The training and prediction algorithms for density-based conformal prediction are defined in Algorithms 1 and 2.

Algorithm 1: Training algorithm

Input: Training data $Z = (x_i, y_i), i = 1 \dots n$, Class list $\mathcal{Y}$, Confidence level α, Ratio p
Output: $\hat{p}_{\text{list}}, \hat{q}_{\text{list}}$
Initialize: $\hat{p}_{\text{list}} = \text{list}, \hat{q}_{\text{list}} = \text{list}$
for $y \in \mathcal{Y}$ **do**
  $X_y^{tr}, X_y^{cal} \leftarrow \text{SubsetData}(Z, \mathcal{Y}, p)$
  $\hat{p}_y \leftarrow \text{LearnDensityEstimator}(X_y^{tr})$
  $\hat{q}_y \leftarrow \text{Quantile}\big(\hat{p}_y(X_y^{cal}), \alpha\big)$
  $\hat{p}_{\text{list}}.\text{append}(\hat{p}_y)$
  $\hat{q}_{\text{list}}.\text{append}(\hat{q}_y)$
end for
return $\hat{p}_{list}, \hat{q}_{list}$

Algorithm 2: Prediction algorithm

Input: Input x, Trained $\hat{p}_{\text{list}}, \hat{q}_{\text{list}}$, Class list $\mathcal{Y}$
Output: $\mathcal{C}$
Initialize: $\mathcal{C} = \text{list}$
for $y \in \mathcal{Y}$ **do**
  if $\hat{p}_y(x) \geq \hat{q}_y$ **then**
    $\mathcal{C}$.append(y)
  end if
end for
return $\mathcal{C}$

C Downweighing model prediction in the conformal prediction approach

The downweighting factor $\frac{1}{w_i}$ in Equation 11 is determined by the conformal coverage level α and the proportion of null set predictions in the batch. For each new instance, we compute two density scores $\hat{p}(x_i|y)$ (one for each class in a binary setting). If neither score exceeds the quantile $\hat{q}_y$ defined during calibration, the instance is classified into the null set. These quantiles are selected to meet the desired coverage level α. Lower α values represent a more conservative approach to detecting OOD data, prioritizing accuracy and reducing null set predictions, while higher α allows for more null set predictions with a higher error rate (Figure 6). For each instance, we can find the smallest alpha $\alpha_\emptyset$ for which the instance is predicted as the empty set α by gradually increasing the coverage level α. This process requires minimal additional computation since the density scores are already available; we simply need to find the $1 - \alpha$ quantile. The smaller $\alpha_\emptyset$, the more OOD the instance is considered, and one approach is to set $w_i = \alpha_\emptyset$.

It's important to differentiate between instances classified as null sets due to the desired coverage level and those classified as null sets due to data drift. In the absence of data drift, the proportion of null set predictions should approximate α but should not exceed it. If the proportion of null set predictions does exceed α, this suggests that accuracy has fallen below $1 - \alpha$, indicating a lack of data exchangeability and confirming the presence of drift (as illustrated in Figure 6). Consequently, we should only downweigh instances classified as null sets due to data drift, rather than those classified as null sets to meet the coverage level.

To build on this reasoning, we can detect OOD data by monitoring the ratio $\rho_{\emptyset,\alpha_\emptyset}/\alpha_\emptyset$, where $\rho_{\emptyset,\alpha_\emptyset}$ represents the proportion of null set predictions at a specific coverage level $\alpha_\emptyset$. This ratio compares the actual proportion of null set predictions with the expected proportion based on $\alpha_\emptyset$. A high ratio suggests that the proportion of null set predictions is higher than what is anticipated under normal circumstances, indicating potential data drift. By calculating this ratio for each instance, we can distinguish null set predictions due to OOD data from those due to the desired coverage level $\alpha_\emptyset$.

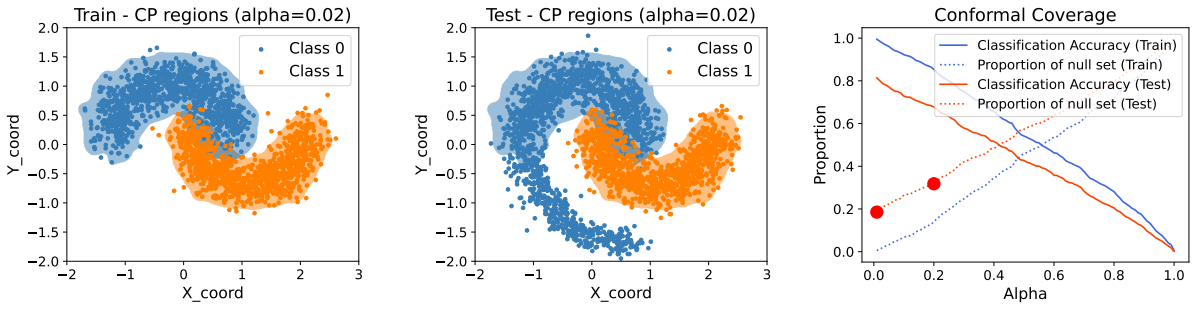

Figure 6: Example of how density-based conformal prediction is affected by data drift. At test time the distribution changes significantly, increasing the number of empty set predictions.

For example, in Figure 6, consider instances predicted as null sets at $\alpha_\emptyset = 0$. Since $\rho_{\emptyset,\alpha_\emptyset} = 0.2$, the ratio becomes $\frac{0.2}{0} = \infty$, leading to these instances being fully downweighted with $\frac{1}{\infty} = 0$. For instances with $\alpha_\emptyset = 0.2$, the ratio is $\frac{0.4}{0.2} = 2$, resulting in a downweighing factor of $\frac{1}{2}$.

One challenge here is that an instance predicted as a null set at a particular $\alpha_\emptyset$ will also remain a null set for all higher α values. This inflates the null set proportion $\rho_{\emptyset,\alpha_\emptyset}$, not specifically due to the amount of null sets at $\alpha_\emptyset$, but due to previously identified null sets at lower α levels. For instance, in the previous example, all OOD data is detected for α as low as 0.02; however, the instance that is predicted as the null set for $\alpha = 0.2$ is still downweighed by 0.5.

To address this, rather than directly comparing $\rho_{\emptyset,\alpha_\emptyset}$ with $\alpha_\emptyset$, we fit a line between the points $(\alpha_\emptyset - s, \rho_{\emptyset,\alpha_\emptyset - s})$ and $(1, 1)$, where s is the step size between successive values of α. This approach is based on the assumption that at $\alpha = 1$, we expect 100% of instances to be classified as null sets. The value of this line provides a heuristic for the expected proportion of null sets, assuming no additional OOD data is identified as α increases. Therefore, this adjustment allows determining if additional OOD data is identified at $\alpha_\emptyset$. The resulting weight w_i for an instance x_i is then calculated as:

$$w(x_i, \mathcal{X}_{\text{batch}}, \mathcal{C}_{\{\alpha_i\}_{i=1}^n}) = \frac{\rho_{\emptyset,\alpha_\emptyset}}{\rho_{\emptyset,\alpha_\emptyset - s} + \frac{1-\rho_{\emptyset,\alpha_\emptyset - s}}{1-(\alpha_\emptyset - s)} s}, \tag{16}$$

where $\rho_{\emptyset,\alpha_\emptyset} = \frac{|\mathcal{C}_{\alpha_\emptyset}(\mathcal{X}_{\text{batch}}) = \emptyset|}{|\mathcal{X}_{\text{batch}}|}$ represents the proportion of null set predictions at $\alpha_\emptyset$, based on the instance x_i and the batch $\mathcal{X}_{\text{batch}}$. The denominator provides the expected proportion of null sets, derived from fitting the line as described, where s is the step size between successive values of α. If $\alpha_\emptyset = 0$, the denominator is set to 0, ensuring that fully OOD instances are appropriately downweighted.

By applying this approach to the example in Figure 6, the first point in the conformal coverage plot (corresponding to $\alpha_\emptyset = 0$) would still be significantly downweighed, as the high proportion of null set predictions indicates a clear detection of OOD data. However, the second point, which corresponds to a higher $\alpha_\emptyset$, would not be downweighed at all, since all OOD data had already been identified at lower α levels. This behavior is exactly what we want, as it prevents the system from penalizing in-distribution data that comes after OOD data has been detected at lower $\alpha_\emptyset$.

D Optimizing the conformal coverage level

In our conformal prediction framework, an empty set prediction indicates high epistemic uncertainty, controlled by the coverage level α: a higher α leads to more empty set predictions overall. Figure 7 illustrates two complementary views of this behavior. The left plot shows, for noisy (OOD) instances only, the proportion predicted as the empty set as α increases. The right plot shows the proportion of empty set predictions corresponding to noisy instances. Note that when $\alpha = 1$, all instances are predicted as the empty set, which explains why, in the right plot, the y-axis lower bound is 0.2 (the fraction of noisy data we introduced) and why in the left plot all noisy data is eventually identified.

The key question is how easily (i.e., at low α values, where only a few empty sets are allowed overall) noisy data can be detected. In the high-noise setting, even for very low α, nearly all noisy instances are already identified as empty sets, with the right plot showing that these empty set predictions almost exclusively originate from noisy data. By contrast, in the low-noise setting, noisy data is harder to separate: the proportion of noisy instances predicted as empty sets is lower, with a substantial fraction of empty set predictions coming from in-distribution data. This is expected, as in this setting the noisy data is more similar to the original distribution. Nevertheless, even under low noise, at very small α values, roughly 60% of empty set predictions still correspond to noisy data, meaning the method identifies the most evident OOD instances. These results confirm the method's ability to identify noisy instances and defer such cases via ReL, especially in high noise scenarios.

In Table 6, we present the proportion of empty set predictions (with 95% confidence intervals) for each experimental setting. These results are averaged over five runs, each using a different random seed for noise injection.

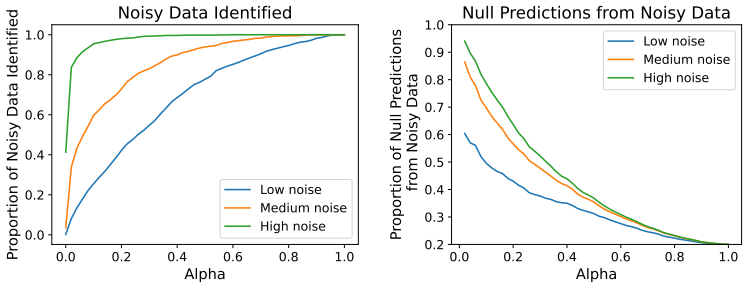

Figure 7: Conformal prediction for OOD detection. Left: proportion of noisy (OOD) instances predicted as empty sets. Right: proportion of empty set predictions originating from noisy data.

Our optimization process (Equation 11) is highly dependent to both the noise setting and the deferral rate. The noise setting controls the degree of distributional shift in the test set: higher noise levels lead to more empty set predictions at lower coverage levels. For instance, in the low-noise scenario, fewer than 4% of instances on average are predicted as the empty set; in contrast, under high noise, this proportion can rise to 20%, reflecting the fact that around 20% of the test instances are corrupted.

The deferral rate further influences the optimization process. As the deferral rate increases, the solver selects a coverage level α that typically yields a higher fraction of empty sets, ensuring that the number of cases referred to human experts remains within their workload constraints. Consistently across all settings, we observe a rising percentage of empty set predictions with increasing deferral rates.

Table 6: Conformal Coverage Optimization for the representative settings.

Setting				**Empty set predictions**	
Noise	NE	DR	DA	$\emptyset$ %	**Alpha**
low	1	20	1/1	4.29% ± 0.66	0.03 ± 0.0068
low	1	20	1/5	3.08% ± 0.67	0.02 ± 0.0054
low	1	40	1/1	8.23% ± 0.63	0.07 ± 0.0064
low	1	40	1/5	4.61% ± 0.88	0.03 ± 0.0090
low	3	20	1/1	2.57% ± 0.32	0.01 ± 0.0026
low	3	20	1/5	2.38% ± 0.42	0.01 ± 0.0024
low	3	40	1/1	4.11% ± 0.44	0.03 ± 0.0055
low	3	40	1/5	3.55% ± 0.67	0.02 ± 0.0059
low	5	20	1/1	2.58% ± 0.23	0.01 ± 0.0018
low	5	20	1/5	2.37% ± 0.45	0.01 ± 0.0032
low	5	40	1/1	3.65% ± 0.59	0.02 ± 0.0054
low	5	40	1/5	3.25% ± 0.51	0.02 ± 0.0037
medium	1	20	1/1	9.93% ± 0.42	0.03 ± 0.0034
medium	1	20	1/5	9.03% ± 0.44	0.03 ± 0.0022
medium	1	40	1/1	15.04% ± 0.58	0.08 ± 0.0067
medium	1	40	1/5	10.94% ± 1.14	0.04 ± 0.0085
medium	3	20	1/1	8.47% ± 0.13	0.02 ± 0.0013
medium	3	20	1/5	8.35% ± 0.25	0.02 ± 0.0023
medium	3	40	1/1	9.77% ± 0.43	0.03 ± 0.0015
medium	3	40	1/5	9.56% ± 0.53	0.03 ± 0.0019
medium	5	20	1/1	8.55% ± 0.06	0.02 ± 0.0014
medium	5	20	1/5	8.28% ± 0.11	0.02 ± 0.0019
medium	5	40	1/1	9.69% ± 0.51	0.03 ± 0.0022
medium	5	40	1/5	9.44% ± 0.43	0.03 ± 0.0027
high	1	20	1/1	17.03% ± 0.59	0.02 ± 0.0020
high	1	20	1/5	16.62% ± 0.95	0.02 ± 0.0023
high	1	40	1/1	20.19% ± 0.61	0.05 ± 0.0073
high	1	40	1/5	18.33% ± 0.29	0.03 ± 0.0034
high	3	20	1/1	15.12% ± 0.88	0.01 ± 0.0023
high	3	20	1/5	14.93% ± 1.00	0.01 ± 0.0025
high	3	40	1/1	17.90% ± 0.17	0.02 ± 0.0012
high	3	40	1/5	17.75% ± 0.25	0.02 ± 0.0008
high	5	20	1/1	16.25% ± 0.95	0.02 ± 0.0028
high	5	20	1/5	15.96% ± 0.88	0.02 ± 0.0022
high	5	40	1/1	17.79% ± 0.19	0.02 ± 0.0009
high	5	40	1/5	17.72% ± 0.24	0.02 ± 0.0020