Learning to Defer: A Survey

Joshua Strong [1*], Emma Sun [1], Harry Rogers [1], Helen Higham [2], and J. Alison Noble [1]

[1]Department of Engineering Science, University of Oxford, UK ,
[2]Nuffield Department of Clinical Neurosciences, University of Oxford, UK
[*]Correspondence: joshua.strong@eng.ox.ac.uk

Preprint — May 8, 2026

Abstract

Learning to defer (L2D) enables AI systems to choose between autonomous prediction and deferral to experts. This survey consolidates the fast-growing literature through a four-branch taxonomy: methodological frameworks; optimization and theory; task generalizations; and real-world adaptations. We outline contrasts between score-based and predictor–rejector formulations; one-stage, two-stage, and post-hoc training; unify surrogate losses with theoretical guarantees; and synthesize extensions to regression, multi-task prediction, top-k committees, sequential settings, and causal pipelines. Practical considerations include limited annotations, dynamic expert pools, workload/budget control, fairness, interpretability, robustness, and uncertainty handling, concluding with open challenges for reliable human–AI decision systems.

Keywords: learning to defer, selective prediction, rejection learning, human–AI collaboration, uncertainty, decision referral, human-in-the-loop

Contents

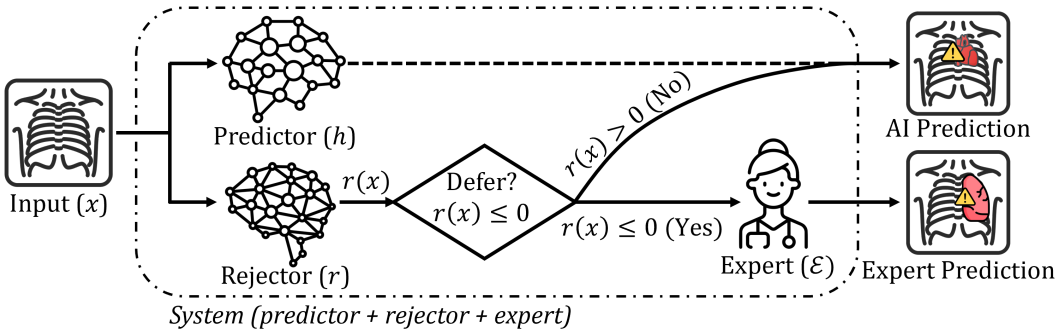


Figure 1: L2D (inference) framework. An input x is passed through a **rejector** and **predictor**. Depending on the output of the rejector, the system decides to either defer to the expert, or to autonomously predict using the predictor's output. The Figure depicts the predictor-rejector formulation, as described in §3.2. Notation in parentheses is introduced in §3.1.

1 Introduction

Artificial intelligence (AI) is increasingly being deployed in high-stakes environments where the cost of an error is significant: an error in a healthcare setting could lead to incorrect treatment and patient harm, an error in finance could mean failing to stop a major fraudulent act, and a failure in an autonomous car could cause a catastrophic accident. To mitigate these risks, Learning to Defer (L2D) is emerging as a prominent area of research within Human-AI Collaboration (HAIC). L2D equips an AI model with the ability to either make an autonomous prediction or to defer its decision to an external human expert. In this paper, we use "expert" to mean a skilled human; experts vary in their decisions and may be biased or fallible. This deferral acts as a fail-safe, redirecting challenging or uncertain cases to an expert who can provide more reliable or nuanced judgment. Deferral can provide additional benefits beyond improved task performance, such as shared accountability in decision-making, increased trust [66] and hence adoption, and a more efficient and cost-effective use of human time and expertise. Overall, the central objective of L2D is to optimize the overall system performance of this human-AI collaboration, by finding the optimal decision between deferring or autonomously predicting for each case.

The L2D problem can be explained as a system with three principal components: a task **predictor** model, a **rejector** model, and a **human expert** (Figure 1). In this framework, for any given input, the predictor generates a potential prediction. The rejector model then decides whether to accept this prediction or to defer to the expert. This decision is governed by a cost structure, which typically includes a penalty for incorrect predictions and a specified cost for querying the expert. The fundamental task in L2D is therefore to learn prediction and deferral policies that jointly minimize the total expected cost of this hybrid Human-AI system.

1.1 Contributions and Organization of This Survey

While L2D is a rapidly expanding field, there is a distinct lack of a comprehensive survey. This survey fills that gap. The most relevant existing work is the 2022 perspective paper by Leitão et al. [36]. This work provided a good introduction of the field in its infancy and identified several critical open challenges, such as data requirements, expert capacity management, and model brittleness. However, the L2D field has advanced considerably since this work. Since the original paper was based on seven L2D publications, there have been over forty new ones that have not only addressed many of the initial challenges but have also introduced new research

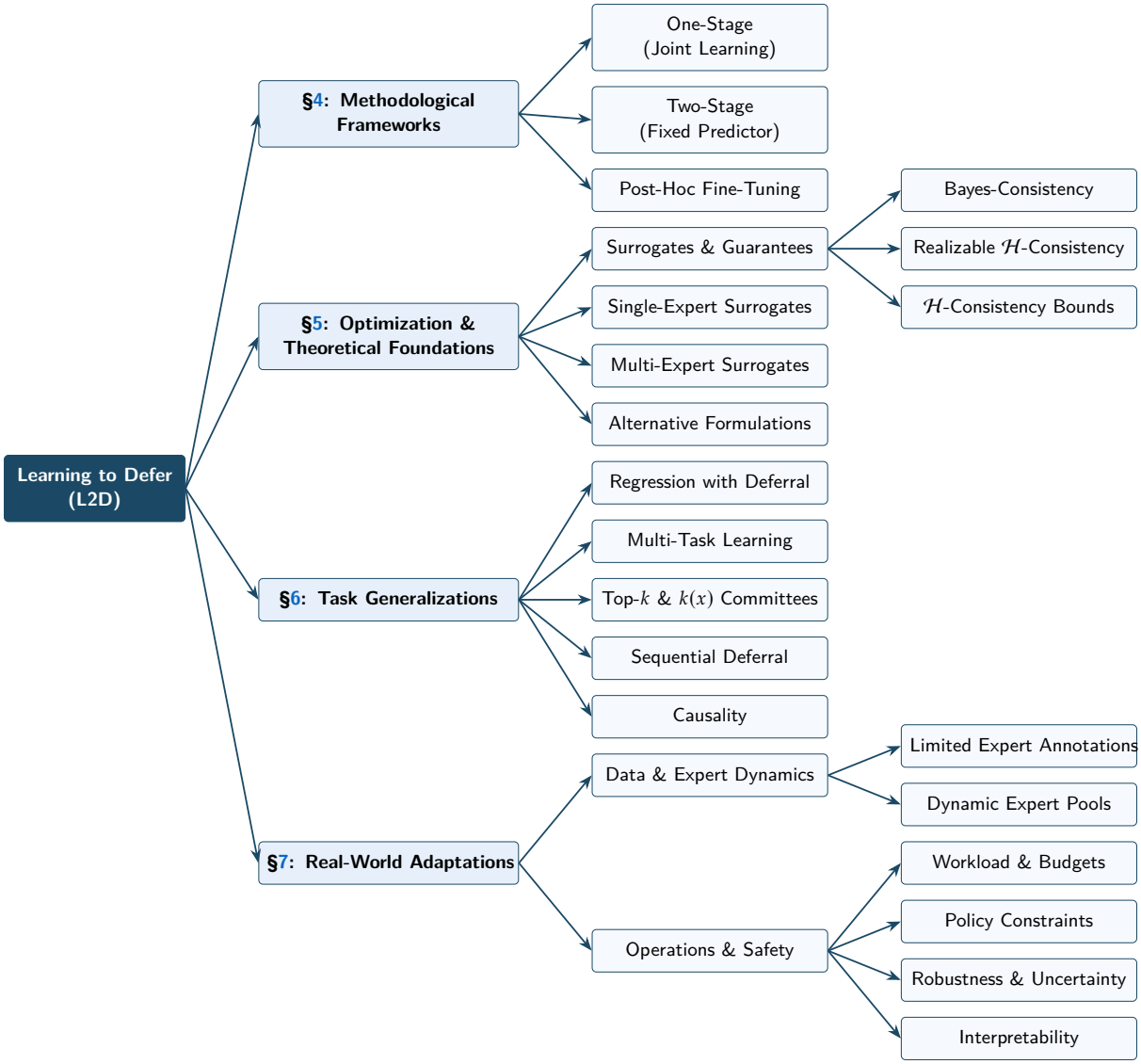


Figure 2: A taxonomy of learning to defer (L2D), with the literature organized into methodological frameworks, optimization and theory, real-world adaptations, and task generalizations.

directions. These include significant advancements in the theoretical understanding of surrogate losses [47, 49, 56], the introduction of new methodological frameworks (e.g., two-stage learning) [43, 50, 52], the generalization of L2D to complex tasks (e.g., regression, multi-task learning) [48, 53], and sophisticated frameworks for handling real-world constraints (e.g., dynamic expert pools, fairness) [64, 67, 68].

The rapid development of these new theories and methods has led to a fragmented body of literature, making it difficult for researchers to gain a holistic view of the field. This survey provides a structured, comprehensive analysis and the first categorization of the L2D field, addressing this gap. The primary contributions of this survey are:

- **A four-branch taxonomy of the L2D field:** We introduce a taxonomy that organizes L2D research into four core themes: *Methodological Frameworks*, *Optimization & Theoretical Foundations*, *Task Generalizations*, and *Real-World Adaptations* (Figure 2).
- **Systematic review of methodological frameworks:** We contrast one-stage (joint), two-stage (fixed predictor), and post-hoc fine-tuning frameworks, and analyze their trade-offs in

performance, practicality, and robustness (§4).

- **Comprehensive synthesis of optimization and theory:** We trace the development of surrogate objectives across one-stage and two-stage settings and clarify their theoretical guarantees, alongside calibration and cost-sensitivity issues; we also discuss the alternative formulation of dependent Bayes optimality (§5, §5.4).
- **Task generalizations:** We cover extensions beyond the canonical classification setting to regression, multi-task learning, and top-k deferral (§6).
- **Analysis of practical adaptations:** We synthesize approaches for limited expert data, dynamic expert pools, workload/budget control, and responsible AI considerations such as fairness, interpretability, adversarial robustness, and uncertainty-aware abstention (§7).
- **Delineation of open challenges and a research agenda:** We identify gaps and outline promising directions for future work (§8).

2 Related Fields

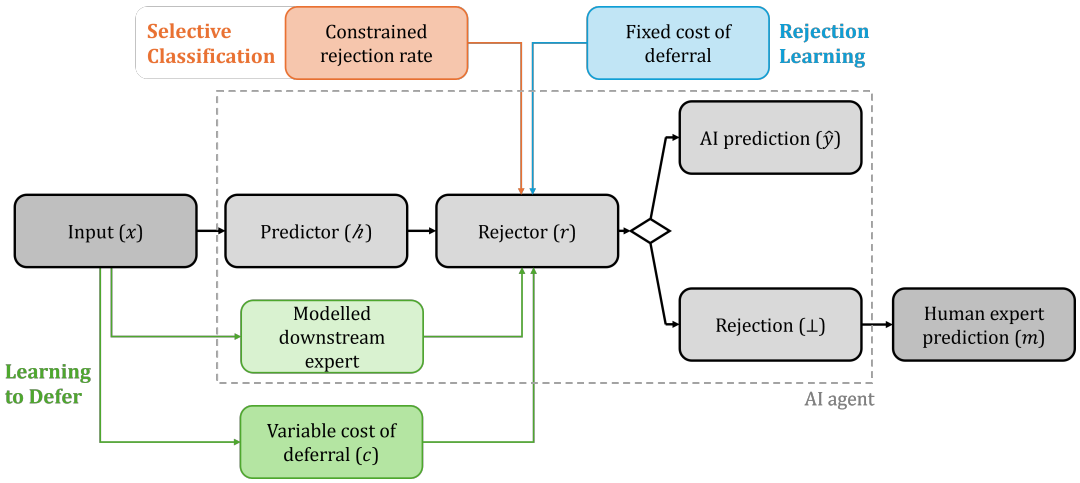


Figure 3: Differences between rejection learning, selective classification and L2D. Notation in parentheses is introduced in §3.1.

2.1 Direct Predecessors of Learning to Defer

The L2D field builds upon a well-established history of research into models that have the ability to abstain from making a prediction. This family of methods acknowledges that it can be beneficial for a model to recognize its own uncertainty and reject an instance rather than risk an erroneous low-confidence prediction. The primary differentiator for L2D lies in its reconceptualization of what happens *after* a model abstains and, crucially, how that action is valued within the system. We discuss two major fields in this domain: **Rejection Learning** and **Selective Classification**.

Rejection Learning

Learning with a reject option, or rejection learning, is the most direct ancestor of L2D. First formalized by Chow [13, 14] and further developed by Cortes et al. [16], this concept introduces an additional option for a classifier, alongside the standard class predictions: the reject option.

The AI model can choose to abstain from making a prediction if its confidence in its own output is below a certain threshold. This rejection is generally prompted by one of two conditions: *ambiguity rejection*, for instances that are confusing or near a decision boundary, or *novelty rejection*, for instances that are far away from the training data [29, 75]. The key assumption in the standard rejection learning framework is that the act of rejection incurs a predefined, constant cost. The model is therefore trained to optimize a trade-off between the classification error on the instances it accepts and the fixed penalty it pays for the instances it rejects. The goal is to improve the model performance on the non-rejected subset of data. A key limitation of this method is that it is inherently non-adaptive to any downstream human expert's performance, as it only considers the model uncertainty, in isolation from external factors. This leaves it open to a significant limitation: *what if the expert performs poorly on deferred cases, or outperforms AI on those retained by the model?*

L2D can be viewed as an adaptive generalization of the learning with rejection framework. This generalization is achieved by replacing the fixed cost with a variable, instance-dependent cost, which can represent the performance of the downstream expert on that specific instance (Figure 3). The seminal work from Madras et al. [40] introduces **adaptive rejection learning**, termed **learning to defer**, where the decision to defer is based not only on the model confidence but also on the expected expert competence. The model must learn not only about its own weaknesses but also the specific strengths and weaknesses of the expert it is collaborating with, in relation to itself. This original framework conceptualizes the system as a mixture of Bernoullis such that it uses a learned switch that, for each case, chooses whether the final decision comes from the model or from the human expert. This arrangement functions similarly to a mixture-of-experts (MoE) model [30], but with the distinction that the learning algorithm refines only the automated model and its deferral strategy, while the characteristics of the external expert are considered fixed. Conceptually, L2D serves as a sophisticated form of ambiguity rejection, as it directly learns which uncertain instances are best handled by the model versus the expert.

Selective Classification

A closely related approach is selective classification [20]. As with rejection learning, it allows a model to abstain from predicting. However, its objective is different; instead of rejection carrying a cost, the aim is to maximize accuracy without violating a predefined constraint on the rejection rate or coverage. For example, the goal might be to maximize accuracy while ensuring the model predicts on at least 80% of the data. The focus is on controlling the volume of abstention to guarantee a certain level of performance on the selected examples [23, 24].

In comparison, the optimal deferral rate in L2D is not predefined but emerges from system-level cost optimization (explained further in §3.3). The model defers when it is economically rational to do so for the overall system. While this principle is central to L2D, a key insight from the field's evolution is that this unconstrained optimization can lead to an impractically high deferral rate, overwhelming the human expert [1]. Recognizing this, more recent L2D frameworks introduce explicit mechanisms for workload balancing and budget control, a topic we explore further in Section 7.5. Furthermore, L2D explicitly models the downstream expert, whose varying performance is central to the problem, a component absent from the standard selective classification setup.

2.2 Alternative Paradigms of Human-AI Collaboration

Beyond methods that simply abstain, L2D can be further contextualized by comparison to other established frameworks for HAIC such as Human-in-the-loop (HITL). While these fields also involve collaboration between automated systems and humans, they differ fundamentally in whether the human or AI has decision-making authority, the workflow and the purpose of the interaction.

Human-in-the-Loop Machine Learning

L2D can be situated within the broader field of HITL, which encompasses various techniques where humans and models interact. These interactions are often categorized by who, or what, is in control of the process: the model, the human expert, or a shared partnership [55]. L2D presents a specific configuration of this relationship, which becomes clear when contrasted with other common HITL approaches such as *Active Learning* and *AI-assisted decision-making*.

In *Active Learning*, the model is in control of the decisions made. Active learning is a training procedure designed to minimize the cost of data labeling by having a model select the most informative examples for a human to label [37]. In this setup, the model queries a human for ground truth labels on instances where it is most uncertain. The goal is to achieve higher accuracy with fewer labeled examples compared to random sampling. The human's input is used to improve the model itself during the training phase. The fundamental difference from L2D lies in the phase and purpose of the human interaction. Active learning is a *training-time* procedure where the human's role is to provide ground truth to improve the model. In contrast, L2D is an *inference-time* procedure. The L2D model defers to the expert during deployment, and the expert's decision becomes the final output for that instance.

Another common HITL configuration involves an AI system that provides an initial prediction or recommendation, which a human expert then reviews. The expert holds the final authority and can choose to accept, reject, or modify AI output. This model is prevalent in current medical diagnosis, where an AI might flag potential issues on a medical scan for a radiologist to confirm. The human is responsible for oversight and final accountability [5, 6]. Mozannar and Sontag [57] explicitly refer to L2D as the "reverse setting" compared to existing AI-assisted decision-making, where the human expert has the final say and acts as a check on the model. In L2D, the model has the initial agency to decide whether to handle a case itself or to defer it to the human. The decision flow is initiated by the model's self-assessment of its capabilities relative to the expert, rather than the expert assessment of the model output.

2.3 Related Algorithmic Frameworks and Architectures

The implementation of L2D relies on adapting existing algorithmic frameworks to its unique problem setting. By examining these underlying structures, we can better understand both the mechanics of L2D and the novel constraints it imposes on these familiar architectures.

Cascading Models

Cascading classifiers are multi-stage architectures designed to improve computational efficiency and, in some cases, accuracy. A typical cascade consists of a sequence of classifiers, starting with a simple, fast model and progressing to more complex, computationally expensive ones (Figure 4 (i)). The initial models filter out "easy" or obvious negative examples, allowing the more powerful models to focus only on the more difficult or ambiguous instances that remain [18].

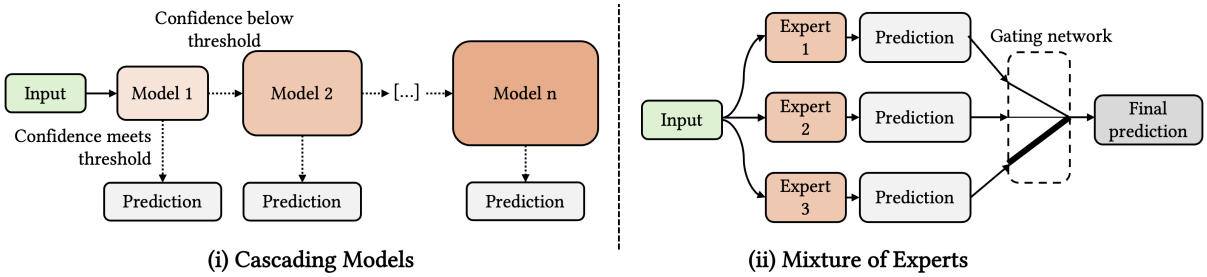


Figure 4: (i) Cascading Models framework, where increasingly complex models output a prediction if a confidence threshold is met, and otherwise pass the problem to the next model in the cascade, and (ii) Mixture-of-Experts framework, where a gating function (analogous to the rejector) combines the output of multiple expert predictions on the same input.

This structure is conceptually similar to L2D, where the classifier acts as the first stage and the expert as the second.

While structurally analogous, the relationship has been recently formalized, with research demonstrating that classical model cascades arise as a restricted special case of generalized L2D frameworks [52]. The key distinctions lie in flexibility and the scope of the objective. Cascading models are typically designed with a fixed sequence of models to optimize for computational efficiency, stopping at the first model that meets a confidence threshold. L2D, by contrast, learns a dynamic, input-dependent policy to route queries to the most cost-effective agent (or even a committee of agents), optimizing for overall system performance, which can include metrics such as accuracy and fairness [40]. Furthermore, the final stage in a cascade is another trainable model, whereas in L2D, it is often a human expert, whose performance characteristics are taken to be inherent, unchangeable, and often non-uniform.

Mixture-of-Experts (MoE)

The Mixture-of-Experts (MoE) model [31] is a commonly utilized ensemble technique. It consists of several expert networks, each specializing in a different part of the input space, and a gating network that learns to assign weights to each expert's prediction for a given input. The final output is a weighted combination of the expert outputs (Figure 4 (ii)). The single-expert L2D framework can be viewed as a specialized form of a two-expert MoE system, where the gating function corresponds to the rejector, and the two "experts" are the AI classifier and the human.

The important distinction is that in L2D, one of the experts (the human) is a fixed, external, and non-differentiable component. Unlike in a standard MoE, where all experts and the gating network are trained jointly via backpropagation, the L2D model has no control over the human expert's parameters or behavior. The optimization is constrained: the system can only learn the parameters of one expert (the classifier) and the gating function (the rejector) to best complement the fixed, pre-existing human expert. This makes the problem fundamentally different from standard end-to-end MoE training. Mozannar and Sontag [57] further critique the direct application of a MoE loss from Madras et al. [40], proving it is not classification-consistent and can lead to suboptimal behavior where the model learns never to defer.

3 The Learning to Defer Problem Formulation and Introduction

This section gives a compact formalization of Learning to Defer (L2D) and orients the reader to the four-part taxonomy developed in this survey (Figure 2). At a high level (Figure 1), an L2D system decides, per input, whether to *act* with an automated predictor or *defer* to an external

expert (human or stronger model) so as to minimize a system-wide cost.

3.1 Notation and Setup

Table 1: Summary of mathematical notation for Learning to Defer (L2D) in this survey paper.

Symbol	Meaning	Symbol	Meaning
$\mathcal{X}$	Input space	$\mathcal{Y}$	Output label space
(X, Y)	Input/label random variables	(x, y)	A sampled data point from $\mathcal{D}$
$\mathcal{D}$	Joint distribution over $\mathcal{X} \times \mathcal{Y}$	$c(x, y)$	Cost of deferral
$\perp$	Deferral action	$\mathcal{Y}_\perp$	Extended label space
h	Predictor function	$\mathcal{H}$	Predictor hypothesis class
r	Rejector function	$\mathcal{R}$	Rejector hypothesis class
s	Scoring function	$\mathcal{E}$	Expert function
m	Expert prediction	$\mathcal{M}$	Expert prediction space
K	Number of classes	J	Number of experts

The Learning to Defer (L2D) framework extends standard supervised learning by enabling a model to either autonomously predict or defer the decision to an expert. This expert could be a human, or a more powerful AI model. This framework is built upon three core components, which we collectively refer to as the *system*: an autonomous **predictor** (h), a fixed **expert** ($\mathcal{E}$), and a **deferral mechanism** (r or s, cf. §3.2) that routes inputs to either the predictor or the expert to minimize a system-wide loss. Here, we introduce the L2D problem formulation for the standard task of multi-class classification.

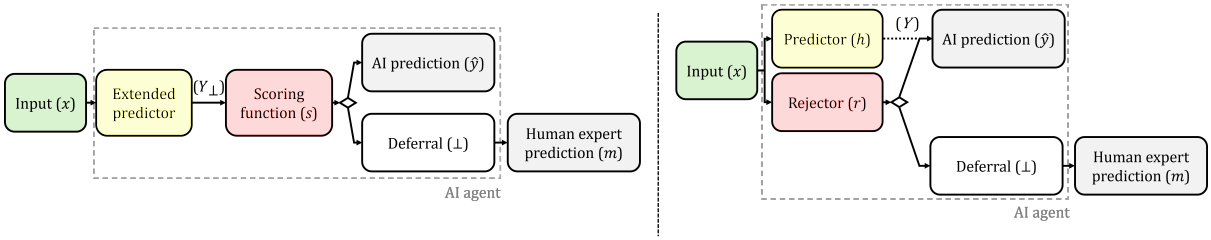
Let $\mathcal{X}$ be the input space and $\mathcal{Y} = \{1, \ldots, K\}$ be the output label space. We assume there is a joint probability distribution $\mathcal{D}$ over the space $\mathcal{X} \times \mathcal{Y}$. We denote the random variables for the input and true label as (X, Y), and a single data point (x, y) is a realization sampled from $\mathcal{D}$. The expert $\mathcal{E}$ is represented by a fixed, deterministic function $\mathcal{E} : \mathcal{X} \to \mathcal{M}$, where $\mathcal{M}$ is the expert prediction space (typically $\mathcal{M} \equiv \mathcal{Y}$). For any input x, the expert's prediction is $m = \mathcal{E}(x)$. For training, we assume access to triplets (x, y, m), where m is generated by the expert. The core mathematical notation is summarized in Table 1.

3.2 Architectural Formulations (Score-Based vs. Predictor-Rejector)

A central design choice in L2D lies in the architecture of the deferral mechanism: a unified, **score-based** approach that integrates prediction and deferral into a single model, and a modular, **predictor-rejector** approach that decouples them into separate models. Understanding the trade-offs between these two formulations is useful not only for deeply understanding L2D, but also for appreciating *why* frameworks are designed in particular ways and for developing new tasks. We further detail these formulations and their optimization implications in §4: *Methodological Frameworks in L2D*.

The Score-Based Formulation

The Score-Based (SB) formulation (Figure 5 (i)), introduced in [57], frames the deferral problem as a single, unified classification task. This is achieved by extending the label space to $\mathcal{Y}_\perp = \mathcal{Y} \cup \{\perp\}$, where the new label $\perp$ represents the action of deferring to the expert. A single, multi-output **scoring function**, $s : \mathcal{X} \times \mathcal{Y}_\perp \to \mathbb{R}$, is then learned to assign a score to each possible class and to the deferral action. The system's decision is given by a single argmax:



(i) Score-Based Formulation **(ii) Predictor-Rejector Formulation**

Figure 5: (i) Score-based formulation, with prediction and deferral coupled, and (ii) Predictor–rejector formulation, with prediction and deferral performed by distinct functions. Figure 1 features the predictor–rejector formulation.

action$(x) = \text{argmax}_{a \in \mathcal{Y}_\perp} s(x, a)$. In this unified architecture, the deferral decision is implicitly coupled with the class predictions via a competition among all $K+1$ outcomes.

The Predictor–Rejector Formulation

In contrast, the Predictor–Rejector (PR) formulation (Figure 5 (ii)) treats prediction and deferral as two distinct tasks governed by separate functions, following early work on classification with a reject option [17]. It comprises two components: a **predictor** $h : \mathcal{X} \to \mathcal{Y}$ from a hypothesis class $\mathcal{H}$, and a **rejector** $r : \mathcal{X} \to \mathbb{R}$ from a hypothesis class $\mathcal{R}$. For a given input x, the system defers to the expert if $r(x) \leq 0$; otherwise, it predicts $h(x)$.

Score-Based vs. Predictor–Rejector: Architectural Trade-Offs

While both formulations aim to recover the same ideal deferral strategy under an appropriate risk, their architectural differences yield distinct practical implications.

Treating deferral as an additional class, the score-based formulation utilizes the existing theory of multiclass models and naturally supports *one-stage (joint) training* in which prediction and deferral are learned simultaneously (cf. §4). A potential limitation is that the deferral mechanism is tied to the predictor capacity: achieving optimal deferral may necessitate a higher-capacity joint model even when the optimal predictor and rejector are themselves simple [42]. Moreover, this construction is specific to classification and does not extend naturally to regression due to the continuous output space.

The predictor-rejector formulation explicitly separates the rejector from the predictor, allowing the two to come from different hypothesis classes (e.g., a deep predictor h paired with a lightweight r). This makes the predictor-rejector formulation especially well-suited to a *two-stage training setup* in which r is learned around a fixed, pre-trained predictor (cf. §4.2). Practically, this can reduce computational cost when the predictor is large, and it enables broader applicability: the same rejector principle can be used beyond classification, including regression [48] and heterogeneous or cross-modal systems [53]–for example, an image classifier that defers to a human expert who returns a textual report.

3.3 The Deferral Target Loss and Optimization Objective

Regardless of the formulation, the goal is the same: to learn a deferral mechanism that minimizes a unified measure of overall cost-constrained system performance. This is formalized by the **target deferral loss**, which accounts for both autonomous prediction and expert deferral. This subsection describes the intuition behind the optimization objectives that have been proposed.

The Optimization Objective When the system makes an autonomous prediction, it incurs a standard 0-1 misclassification loss. When it defers, it incurs a case-specific **cost of deferral**, denoted by the function $c(x, y)$. This cost is an important modeling choice; a common example is the expert's own misclassification error, represented by an indicator function, $c(x, y) = \mathbb{1}_{\mathcal{E}(x)\neq y}$, but it can also be a more complex penalty. The learning objective is to find the strategy that minimizes the system expected error.

Target Loss for the Score-Based Formulation.

In the score-based formulation, the system learns a scoring function $s(x, a)$ over $\mathcal{Y}_\perp$, and the system action is $a^*(x) = \text{argmax}_{a\in\mathcal{Y}_\perp} s(x, a)$. The target loss ℓ_{SB} is defined over this single function [42]:

$$\ell_{\text{SB}}(s, x, y) = \mathbb{1}_{a^*(x)\neq y} \cdot \mathbb{1}_{a^*(x)\neq\perp} + c(x, y) \cdot \mathbb{1}_{a^*(x)=\perp} \tag{1}$$

Here, the system incurs prediction loss if the action $a^*(x)$ is a class label; it incurs the deferral cost if the action is deferral ($\perp$).

Target Loss for the Predictor-Rejector Formulation.

In the predictor-rejector formulation, the system learns a pair of functions (h, r), and the target loss ℓ_{PR} for a single instance (x, y) is explicitly defined over this pair [42]:

$$\ell_{\text{PR}}(h, r, x, y) = \mathbb{1}_{h(x)\neq y} \cdot \mathbb{1}_{r(x)>0} + c(x, y) \cdot \mathbb{1}_{r(x)\leq 0} \tag{2}$$

Here, $r(x)$ acts as a hard switch: the system incurs the predictor loss if $r(x) > 0$; the deferral cost if $r(x) \leq 0$.

3.4 The Bayes-Optimal Strategy and the Need for Surrogates

The ideal learned decision boundary, known as the **Bayes-optimal** strategy, represents the theoretical limit of performance, independent of the chosen architecture. This strategy compares the expected cost of deferral with the Bayes risk (the lowest possible error rate achievable by any classifier). It dictates that one should defer if and only if the expected deferral cost is lower than or equal to the Bayes risk [13, 57]:

$$\text{Defer } x \iff \mathbb{E}[c(x, Y)|X = x] \leq 1 - \max_{y'\in\mathcal{Y}} \mathbb{P}(Y = y'|X = x) \tag{3}$$

This rule defines the canonical target for an idealized L2D system. We permit the expert to use information unavailable to the classifier and therefore to outperform the Bayes classifier; the rule is used primarily as a benchmark for analyzing surrogate consistency. Beyond Consistency, more recent and stronger guarantees are described in §5.1.

The target losses (Eqs. 1-2) are non-convex and non-differentiable due to the 0-1 indicator functions. Direct optimization is thus computationally intractable, presenting a central challenge in L2D: designing tractable **surrogate losses** that provably converge to an optimal strategy.

Consistent Surrogate Losses

A surrogate loss is the smooth, tractable objective we minimize during training in place of the target deferral losses in Eqs. 1–2, which involve non-differentiable 0–1 decisions. A surrogate is *consistent* for L2D if, with enough data and a sufficiently rich model class, minimizing its average value leads to the same decisions as the Bayes-optimal deferral rule in Eq. 3; that is, the

learned system makes the correct *routing* choice (predict vs. defer) and, when predicting, the correct *label*. Practically, a consistent surrogate aligns the routing trade-off with the expert's instance-dependent cost (higher deferral cost shrinks the defer region and vice versa), avoids perverse incentives like always or never deferring unless those are truly optimal, and reduces to a classification-consistent surrogate when deferral is disabled [7]. Consistency ensures that success on the surrogate translates into correct L2D behavior.

The specific design of these surrogate losses depends heavily on the chosen training framework – specifically, whether the predictor and deferral mechanism are learned jointly or sequentially. While we explore these frameworks in detail in §4, we first introduce here the standard forms of consistent surrogates typically employed when all components (predictor and rejector) are optimized **jointly** (the one-stage approach). The historical development of L2D surrogate losses is detailed in §5.

A Consistent Score-Based Surrogate. A consistent surrogate for the SB formulation L_{SB} can be built from a standard multi-class surrogate ℓ (e.g., cross-entropy) by applying it to the augmented label space $\mathcal{Y}_\perp$ and re-weighting the deferral option [42]:

$$L_{SB}(s, x, y) = \ell(s, x, y) + (1 - c(x, y))\ell(s, x, \perp) \tag{4}$$

This general form, when instantiated with the cross-entropy loss, recovers the seminal surrogate from Mozannar and Sontag [57] for the special case where the cost of deferral is the expert's misclassification $c(x, y) = \mathbb{1}_{\mathcal{E}(x)\neq y}$.

Consistent Predictor-Rejector Surrogates. Designing consistent surrogates for the PR setting involves creating a loss that smoothly approximates the hard-switching behavior of the target loss (Eq. 2). An intuitive approach is to use "soft-gating" functions (like the sigmoid) to create a convex interpolation between the predictor loss and the deferral cost. While this approach is consistent for L2D in regression [48] (cf. §6.1), its consistency in the multi-class classification setting is generally not guaranteed [42]. The standard structure for PR surrogates in classification, extending the approach from the binary setting [16], typically replaces the hard indicators in the target loss with a margin-based surrogate Φ (e.g., exponential loss $\Phi(z) = e^{-z}$ or logistic loss):

$$L_{PR}(h, r, x, y) = \ell_{\text{pred}}(h, x, y) \cdot \Phi(-r(x)) + c(x, y) \cdot \Phi(r(x)) \tag{5}$$

Here, Φ acts as a modulator: when $r(x) > 0$ (accept), $\Phi(-r(x))$ is large, emphasizing the predictor loss ℓ_{pred}; when $r(x) < 0$ (defer), $\Phi(r(x))$ is large, emphasizing the cost $c(x, y)$.

Ensuring the consistency of this surrogate in the multi-class setting was noted as an open problem by Ni et al. [60]. Recent work (specifically regarding the **joint** training of h and r) has shown that the consistency of PR surrogates depends crucially on the training framework and the specific choice of ℓ_{pred}. When training h and r jointly using the surrogate in Eq. 5, consistency is only guaranteed if the predictor surrogate ℓ_{pred} satisfies stringent conditions relating its infimum to the Bayes risk [45]. This condition is met by specific, often non-convex losses, such as the Mean Absolute Error (MAE) loss or the ρ-Margin loss, but notably *not* by the standard cross-entropy loss [45].

The discussion above assumes an idealized setting (e.g., fixed experts, ample annotations, unconstrained routing). We synthesize deployment-oriented adaptations—handling sparse/noisy expert labels, dynamic expert pools, workload/budget and fairness constraints, and safety via robustness, interpretability, and uncertainty in §7: *Real-World Adaptations for L2D*.

4 Methodological Frameworks in L2D

In the early development of L2D, the standard approach was to train the entire system jointly from scratch [40, 57]. While theoretically elegant, this approach faces significant practical challenges in modern machine learning frameworks. The prevalence of large, pre-trained foundation models, which are often prohibitively expensive to retrain or accessible only via black-box APIs, necessitates more flexible frameworks. These practical constraints have led to the development of distinct frameworks for training L2D systems. These are primarily distinguished by two factors: (1) whether the predictor and deferral mechanism are trained simultaneously or sequentially, and (2) the degree of tunability assumed for the predictor model. We classify these frameworks into three major frameworks (Figure 6):

(i) **One-Stage (Joint) Learning:** The predictor and deferral mechanism are trained simultaneously from a blank slate to optimize the joint system objective.

(ii) **Two-Stage Learning:** Assumes the pre-trained predictor is fixed or "frozen" (e.g., a black-box API). A separate deferral mechanism is learned sequentially in a second stage, without altering the predictor.

(iii) **Post-Hoc Fine-Tuning:** Starts with a pre-trained predictor, but assumes its weights are tunable. The predictor and a new deferral mechanism are then jointly fine-tuned, often iteratively, to co-adapt to the deferral task.

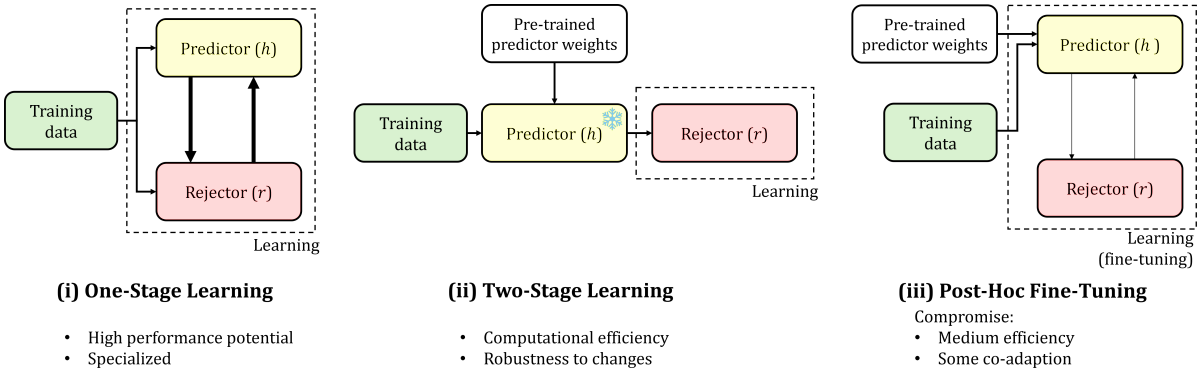


Figure 6: (i) One-stage learning, (ii) two-stage learning, and (iii) post-hoc fine-tuning frameworks, depicted for the predictor-rejector formulation (§3.2) for simplicity.

Notation. We follow §3.1 (Table 1). The predictor is h, the deferral mechanism is either a score s (score-based, SB) or a rejector r (predictor–rejector, PR); expert(s) are $\mathcal{E}$ or $\{\mathcal{E}_j\}_{j=1}^{J}$ with cost $c(x, y)$. In two-stage sections h is *fixed*; in one-stage and post-hoc sections h and the deferral mechanism are optimized jointly or co-adapted.

4.1 One-Stage (Joint) Learning

The one-stage framework represents the theoretically ideal approach to the L2D problem, introduced in the seminal works of Madras et al. [40] and Mozannar and Sontag [57]. In this framework, the classifier and the deferral mechanism are trained simultaneously by minimizing a single surrogate loss (detailed in §5).

Co-adaptation and Specialization. The defining characteristic of one-stage training is *co-adaptation*. The system does not merely learn a general-purpose classifier and then decide when

to use it. Instead, the optimization dynamics allow the predictor to *specialize*: the predictor can focus its learning capacity on the region of the input space where it is expected to make autonomous predictions, effectively ignoring the regions that the deferral mechanism learns are better handled by the expert [57]. Simultaneously, the deferral mechanism adapts not just to the expert's strengths, but also to the specialized capabilities of the evolving predictor. This end-to-end optimization aims for a globally optimal solution for the entire decision system.

Architectural Synergy and Challenges. This framework can be implemented using either the SB or PR formulation (§3.2), though they present different challenges:

- The **SB formulation** is inherently suited for joint training of the one-stage framework, as the predictor and deferral scores emerge from a single classifier model where deferral is treated as an additional class. This is the most common approach in the literature [8, 38, 51, 57, 68–70].
- The **PR formulation** can also be trained jointly, typically by optimizing a modulated surrogate loss (e.g., Eq. 5). However, ensuring the consistency of this joint optimization has proven theoretically challenging. Consistency in the joint PR setting is only guaranteed if the predictor loss function satisfies stringent conditions not met by standard losses like cross-entropy [45]. This complexity makes the SB formulation generally preferred for one-stage learning.

Practical Limitations. Despite its potential for achieving superior performance, the one-stage approach has drawbacks. One limitation is the requirement to train the model from scratch, which is computationally expensive and can be impractical, especially for large-scale models. Furthermore, the aforementioned specialization that drives greater system performance can also lead to several structural drawbacks:

- **Loss of Generalization**: AI trades generalization for specialization. Consequently, the resulting classifier becomes weaker in the regions of the input space that are likely to be deferred to the expert [1, 36].
- **Brittleness to Change**: This specialization makes the system brittle and highly susceptible to post-deployment changes [36, 68]. If the expert's availability temporarily declines, or if the system is forced to predict instances originally intended for deferral, AI performance may decline.
- **Mandatory Retraining upon Expert Change**: If the expert's characteristics or identity changes (e.g., a new expert joins the team), the entire system will break [68], necessitating the costly full retraining of the system from scratch.
- **Incompatibility with Advisory Roles**: Because the classifier specializes away from difficult/deferred instances, it is deemed unsuitable in domains where AI advises the human decision-makers (HAIC settings). In such cases, the specialized classifier cannot provide a meaningful score, tentative prediction, or explanation for the deferred cases, which are often required by the human expert [36].

4.2 The Two-Stage Setup: Deferral for Fixed Predictors

The two-stage framework, formalized by Mao et al. [43], addresses the practical limitations of the one-stage approach by decoupling the predictor training from the deferral learning. In this setup, the AI predictor is assumed to be pre-trained (e.g., an off-the-shelf classifier or a large foundation model) and is frozen during the L2D optimization. The core task is to learn a separate deferral function, a second stage that optimally routes queries between the fixed

AI and the available experts. This "plug-and-play" approach is the most practical framework for modern machine learning systems. It allows users to leverage existing, high-performing models without the prohibitive cost of retraining. It is the only viable approach when the predictor is a black box or accessible only via an API, as it does not require access to the model internal weights. This framework inherently favors the Predictor-Rejector architecture, where a lightweight deferral model is trained to "tack onto" the fixed predictor.

Unlike the one-stage framework, there is no co-adaptation. The predictor is a general-purpose model trained on the full data distribution. Consequently, two-stage systems are generally more robust to a change of expert at test time or deferral constraints compared to one-stage models at a trade-off for potentially greater system performance from specialization.

Implementations.

This framework has led to various practical implementations. The **LECODU** system [76], for instance, learns a deferral policy for a pre-trained predictor while also handling noisy, multi-rater expert data. Similarly, **DeCCaF** [1] uses a two-stage pipeline for cost- and capacity-aware deferral: it first trains a classifier and a human-expertise model separately to estimate per-instance correctness; it then uses constraint programming to assign cases to the model or experts so as to get the most correct decisions on average, given error/deferral costs and per-expert capacity limits. In extractive question-answering, **Optimal Query Allocation** [54] is a two-stage, multi-expert, cost-aware router (frozen predictor + tiny rejector) that enforces single-agent span selection. Additionally, there exist extensions to multi-task learning [53], adversarial robustness [50], top-k deferral [52], and causal discovery [15].

4.3 Post-Hoc Fine-Tuning and Calibration

The post-hoc fine-tuning framework offers a middle ground between the one-stage and two-stage approaches. It begins with a pre-trained model, leveraging the computational investment already made, but assumes "white-box" access, allowing the model weights to be updated. In this framework, the pre-trained predictor and a newly initialized deferral policy are iteratively refined in tandem. A key example is the **Differentiable Triage** framework of Okati et al. [61]. It proposes an iterative, co-adaptive process: the deferral policy is updated to identify the current model's weaknesses, and the model weights are then fine-tuned only on the instances it is not deferring.

This approach aims to achieve some of the performance benefits of one-stage co-adaptation without the cost of training from scratch. However, it still requires write-access to model parameters and involves the complexity of fine-tuning large models, which may not always be feasible or desirable (e.g., due to concerns about catastrophic forgetting).

4.4 The Trade-Off Between Frameworks

The choice between one-stage, two-stage, and post-hoc learning represents a fundamental design trade-off in L2D, balancing performance potential against practical constraints (Table 2). Ultimately, the choice of an L2D framework is dictated by the specific constraints of the application, including model accessibility, computational resources, and the required level of system robustness.

Table 2: Trade-offs among L2D training frameworks.

Criterion	One-Stage (Joint)	Two-Stage (Fixed Predictor)	Post-Hoc Fine-Tuning
Performance potential	Highest ceiling via full co-adaptation and specialization.	Bounded by the fixed pre-trained predictor; learns optimal routing only.	Intermediate: some co-adaptation without training from scratch.
Computational cost (training)	Highest (end-to-end training).	Lowest for deferral (train a lightweight rejector only).	Moderate (fine-tune predictor + train deferral).
Robustness to expert changes	Lower; specialization can be brittle if expert identity/availability or constraints change.	Higher; general-purpose predictor remains intact.	Intermediate; partial specialization can reduce robustness.
Specialization behavior	Strong specialization on non-deferred regions.	Minimal specialization (predictor is frozen).	Some specialization due to co-adaptation during fine-tuning.
Model accessibility	Requires full "white-box" access to predictor.	Works with black-box/API predictors.	Requires write access to predictor weights.
Modularity	Low; predictor and deferral tightly coupled.	High; plug-and-play rejector around a fixed predictor.	Moderate; coupled through fine-tuning.

5 Optimization & Theoretical Foundations (Surrogate Losses & Guarantees)

The choice of methodological framework (§4) dictates the optimization strategy. A central challenge across all frameworks is the design of tractable surrogate losses suitable for the task that provide strong theoretical guarantees. This section synthesizes the history of surrogate loss design for the one-stage setting (§5.2). Following this, we discuss optimization the more recently proposed two-stage setting (§5.3). We begin with background on the related theoretical guarantees for surrogate loss functions required to comprehend this section (§5.1).
Notation. Symbols follow §3.1 (Table 1). Target deferral losses are the score-based (SB) and predictor-rejector (PR) objectives in Eqs. 1–2.

5.1 Background: Theoretical Guarantees for Surrogate Loss Functions

Having introduced the problem formulations (§3) and methodological frameworks (§4) of L2D, we now turn to the question: *what guarantees do we have that minimizing a surrogate loss will lead to a system that is near-optimal for the original deferral task?* In this subsection, we describe a hierarchy of theoretical guarantees proposed to evaluate such surrogate losses. These guarantees apply generally to both the PR and SB formulations. We review the three principal guarantees of **Bayes-consistency**, **realizable $\mathcal{H}$-consistency**, and **$\mathcal{H}$-consistency bounds**.

To formalize these guarantees, we define the expected risk. For a distribution $\mathcal{D}$, a loss function L, and a hypothesis f from a class $\mathcal{F}$ (where f represents the learnable components, e.g., (h, r) or s), the expected risk is

$$\mathcal{E}_L(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(f, x, y)].$$

The optimal risk within the class is the best-in-class risk,

$$\mathcal{E}^*_L(\mathcal{F}) = \inf_{f\in\mathcal{F}} \mathcal{E}_L(f).$$

The core objective in surrogate analysis is to ensure that minimizing the *estimation error* of the surrogate, $\mathcal{E}_{\mathcal{L}_{\text{surr}}}(f) - \mathcal{E}^*_{\mathcal{L}_{\text{surr}}}(\mathcal{F})$, also minimizes the estimation error of the true deferral loss, $\mathcal{E}_{\ell_{\text{def}}}(f) - \mathcal{E}^*_{\ell_{\text{def}}}(\mathcal{F})$.

Bayes-Consistency

Bayes-consistency is the most fundamental requirement for a surrogate loss. It addresses an asymptotic question: *if we could minimize the surrogate loss perfectly over the space of all possible measurable functions ($\mathcal{F}_{all}$), would we recover the Bayes-optimal strategy for the true deferral loss?*

Definition 1 (Bayes-Consistency [74])**.** A surrogate loss $\mathcal{L}_{\text{surr}}$ is Bayes-consistent with respect to the target deferral loss ℓ_{def} if, for any sequence of hypotheses $\{f_n\}_{n\in\mathbb{N}} \subset \mathcal{F}_{\text{all}}$, the convergence of the surrogate excess risk to zero implies the convergence of the deferral excess risk to zero:

$$\left[\mathcal{E}_{\mathcal{L}_{\text{surr}}}(f_n) - \mathcal{E}^*_{\mathcal{L}_{\text{surr}}}(\mathcal{F}_{\text{all}})\right] \xrightarrow{n\to\infty} 0 \quad \Longrightarrow \quad \left[\mathcal{E}_{\ell_{\text{def}}}(f_n) - \mathcal{E}^*_{\ell_{\text{def}}}(\mathcal{F}_{\text{all}})\right] \xrightarrow{n\to\infty} 0.$$

Bayes consistency is a population-level guarantee over an unrestricted function class (effectively ignoring model capacity and optimization error). Hence the need for $\mathcal{H}$-dependent guarantees that tie surrogate regret to 0–1 regret for a fixed class.

Realizable $\mathcal{H}$-Consistency

To address the practical limitations of Bayes-consistency, realizable $\mathcal{H}$-consistency provides a stronger guarantee for the important *realizable case*. This setting assumes there exists a "perfect" hypothesis $f^* \in \mathcal{F}$ that achieves zero deferral loss, i.e., $\mathcal{E}_{\ell_{\text{def}}}(f^*) = 0$.

Definition 2 (Realizable $\mathcal{H}$-Consistency [39])**.** A surrogate loss $\mathcal{L}_{\text{surr}}$ is realizable $\mathcal{H}$-consistent with respect to ℓ_{def} if, for any realizable distribution, any hypothesis $f \in \arg\min_{f'\in\mathcal{F}} \mathcal{E}_{\mathcal{L}_{\text{surr}}}(f')$ also achieves zero deferral loss, i.e., $\mathcal{E}_{\ell_{\text{def}}}(f) = 0$.

This property is highly desirable as it guarantees optimality in noise-free settings. The importance of this guarantee was highlighted when it was shown that some prominent Bayes-consistent L2D surrogates do not satisfy realizable $\mathcal{H}$-consistency [56]. This result has motivated the design of new loss functions that are provably both Bayes-consistent and realizable $\mathcal{H}$-consistent [47].

$\mathcal{H}$-Consistency Bounds

While realizable $\mathcal{H}$-consistency provides a crucial guarantee for the noise-free case, $\mathcal{H}$-consistency bounds offer the most powerful and general framework. They are *non-asymptotic*—i.e., they give a quantitative inequality at the population level without taking $n \to \infty$—and they tie surrogate performance directly to target deferral loss for a *specific* (possibly restricted) hypothesis class $\mathcal{F}$ and for *any* data distribution.

Definition 3 ($\mathcal{H}$-Consistency Bound [3, 4])**.** A surrogate loss $\mathcal{L}_{\text{surr}}$ admits an **$\mathcal{H}$-consistency bound** with respect to ℓ_{def} if there exists a non-decreasing concave function $\Gamma : \mathbb{R}^+ \to \mathbb{R}^+$ with $\Gamma(0) = 0$ such that, for all $f \in \mathcal{F}$ and all distributions $\mathcal{D}$,

$$\mathcal{E}_{\ell_{\text{def}}}(f) - \mathcal{E}^*_{\ell_{\text{def}}}(\mathcal{F}) + M_{\ell_{\text{def}}}(\mathcal{F}) \;\le\; \Gamma\left(\mathcal{E}_{\mathcal{L}_{\text{surr}}}(f) - \mathcal{E}^*_{\mathcal{L}_{\text{surr}}}(\mathcal{F}) + M_{\mathcal{L}_{\text{surr}}}(\mathcal{F})\right).$$

An $\mathcal{H}$-consistency bound implies Bayes-consistency (taking $\mathcal{F}$ to be unrestricted and noting the gaps vanish). The function Γ acts as a *transfer* (or calibration) function: small surrogate excess risk forces small target excess risk (up to the additive gaps).

Note that, a key feature of this modern formulation is the **minimizability gap**,

$$M_L(\mathcal{F}) \;=\; \mathcal{E}^*_L(\mathcal{F}) \;-\; \mathbb{E}_x\left[\inf_{f\in\mathcal{F}} \mathbb{E}_{y|x}\left[L(f, x, y)\right]\right],$$

which measures how far the *best-in-class global* risk is from the expected *per-x* best-in-class conditional risk. It vanishes when $\mathcal{F}$ contains a conditional risk minimizer almost surely, and it isolates limitations intrinsic to the chosen class $\mathcal{F}$ (distinct from the usual approximation error that compares $\mathcal{F}$ to an unrestricted class). Accounting for $M_L(\mathcal{F})$ yields bounds that are both sharper and more faithful to practical modeling constraints. Recent work derives L2D surrogates that admit strong $\mathcal{H}$-consistency bounds, providing theoretical justification for their use in practice [46].

The Interplay Between Consistency Guarantees in L2D

At first glance, these guarantees appear to form a strict hierarchy of strength: $\mathcal{H}$-consistency bounds imply Bayes-consistency, and one might assume they also imply realizable $\mathcal{H}$-consistency. In standard classification, this is often the case. However, a key theoretical insight in the L2D literature is that this relationship breaks down. As shown by Mao et al. [47], the minimizability gap for some L2D surrogates does not necessarily vanish, even in the realizable case. This seemingly technical detail has a profound implication: an $\mathcal{H}$-consistency bound no longer guarantees realizable $\mathcal{H}$-consistency in the L2D setting.

Consequently, because $M_L(\mathcal{F})$ can remain positive even when $\mathcal{F}$ is realizable, the two guarantees decouple. The independence manifests in both directions:

- A surrogate may *have* strong $\mathcal{H}$-consistency bounds yet fail in the simple realizable case. For instance, the SB L2D cross-entropy surrogate (L_{CE}) [57] has such bounds but is not realizable $\mathcal{H}$-consistent [44, 56].
- Conversely, a surrogate may be realizable $\mathcal{H}$-consistent yet lack the stronger non-asymptotic guarantees of an $\mathcal{H}$-consistency bound for general, non-realizable distributions.

This distinction is not just academic; it clarifies why the search for a single surrogate loss that satisfies *all three* properties became a central goal of modern L2D research, a goal first achieved by the unified frameworks of Mao et al. [47].

5.2 Optimization in the One-Stage Setup

In this subsection, the historical developments of surrogate loss functions in the one-stage setup for single- and multi-expert settings are detailed chronologically. Findings are summarized in Table 3. Considerations for selecting surrogate loss following the findings of this subsection are outlined in §5.5.

5.2.1 Single-Expert Settings: Consistency, Calibration, Underfitting, and Realizability

The research on single-expert L2D surrogates began with the initial goal of establishing Bayes-consistency (Def. 1) [57], then moved to address a series of practical and theoretical challenges that arose (calibration, underfitting, and realizability) [8, 38, 70], and culminated in a unified framework satisfying three key theoretical guarantees [47]. We categorize this progression in three distinct generations:

(1) Foundational Bayes-Consistent Surrogates and Their Practical Limitations The initial wave of research in L2D focused on establishing the initial property of Bayes-consistency [74]. The seminal cross-entropy-based surrogate, L_{CE}, proposed by Mozannar and Sontag [57], was the first to be proven Bayes-consistent for the multi-class L2D setup. Following this, a unifying theoretical framework was proposed by Charusaie et al. [9], who demonstrated a general method to convert any consistent multi-class surrogate loss into a consistent, cost-sensitive loss suitable

Table 3: A Summary of Key Developments in One-Stage L2D Surrogate Losses. The table charts the chronological progression from foundational Bayes-consistent losses to unified frameworks with the strongest theoretical guarantees.

Citation	Year	Surrogate (Abbr.)	Bayes-Consistent	Realizable $\mathcal{H}$-Consist.	$\mathcal{H}$-Consist. Bounds	Key Contribution and Significance
— §5.2.1 *Single-Expert Settings: Consistency, Calibration, Underfitting, and Realizability* —						
Mozannar and Sontag [57]	2020	Cross-Entropy (L_{CE})	Yes	No *(per [56])*	Yes *(via* [44])	**Seminal paper.** Introduced the first Bayes-consistent surrogate loss for L2D, establishing the standard framework.
Charusaie et al. [9]	2022	Cost-Sensitive Family (L_ϕ)	Yes	No	Yes *(via* [46])	**Generalized the L2D framework.** Proposed a method to convert any consistent multi-class loss into a consistent L2D loss, creating a broad family of surrogates. First to provide excess risk bounds.
Verma and Nalisnick [70]	2022	One-vs-All (L_{OvA})	Yes	No *(per [56])*	Yes *(via* [44])	Introduced an OvA-based alternative. **First to identify the calibration problem**, showing the softmax surrogate leads to unbounded estimates.
Cao et al. [8]	2023	Asymmetric SM ($L_{\tilde{\psi}}$)	Yes	No	Yes	**Defended the softmax approach.** Showed the calibration issue was due to loss symmetry, not softmax itself. Proposed a bounded, consistent asymmetric softmax loss.
Mozannar et al. [56]	2023	Realizable (L_{RS})	Yes *(per* [47])	Yes**	Yes *(per* [47])	**Critical turning point.** Proved prior losses were **not realizable** and introduced the first realizable surrogate. Raised a key open question about Bayes-consistency.
Liu et al. [38]	2024	Label-Smoothing-Free (L_{LSF})	Yes	No	No	**Solved the underfitting problem** caused by implicit label-smoothing when deferral costs are non-zero. Proposed a "label-smoothing-free" formulation that improves performance and robustness to cost.
Mao et al. [47]	2024c	Realizable L2D (L_{RL2D})	Yes	Yes	Yes	**Solved the open problem** from Mozannar et al. [56]. Provided the first surrogate loss satisfying all three core consistency guarantees in the single-expert setting.
— §5.2.2 *The Multi-Expert Problem: Generalization and Stronger Guarantees* —						
Verma et al. [69]	2023	Multi-Expert OvA and CE (L^J_{SM} & L^J_{OvA})	Yes (per [46])	No	Yes	**First to generalize L2D to multiple experts** with a Bayes-consistent surrogate. Retained the same realizability gap as earlier single-expert work.
Mao et al. [46]	2024b	General L2D (L_{general})	Yes	No	Yes	**First to provide $\mathcal{H}$-Consistency Bounds** in L2D, offering stronger, non-asymptotic guarantees for the multi-expert case.
Mao et al. [49]	2025	Unified L2D (L_Ψ)	Yes	Yes	Yes	**Unified all guarantees for the one-stage, multi-expert case.** Introduced the first surrogate loss to satisfy all three theoretical properties in this setting.

** Realizable $\mathcal{H}$-consistent when $\mathcal{H}$ is closed under scaling, i.e., if $h \in \mathcal{H} \implies \alpha h \in \mathcal{H},\ \forall\, \alpha \in \mathbb{R}$.

for deferral. While this created a broad family of potential surrogates, two practical issues with this framework arose.

First, a **calibration problem** was identified by Verma and Nalisnick [70], demonstrating that the score-based cross-entropy surrogate L_{CE} [57] leads to unbounded and poorly calibrated estimates of expert correctness, a problem their bounded One-vs-All (OvA) surrogate mitigated. The cause of this calibration issue was subsequently diagnosed by Cao et al. [8], who proved that the unboundedness was not a flaw in the softmax function itself, but a fundamental consequence of using any standard, symmetric loss within the general framework of Charusaie et al. [9]. More importantly, their work re-contextualized the field by showing that the successful, bounded losses from both their own work and that of Verma and Nalisnick [70] could be understood as special instances of this unified framework, succeeding precisely because they are derived from novel, asymmetric base losses. This work solidified the understanding that achieving well-calibrated estimates requires a careful, asymmetric design of the underlying loss.

Second, even with these well-calibrated, consistent losses, a further practical limitation was identified: **underfitting induced by non-zero deferral costs**. First noted by Narasimhan et al. [58], the performance of these surrogates can degrade significantly when a non-zero fixed cost $c_0 > 0$ is introduced for consulting an expert, such that the total cost is $c(x, y) = c_0 + \mathbb{1}_{\mathcal{E}(x)\neq y}$. This issue was systematically addressed by Liu et al. [38], who diagnosed that the general framework of Charusaie et al. [9] implicitly introduces a redundant label-smoothing term whenever $c_0 > 0$. This term flattens the training distribution and degrades the performance of the classifier component as the deferral cost increases. To solve this, they proposed a novel "label-smoothing-free" formulation L_{LSF} which eliminates this harmful effect by applying the cost penalty in a targeted, non-uniform manner, using the model's own intermediate prediction to guide the penalty. This provided a family of surrogates that maintain high performance in realistic, cost-sensitive scenarios.

(2) The Challenge of Realizable $\mathcal{H}$-Consistency A turning point in the field came when Mozannar et al. [56] showed a significant limitation in all prior approaches. They demonstrate that the prior popular Bayes-consistent surrogates are not realizable $\mathcal{H}$-consistent (Def. 2). This property is of practical importance: it guarantees that if a perfect, zero-error predictor-rejector pair exists within the chosen hypothesis class (e.g., linear models), the learning algorithm is guaranteed to find it. The failure to satisfy this property means that even in a simple, noise-free setting where a perfect linear solution exists, these established surrogates could converge to a suboptimal solution with non-zero error. Mozannar et al. [56] make this abstract concern concrete with a synthetic counterexample, highlighting that Bayes-consistency alone is insufficient and motivating the search for surrogates with stronger guarantees.

In the same work, Mozannar et al. [56] propose a new surrogate loss to address this gap, which they term the *Realizable Surrogate* (L_{RS}). The key innovation is to make the human's correctness acts as a switch *inside* the logarithm of a softmax-like term, rather than as an external weight. This seemingly small change has a profound effect: the authors prove that L_{RS} is indeed realizable $\mathcal{H}$-consistent for any hypothesis class closed under scaling. However, this work left a theoretical question unanswered: while the new loss is realizable, it is not proven to be Bayes-consistent.

(3) A Unified Framework with All Theoretical Guarantees The open question from Mozannar et al. [56] was recently resolved by Mao et al. [47]. Their work provides two key contributions that unify the theoretical landscape for single-expert L2D. First, they formally prove that the surrogate L_{RS} is, in fact, also Bayes-consistent, establishing L_{RS} as the first surrogate loss to satisfy both guarantees. More significantly, they introduced a broad, unified family of surrogate

losses, L_{RL2D}, parameterized by a non-increasing function Ψ. This family is constructed by re-deriving the surrogate from first principles and takes the general form:

$$L_{\text{RL2D}}(h, x, y) = c(x, y) l_{\text{comp}}(h, x, y) + (1 - c(x, y)) \tilde{l}_{\text{comp}}(h, x, y) \tag{6}$$

where l_{comp} is a standard comp-sum loss (e.g., cross-entropy, MAE) that maximizes the correct-class score, and $\tilde{l}_{\text{comp}}$ modifies it to maximize the sum of the correct-class and deferral scores. Under the mild conditions on Ψ (non-increasing, $\Psi\left(\frac{2}{3}\right) > 0$, and $\lim_{t \to 1} \Psi(t) = 0$) the formulation is Bayes-consistent, realizable $\mathcal{H}$-consistent, and admits explicit $\mathcal{H}$-consistency bounds (Def. 3). This yields a family of losses parameterized by the link Ψ and cost c.

5.2.2 The Multi-Expert Problem: Generalization and Stronger Guarantees

While the single-expert setting provides a crucial theoretical foundation, many real-world, practical applications involve routing decisions among a pool of multiple experts with diverse specializations and consultation costs. Early approaches to the multi-expert problem, such as the work by Hemmer et al. [27], extended the existing one-stage framework by simultaneously learning a classifier and a multi-expert routing system, drawing inspiration from Mixture-of-Experts models [31]. However, these initial methods were often heuristic and lacked theoretical guarantees, a limitation later addressed by a new generation of research focused on developing provably consistent surrogate losses for the multi-expert setting.

First-Generation Consistent Surrogates The first principled extension of consistent surrogate losses to the multi-expert setting was presented by Verma et al. [69], who generalized the initial single-expert surrogate losses (i.e., the softmax and One-vs-All (OvA) formulations) to handle J experts. Their multi-expert softmax surrogate takes the form:

$$\Phi^{J}_{SM}(s, x, y, \mathbf{m}) = -\log\left(\frac{\exp s(x, y)}{\sum_{a \in \mathcal{Y}_\perp} \exp s(x, a)}\right) - \sum_{j=1}^{J} \mathbb{1}_{\mathcal{E}_j = y} \log\left(\frac{\exp s(x, \perp_j)}{\sum_{a \in \mathcal{Y}_\perp} \exp s(x, a)}\right) \tag{7}$$

Here, $\mathcal{Y}_\perp$ is the augmented label space including J distinct deferral options $\{\perp_1, \ldots, \perp_J\}$, $s(x, a)$ is the score for action a, and $\mathbf{m} = \{\mathcal{E}_1(x), \ldots, \mathcal{E}_J(x)\}$ is the vector of J expert predictions. Verma et al. [69] proved that this loss is Bayes-consistent (Def. 1) with the optimal multi-expert routing strategy. However, these first-generation multi-expert surrogates inherited the same theoretical limitation as their single-expert counterparts in that they were later shown to lack the stronger guarantee of realizable $\mathcal{H}$-consistency (Def. 2) by Mozannar et al. [56].

Principled Generalizations and $\mathcal{H}$-Consistency Bounds A significant theoretical step forward was made by Mao et al. [46], who introduced a principled derivation for generalizing multi-expert surrogate losses. Rather than extending specific losses, they show how any multiclass surrogate ℓ that admits an $\mathcal{H}$-consistency bound (Def. 3) can be adapted to the multi-expert deferral task, and then prove that the resulting family inherits these bounds. These guarantees are hypothesis-set–specific and non-asymptotic, and the paper further gives the first finite-sample learning bound for multi-expert L2D. Unlike earlier excess-risk bounds (i.e., classical surrogate-to-target bounds based on approximation error), $\mathcal{H}$-consistency bounds account for the hypothesis class and the minimizability gap, thereby implying Bayes-consistency while providing a more practical quantitative link between surrogate and target losses. The authors do not claim realizable $\mathcal{H}$-consistency for this framework.

A Unified Framework for Multi-Expert Deferral Unification of all three major consistency guarantees for the multi-expert setting was recently achieved in the framework of Mao et al. [49]. Their work introduces a novel family of surrogate losses, L_Ψ, designed from first principles to satisfy all three theoretical properties simultaneously. They first derived an alternative formulation of the true deferral loss, then systematically replaced its indicator functions with smooth surrogates from the broad class of comp-sum losses (parameterized by a non-increasing function Ψ). This results in the following general form for their surrogate family:

$$L_\Psi(s, x, y) = \left[\sum_{j=1}^{J} c_j(x, y) + 1 - J\right] \Psi\left(\frac{e^{s(x,y)}}{\sum_{y' \in \mathcal{Y}_\perp} e^{s(x,y')}}\right) + \sum_{j=1}^{J} [1 - c_j(x, y)] \Psi\left(\frac{e^{s(x,y)} + e^{s(x,\perp_j)}}{\sum_{y' \in \mathcal{Y}_\perp} e^{s(x,y')}}\right) \quad (8)$$

where $s(x, y)$ is the score for label y, $s(x, \perp_j)$ is the score for deferring to expert j, J is the number of experts, and $\mathcal{Y}_\perp$ is the augmented label space. The key innovation lies in the final term, where the scores for the correct label y and the correct deferral option $\perp_j$ are summed in the numerator, a modification crucial for achieving realizability. Their main theoretical results prove that for specific choices of Ψ (e.g., corresponding to mean absolute error, where $\Psi(t) = 1 - t$), the resulting surrogate loss is simultaneously Bayes-consistent, realizable $\mathcal{H}$-consistent, and admits $\mathcal{H}$-consistency bounds. This work provides the most complete theoretical foundation to date for one-stage, multi-expert L2D, closing theoretical gaps and offering practitioners a principled and flexible framework with the strongest possible guarantees.

5.3 Optimization in the Two-Stage Setup

In the two-stage setting (§4.2), we have a frozen, pre-trained predictor h. The optimization challenge is to learn an optimal deferral mechanism (scorer s_d or rejector r) that routes inputs between h and the J experts. The development of optimization strategies in this framework began with the work of Mao et al. [43], who introduced the first consistent surrogates. Subsequent work by Mao et al. [49] addressed the additional theoretical gaps.

Foundational Two-Stage Surrogates

Mao et al. [43] proposed consistent surrogate losses for both the Score-Based (SB) and Predictor-Rejector (PR) formulations in the two-stage setting. We detail both below.

Score-Based Formulation. Let $s_h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ denote per-class scores (e.g., logits or calibrated probabilities) produced by the predictor h. The predicted label is $h(x) = \arg\max_{y \in \mathcal{Y}} s_h(x, y)$. In the SB approach, the goal is to learn a deferral scoring function $s_d : \mathcal{X} \times [J] \to \mathbb{R}$. This is achieved by constructing a unified scoring function $\bar{s}_d : \mathcal{X} \times (\{0\} \cup [J]) \to \mathbb{R}$, where the predictor h (indexed by 0) contributes its confidence score:

$$\bar{s}_d(x, j) = \begin{cases} \max_{y' \in \mathcal{Y}} s_h(x, y'), & \text{if } j = 0 \\ s_d(x, j), & \text{if } j \in [J] \end{cases} \quad (9)$$

The decision is action$(x) = \operatorname{argmax}_{j \in \{0\} \cup [J]} \bar{s}_d(x, j)$. The surrogate loss, constructed using a standard multi-class surrogate ℓ_2 (e.g., logistic loss), is defined as:

$$L_h(s_d, x, y) = \mathbb{1}_{h(x)=y} \cdot \ell_2(\bar{s}_d, x, 0) + \sum_{j=1}^{J} \bar{c}_j(x, y) \cdot \ell_2(\bar{s}_d, x, j). \quad (10)$$

where $\bar{c}_j(x, y) = 1 - c_j(x, y)$ is the expert reward. This loss encourages high scores for correct actions: it rewards the predictor if it is correct, and rewards expert j weighted by their correctness $\bar{c}_j(x, y)$.

Predictor-Rejector Formulation. In the PR formulation, the rejector is generalized to $r : \mathcal{X} \to \mathbb{R}^J$. The system uses h if $0 < \min_{j\in[J]} r_j(x)$; otherwise, it defers to $\arg\min_{j\in[J]} r_j(x)$. By defining an associated hypothesis $\bar{r}$ where $\bar{r}(x, 0) = 0$ and $\bar{r}(x, j) = -r_j(x)$, the surrogate loss is defined analogously:

$$L_h(r, x, y) = \mathbb{1}_{h(x)=y} \cdot \ell_2(\bar{r}, x, 0) + \sum_{j=1}^{J} \bar{c}_j(x, y) \cdot \ell_2(\bar{r}, x, j). \tag{11}$$

Theoretical Guarantees and Limitations. Mao et al. [43] proved that both surrogates admit $\mathcal{R}$-consistency bounds (Def. 3) (implying Bayes-consistency (Def. 1)). However, they only established realizable $\mathcal{R}$-consistency (Def 2) for the restricted case of *constant* deferral costs (i.e., $c_j(x, y) = \beta_j$). This left a notable gap, as the guarantee did not hold for the common scenario where costs are based on expert misclassification error ($c_j(x, y) = \mathbb{1}_{\mathcal{E}_j(x)\neq y}$).

Achieving Realizable $\mathcal{R}$-Consistency in Two-Stage Deferral

This limitation was recently addressed by Mao et al. [49]. They introduced a new family of surrogate losses designed specifically to achieve realizable $\mathcal{R}$-consistency for classification-error-based costs in the two-stage setting.

This framework simplifies the objective by focusing only on routing among a set of J experts, assuming the pre-trained predictor h is simply one of the available experts (e.g., $h = \mathcal{E}_1$). The goal is to learn a routing function $r : \mathcal{X} \times [J] \to \mathbb{R}$ to minimize the two-stage target deferral loss $L_{\text{tdef}}(r, x, y) = \sum_{j=1}^{J} c_j(x, y)\mathbb{1}_{r(x)=j}$.

Mao et al. [49] propose the following surrogate loss family, based on the comp-sum [44] structure parameterized by a decreasing function Ψ (e.g., $\Psi(u) = 1 - u$ or $\Psi(u) = -\log(u)$):

$$L_\Psi(r, x, y) = \sum_{j=1}^{J} \left(\sum_{j'\neq j} c_{j'}(x, y) - J + 2 \right) \Psi\left(\frac{e^{r(x,j)}}{\sum_{j'\in[J]} e^{r(x,j')}} \right). \tag{12}$$

This formulation applies a weighted penalty to the standard comp-sum loss for each expert j, where the weight depends on the costs of the other experts.

Unified Guarantees and Assumptions. If (i) the score class $\mathcal{R}$ is closed under positive scaling, (ii) costs are classification-error costs as above, (iii) for every (x, y) *at most one expert is perfect* (i.e., at most one j has $c_j(x, y) = 0$), and (iv) $\lim_{u\to 1^-} \Psi(u) = 0$, then L_Ψ is realizable $\mathcal{R}$-consistent with respect to L_{tdef}. Beyond realizability, Ψ surrogates admit $\mathcal{R}$-consistency bounds (hence Bayes-consistency) under symmetry and completeness of $\mathcal{R}$ and the condition $\sum_{j'\neq j} c_{j'}(x, y) \geq J - 2$ for all $j, (x, y)$ (equivalently, on any instance, at most two experts can be correct).

The "at most one perfect expert" assumption enforces uniqueness of the zero-cost expert. It is plausible when experts are deliberately specialized with minimal overlap, but can be strong in settings where multiple experts often agree. The other requirements (scaling closure of $\mathcal{R}$ and $\lim_{u\to 1^-} \Psi(u) = 0$) are mild and satisfied by standard score classes and common Ψ choices.

5.4 An Alternative Formulation: Dependent Bayes Optimality

Although the majority of research has focused on developing surrogates that are consistent with the standard Bayes-optimal rule (Eq. 3), alternative research has emerged that questions the fundamental formulation of the problem itself, namely the statistical dependence between the model and expert predictions. While the standard optimality condition estimates and compares the independent marginal quantities for model and expert confidences, Wei et al. [71] introduce **Dependent Bayes Optimality**, which recasts the deferral decision as a comparison between two mutually exclusive events: the system should defer if and only if it is more probable that (1) the model is wrong, and the expert is correct, than that (2) the model is correct, and the expert is wrong.

Based on this principle, the authors proposed a new surrogate, the Dependent Cross-Entropy loss $L^{\perp}_{\mathrm{DCE}}$, which directly implements the deferral principle during training by directly encouraging a relative ordering of model prediction and deferral scores based on the observed dependence patterns in the training triplets. This approach bypasses the intermediate step of confidence estimation, offering a simpler and potentially more direct path to learning an effective deferral policy.

5.5 Key Considerations for Implementation

This section is intended to equip L2D researchers with principles to inform design choices for building L2D systems that are reliable for the intended task, rather than identifying a single "best" loss. In short: pick the framework that fits your deployment constraints; select surrogates for calibration and guarantees; and evaluate on coverage/cost and reliability, not accuracy alone (Figure 7).

Figure 7: Key Considerations for Implementation

Choose by setting, not by headline accuracy. On common benchmarks, many consistent surrogates tie on system accuracy [49]. Prioritize *calibration of expert-correctness*, *robustness to consultation costs*, and *theoretical guarantees* (cf. Table 3).

Match architecture to constraints. Use *score-based* setups for *one-stage* classification; *predictor–rejector* when doing *two-stage*, *regression*, or when experts are heterogeneous/black-box.

One-stage (joint) rule of thumb. Start with *bounded/asymmetric SB* surrogates for calibration; when realizable $\mathcal{H}$-consistency or full $\mathcal{H}$-bounds matter, choose unified/realizable families (Table 3). Avoid label-smoothing side-effects if you include a non-zero base consultation cost.

Two-stage (fixed predictor) rule of thumb. Use surrogates that admit $\mathcal{H}$-consistency bounds; when costs are error-based, prefer *realizable* two-stage objectives. This framework stays modular and resilient to expert/model changes.

Alternative optimality (when to care). Dependent Bayes optimality is useful when *model–expert dependence* is itself the object of interest; otherwise the standard Bayes target suffices.

Report what matters. Beyond accuracy: (i) Accuracy–Coverage curves; (ii) calibration of expert-correctness (ECE or equivalent); (iii) cost/coverage sensitivity; (iv) workload/budget compliance; and a brief rationale for the chosen guarantees.

6 L2D Task Generalizations

The standard L2D frameworks, as discussed in the preceding sections, were primarily developed for multi-class classification. However, L2D's core principle of optimally routing between an automated model and an expert is a concept with far broader applicability. This section describes

recent efforts to generalize the L2D field beyond simple classification. We will examine how L2D has been reformulated for regression tasks with continuous outputs (§6.1), for multi-task problems requiring complex, structured predictions (§6.2), for settings where the goal is to select a committee of top agents rather than a single one (§6.3), and for sequential scenarios where the objective shifts from immediate accuracy to optimizing long-term outcomes (§6.4).

Notation. We retain the global setup of §3.1 (Table 1). This section changes the task space and/or decision objects per subsection (regression, multi-task, top-k, sequential, causal); each subsection below states its deviations explicitly.

6.1 Regression with Deferral

Extension of L2D to regression tasks presents unique challenges due to the continuous nature of the output space, $\mathcal{Y} \subset \mathbb{R}$. In this setting, the predictor $h : \mathcal{X} \to \mathbb{R}$ is a regression model, and the J experts $\mathcal{E}_1, \ldots, \mathcal{E}_J$ are also regressors. The objective remains to minimize the system loss, but the classification error (e.g., 0-1 loss) is replaced by a standard regression loss L (e.g., squared or absolute error), which is typically assumed to be bounded. This change has architectural consequences. As established in §3, the score-based formulation (§3.2) is fundamentally inapplicable to regression. The core mechanism of augmenting the label space $\mathcal{Y}$ with deferral options $\{\perp_1, \ldots, \perp_J\}$ and learning a unified scoring function over them breaks down, as it is intractable to simultaneously score every real value in the infinite space $\mathcal{Y}$ and the discrete deferral options.

This incompatibility necessitates the adoption of the predictor-rejector (PR) architecture (§3.2), which naturally handles regression by decoupling the continuous output of the predictor h from the selection of the decision-making expert. The first comprehensive framework for regression with multi-expert deferral, addressing both one-stage and two-stage scenarios, was introduced by Mao et al. [48]. Below, we detail the one-stage setup.

The Generalized Predictor-Rejector Formulation for Regression To handle multiple experts in the regression context, the standard binary PR formulation (which decides only whether to predict or defer) must be generalized. In this framework, the rejector r is defined as a multi-output scoring function, $r : \mathcal{X} \times \{0, 1, \ldots, J\} \to \mathbb{R}$. This function scores $J + 1$ options: option 0 corresponds to the predictor h, and options 1 through J correspond to deferral to the respective experts. The system's decision, denoted $r^*(x)$, is determined by which expert receives the highest score: $r(x) = \text{argmax}_{j \in \{0,\ldots,J\}} r(x, j)$. If $r^*(x) = 0$, the system predicts $h(x)$; otherwise, it defers to expert $r^*(x)$.

The target deferral loss L_{def} adapts the PR objective to this multi-expert regression setting:

$$L_{\text{def}}(h, r, x, y) = L(h(x), y) \cdot \mathbb{1}_{r^*(x)=0} + \sum_{j=1}^{J} c_j(x, y) \cdot \mathbb{1}_{r^*(x)=j} \tag{13}$$

where L is the regression loss and $c_j(x, y)$ is the cost of deferring to expert j (e.g., the expert's own regression loss, potentially plus a base cost).

A Predictor-Rejector Surrogate via Transformation Direct optimization of Eq. 13 is intractable. In deriving a tractable surrogate for the one-stage setting (where h and r are learned jointly), Mao et al. [48] employ a novel algebraic transformation of the target loss. The main insight is to rewrite L_{def} to expose a structure that mimics a weighted multi-class classification problem over the $J + 1$ choices. This reformulation allows for leveraging well-understood multi-class surrogate losses from classification, despite the underlying regression task.

By replacing the indicator functions in the transformed loss with a general, smooth multi-class surrogate loss ℓ (such as the logistic loss) that upper-bounds the 0-1 loss, one can derive a one-stage surrogate loss L:

$$L_\ell(h, r, x, y) = \left[\sum_{j=1}^{J} c_j(x, y)\right] \ell(r, x, 0) + \sum_{j=1}^{J} \left[L(h(x), y) + \sum_{k \neq j} c_k(x, y)\right] \ell(r, x, j) - (J-1)L(h(x), y). \tag{14}$$

Here, $\ell(r, x, j)$ represents the value of the chosen multi-class surrogate ℓ when the target class is j. This loss function is jointly differentiable with respect to the parameters of h and r, enabling end-to-end training.

Theoretical Guarantees A significant contribution of [48] is the provision of strong theoretical guarantees. Mao et al. [48] prove that their proposed surrogate loss admits $\mathcal{H}$-consistency bounds (specifically, ($\mathcal{H}$,$\mathcal{R}$)-consistency bounds in the predictor-rejector setting), which are stronger and more practical than traditional Bayes-consistency. These guarantees are non-asymptotic and specific to the chosen hypothesis sets, confirming that minimizing the surrogate loss L_ℓ is a principled approach to minimizing the true target deferral loss in regression. Furthermore, this framework is highly general. It accommodates any bounded regression loss and handles instance- and label-dependent costs. Notably, it generalizes the work of Cheng et al. [12], which can be viewed as a special case of this one-stage formulation restricted to a single expert, the squared loss, and label-independent costs.

6.2 Multi-Task Learning

The first principled framework capable of performing several interdependent tasks simultaneously was introduced by Montreuil et al. [53], focusing on the two-stage L2D setup (cf. §4.2). This framework assumes a pre-trained multi-task model and a set of experts with fixed behaviors. The core learning problem is to train a separate rejector function that optimally allocates each query to the most suitable expert, balancing the accuracy-cost trade-off across all tasks.

Multi-Task Problem Setup

The notation from the predictor-rejector architecture (§3) is extended here. The input space is $\mathcal{X}$, the classification label space is $\mathcal{Y} = \{1, \ldots, K\}$, and we introduce a regression target space $\mathcal{T} \subseteq \mathbb{R}$. A data point is a triplet $z = (x, y, t) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y} \times \mathcal{T}$. The multi-task architecture typically relies on shared representation learning. A backbone encoder $w : \mathcal{X} \to \mathcal{Q}$ maps the input to a latent feature space $\mathcal{Q}$. Task-specific heads then operate on this shared representation.

In this two-stage multi-task setup, the system consists of:

1. **A fixed primary model** $g \in \mathcal{G}$. This is a multi-head network defined by the backbone w, a classification scoring head $h : \mathcal{Q} \times \mathcal{Y} \to \mathbb{R}$, and a regression head $f : \mathcal{Q} \to \mathcal{T}$. It produces a joint prediction $g(x) = (h \circ w(x), f \circ w(x))$ based on the shared representation $w(x)$.
2. **A set of J fixed experts**. Each expert $\mathcal{E}_j$ for $j \in [J]$ also produces a joint prediction $\mathcal{E}_j(x) = (\mathcal{E}_j^h(x), \mathcal{E}_j^f(x))$, where $\mathcal{E}_j^h(x) \in \mathcal{Y}$ and $\mathcal{E}_j^f(x) \in \mathcal{T}$ are its classification and regression predictions, respectively.
3. **A learnable rejector function** $r \in \mathcal{R}$, where $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$. The rejector scores each expert in the set $\mathcal{A} = \{0\} \cup [J]$ (where agent 0 is the primary model g). The final allocation decision $r^*(x)$ is determined by the highest score: $r^*(x) = \arg\max_{j \in \mathcal{A}} r(x, j)$

For ease of notation, the combined set of the primary model and the experts simply as the *agents*. The cost of selecting agent j, c_j, is based on the agent's joint prediction. Let $a_j(x)$ denote the prediction of agent j (where $a_0(x) = g(x)$ and $a_j(x) = \mathcal{E}_j(x)$ for $j > 0$). For the primary model ($j = 0$), the cost is $c_0(a_0(x), z) = \rho(a_0(x), z)$, where ρ is a loss function measuring the joint prediction error. For an expert ($j > 0$), the cost includes the error and a consultation cost $\beta_j \geq 0$, such that $c_j(a_j(x), z) = \rho(a_j(x), z) + \beta_j$. Importantly, ρ is flexible; it can be an aggregate metric or designed to balance the importance of the tasks, often decomposing into a weighted sum.

The True Deferral Loss

The objective is to learn a rejector r that minimizes the true deferral loss, ℓ_{def}, which quantifies the cost incurred by the rejector's chosen agent. Let $z = (x, y, t) \in \mathcal{Z}$ be a data point and $r \in \mathcal{R}$ be a rejector. The true deferral loss is defined as

$$\ell_{def}(r, g, m, z) = \sum_{j=0}^{J} c_j(a_j(x), z) \cdot \mathbb{1}_{r(x)=j}. \tag{15}$$

Due to the indicator function tied to the non-differentiable arg max rule, the true deferral loss is discontinuous and cannot be optimized directly. To overcome this, Montreuil et al. [53] introduce a novel family of convex surrogate losses based on the cross-entropy family [44].

The Multi-Task Deferral Surrogate Loss

To overcome this optimization challenge, a convex, non-negative surrogate loss is proposed [53]. Let Φ^{ν}_{01} be a multi-class surrogate loss from the cross-entropy family, parameterized by $\nu \geq 0$. The surrogate deferral loss Φ^{ν}_{def} for $J + 1$ agents is defined as:

$$\Phi^{\nu}_{def}(r, g, m, z) = \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \cdot \Phi^{\nu}_{01}(r, x, j), \tag{16}$$

where the weights τ_j are the aggregated *complementary* costs, defined as the total cost of choosing any agent *other than* agent j:

$$\tau_j(g(x), m(x), z) = \sum_{i \in \mathcal{A}, i \neq j} c_i(a_i(x), z).$$

The intuition behind this surrogate lies in the weighting scheme. The weight τ_j represents the total cost the system would incur if it selected any agent besides agent j. Consequently, an agent k with a low expected true cost c_k will have a correspondingly high aggregated complementary cost τ_k. This large τ_k multiplies the term $\Phi^{\nu}_{01}(r, x, k)$, creating strong pressure to minimize it. The rejector achieves this by assigning a high score $r(x, k)$, thus guiding the learned deferral policy towards the optimal allocation.

Theoretical Guarantees

This formulation is supported by strong theoretical guarantees, establishing it as the first principled framework for multi-task L2D. Montreuil et al. [53] prove that the surrogate loss Φ^{ν}_{def} is both Bayes-consistent and ($\mathcal{G}$,$\mathcal{R}$)-consistent. The authors additionally provide several theoretical insights specific to this setting, such as explicit consistency bounds, minimizability gap analysis, and encoder-aware bounds.

6.3 Top-k Classification and Deferral

In *top-k* classification a model predicts a set of the most probable labels to better handle ambiguity [34, 72]. Given that aggregating predictions from multiple sources in high-stakes domains can increase reliability [19, 41], this has motivated work towards adapting the top-k concept to create *Top-k L2D*. This framework generalizes L2D by allowing the system to select a *set* of the top-k most cost-effective entities for a given input. Crucially, this set can consist of multiple class labels (enabling top-k autonomous classification), a committee of experts (enabling multi-expert deferral), or even a hybrid of both, providing a richer and more flexible decision-making process. There exist top-k setups for both the one- and two-stage L2D frameworks.

One-Stage Top-k

The first one-stage framework for top-k L2D, unifying prediction and deferral under a single end-to-end objective, was introduced by Montreuil et al. [51]. Their key innovation is to reformulate the problem using a unified, cost-sensitive scoring function over an augmented set of all possible actions (i.e., predicting a class or deferring to an expert).

Let $\mathcal{A} = \{1, ..., K\} \cup \{K+1, ..., K+J\}$ be the set of all possible entities, where indices $j \leq K$ correspond to direct class predictions and indices $j > K$ correspond to deferring to expert m_{j-K}. Note that here k is the size of the selected set (classes and/or experts), while K is the number of classes.

A Unified Cost-Sensitive Target Loss Instead of a hard-coded distinction between prediction and deferral, Montreuil et al. [51] define a uniform cost structure that applies to every entity $j \in \mathcal{A}$. A prediction function $a_j : \mathcal{X} \to \mathcal{Y}$ is associated with each entity, where $a_j(x) = j$ for a class label and $a_j(x) = m_{j-K}(x)$ for an expert. The augmented cost $\tilde{c}_j(x, y)$ for selecting entity j is then:

$$\tilde{c}_j(x, y) = \tilde{\alpha}_j \mathbb{1}_{a_j(x) \neq y} + \beta_j \tag{17}$$

Here, $\tilde{\alpha}_j \geq 0$ is a penalty for misclassification, and $\beta_j \geq 0$ is a fixed consultation cost. For class labels ($j \leq K$), β_j is typically set to zero. This unified structure enables framing the L2D problem as one of cost-sensitive entity selection.

The learning objective adapts existing score-based approaches [46, 57] to train a score-based model $s : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ which identifies the most cost-effective entities. The top-k prediction set, $H_k(x) \subseteq \mathcal{A}$, contains the k entities with the highest scores assigned by $s(x, \cdot)$. The *Top-k Score-Based Target Loss* is the total cost incurred by selecting this set:

$$\ell_{\text{def},k}(H_k(x), y) = \sum_{j \in H_k(x)} \tilde{c}_j(x, y) \tag{18}$$

Note that this target loss is a direct generalization of the standard top-1 score-based L2D loss, which is recovered when $k = 1$.

A k-Independent Surrogate Loss The authors derive a convex surrogate loss that, critically, is independent of the cardinality parameter k. This allows a single model to be trained and then deployed in any top-k regime without retraining. The proposed surrogate is:

$$\Phi_{\text{def}}(s, x, y) = \sum_{j=1}^{K+J} \tau_j(x, y) \cdot \Phi_{\text{ce}}(s, x, j) \tag{19}$$

where Φ_{ce} is a cross-entropy-based loss function and $\tau_j(x, y) = \sum_{i \in \mathcal{A}, i \neq j} \tilde{c}_i(x, y)$ is the *complementary cost*; the total cost of selecting all entities *except* j. Minimizing this objective encourages the model to assign high scores to entities with low complementary costs (and thus low true costs). The authors provide strong theoretical support for this formulation, proving that it is both Bayes-consistent and $\mathcal{H}$-consistent under mild assumptions, ensuring that optimizing the surrogate leads to a near-optimal policy for the true top-k deferral task.

Adaptive Deferral via Top-k(x)

A key extension of this framework is an adaptive variant, top-$k(x)$, which dynamically selects the number of consulted entities per input. This is achieved by learning a separate cardinality function $k : \mathcal{X} \to \{1, \ldots, K + J\}$, which predicts the optimal committee size for each query x. The function $k(x)$ is trained to balance predictive accuracy (e.g., measured by a majority vote of the selected entities) against the total consultation cost of the selected committee. This allows the model to allocate its budget more efficiently, querying a large committee for complex inputs while being frugal on simpler ones, leading to superior accuracy-cost trade-offs in practice.

Adaptation to the Two-Stage Setup

The top-k generalization is equally applicable to the practical two-stage setting (discussed in §4.2), where the underlying predictive agents are fixed and pre-trained. Montreuil et al. [52] introduced the first framework for this scenario. They demonstrate that the core concepts described above of the use of a k-independent surrogate loss based on complementary costs and the adaptive $k(x)$ mechanism can be successfully adapted to train a rejector function that optimally ranks the fixed agents, allowing for flexible committee formation without retraining the base models.

6.4 Sequential Learning to Defer

The standard L2D setup treats inputs as i.i.d. and optimizes myopic outcomes. In sequential settings, the deferral choice affects future states and cumulative utility. Two regimes appear in the literature: (i) sequential *decision-making* [32] and (ii) sequential *outputs* (autoregressive prediction) [65]. Both require objectives that account for downstream effects of deferral.

Sequential Decision-Making.

Sequential Learning to Defer (SLTD) [32] instantiates L2D in a Markov Decision Process. It assumes a fixed target policy π_{tar} and an expert policy π_0 (observed via offline data), and learns a deferral policy that selects, at each state, whether to act with π_{tar} or defer to π_0 (two-stage; cf. §4.2). The deferral rule is *long-term*: it compares the value of deferring now versus executing π_{tar} for one step and possibly deferring later (a delayed comparison with a "mixture" continuation). This enables *pre-emptive* deferral when near-term actions look acceptable but degrade downstream value. SLTD is trained offline with a model-based estimator and decomposes uncertainty (epistemic vs. aleatoric) to aid analysis of deferral triggers.

Sequential Outputs.

When outputs are sequences, whole-sequence deferral can be inefficient. Rayan and Tewari [65] propose two post-hoc rejectors for pre-trained predictors (two-stage; §4.2): (i) a *token-level* rejector that decides at step t whether to accept the model's next token or defer that token to an

expert with next-token capability; and (ii) a *one-time* rejector that selects a deferral point τ after which the expert completes the remaining suffix (for experts without token-level support). They provide convex, Bayes-consistent surrogates for both settings and show improved cost–accuracy trade-offs on sequence tasks (e.g., summarization) relative to whole-sequence deferral. One-time deferral also connects to multi-expert L2D by viewing each candidate τ as an expert that returns the suffix conditioned on the model's prefix.

6.5 Causality and Learning to Defer

Causal inference studies the effect of *interventions* (what would happen under a different action) rather than associations. A *confounder* is a variable that influences both the action (e.g., deferral or treatment) and the outcome; when such variables are unobserved, we have *hidden confounding*, which can bias estimates from observational data.

In L2D, causality enters in three ways: (i) *causal evaluation* of fixed deferral systems, (ii) *causal policy learning with deferral* from observational data under hidden confounding, and (iii) using L2D to arbitrate within *causal discovery* pipelines. We keep the predictor–rejector notation from §3.2: predictor h, expert $\mathcal{E}$, rejector r (or score s), and deferral cost $c(x, y)$.

Causal Evaluation of Deferral Systems.

Standard metrics (e.g., system accuracy) are associational and do not identify the *effect* of deferral. Palomba et al. [62] formalize evaluation with potential outcomes by viewing the deferral decision as the treatment. Let $T \in \{0, 1\}$ indicate the action, with $T = 0$ for autonomous prediction by h and $T = 1$ for deferral to the expert $\mathcal{E}$. Let Y be the correctness outcome (e.g., $Y = 1$ if the final decision is correct). The average treatment effect on deferred cases is

$$\tau_{\text{ATD}} = \mathbb{E}[Y(1) - Y(0) \,|\, T = 1] \,,$$

which measures, for cases actually deferred, the difference between their observed outcome under deferral and the counterfactual outcome had h handled them. When counterfactual predictions are systematically missing, the common score-threshold implementation of L2D enables a *regression discontinuity* design at the deferral cutoff to estimate a local causal effect. This isolates the impact of deferral independent of casemix.

Causal Policy Learning with Deferral.

Here the objective is an *interventional* policy with a deferral action learned from observational logs, where unobserved confounding may bias naive estimates. Ghoummaid and Shalit [25] propose **Causal Action Recommendation with Expert Deferral** which assumes that experts may act using information unavailable to the learner and model hidden confounding via a Marginal Sensitivity Model. They estimate bounds on conditional average potential outcomes for each action and construct a bound-aware, cost-sensitive objective in which the system compares pessimistic/optimistic outcome bounds across (i) acting with the model and (ii) deferring to $\mathcal{E}$. The surrogate thus replaces unknown ground-truth labels with causal bounds, yielding policies that defer in regions of large identification uncertainty and act otherwise. This is a "learning-to-defer-with-causal assumptions" setting, distinct from purely associational L2D.

L2D for Causal Discovery.

L2D can also arbitrate between heterogeneous *causal discovery* sources [73]. Clivio et al. [15] (L2D-CD) consider per-query decisions (e.g., edge existence or direction) between a fixed

data-driven method h and a knowledge-based expert $\mathcal{E}$ (including an LLM using metadata). With h and $\mathcal{E}$ fixed, the problem is two-stage (§4.2). A key simplification is a reduction to binary classification over the *disagreement set* $\{x : h(x) \neq \mathcal{E}(x)\}$: learn a classifier that predicts whether $\mathcal{E}$ is correct on those items and route accordingly. This applies standard L2D machinery to a causal *task*; it does not impose causal assumptions to learn the deferral rule itself.

7 Real-World Adaptations for L2D

The previous sections outlined the theoretical foundations and general formulations of L2D. Deploying these systems in practice, however, introduces additional challenges, since real environments rarely match the assumptions of theoretical models. This section reviews research that adapts L2D frameworks to function under such practical constraints and complexities, including *learning from limited expert data* (§7.1), *adapting to changing pools of experts* (§7.2), *integrating external policy rules such as fairness and budget limits* (§7.3), and *improving the safety of L2D systems* (§7.4).

Notation. Core notation follow §3.1 (Table 1). This section introduces operational quantities only when needed (e.g., base consultation costs β_j, per-expert capacity/budgets, fairness constraints, context sets $\mathcal{D}^E$ for new experts). All other notation remains unchanged.

7.1 Learning with Limited Expert Annotations

A major practical challenge for L2D systems is the high data volume requirement [36]. Standard formulations assume that expert predictions are available for every instance in the training set. This is often infeasible in settings where expert time is costly or where experts change frequently. To address this, a significant branch of research has focused on developing L2D frameworks that can function with a feasibly small amount of expert annotations. Broadly, these methods fall into two main strategies: those that aim to create a (pseudo-)complete dataset before training the L2D system, and those designed to learn directly from the sparse or incomplete data.

Completing the Dataset: Imputation and Active Learning.

The most direct strategy for handling missing labels is to fill them in, creating a complete dataset suitable for a standard L2D algorithm.

One approach is imputation, where a model of expert behavior is learned from a small seed set and then used to generate simulated expert labels. This was formalized by Hemmer et al. [28], who proposed a three-step process: (1) an embedding model is trained on ground-truth labels to create rich feature representations; (2) an "expertise predictor" is trained using semi-supervised methods on the available expert annotations to model the expert's behavior; and (3) this predictor generates pseudo-labels (simulated expert predictions) for the remaining data. This imputation strategy has been refined by subsequent work, such as Chen et al. [11], who introduced a novel consistency loss that encourages the expert model to learn the intrinsic structure of the data, improving the quality of the generated labels.

A contrasting approach is active learning, which intelligently queries the expert for the most informative real labels. This was explored by Charusaie et al. [9], who designed a scheme to efficiently learn the human's error boundary. Their *Disagreement on Disagreements* (DoD) algorithm iteratively identifies points where a committee of models disagrees on the expert's likely performance and queries the expert for those specific instances. This method significantly reduces the number of labels required to learn an effective deferral policy, but presupposes an interactive learning setup where the system can request specific annotations on demand.

Modeling from Incomplete Data: Sparsity-Aware and Probabilistic Approaches.

A second strategy is to design models that inherently handle data scarcity without requiring a fully annotated dataset.

A key example is sparsity-aware model design, which is particularly useful in multi-expert settings where annotations are sparse (e.g., "single-annotator sparsity", where each instance is typically annotated by only one expert). The **DeCCaF** framework by Alves et al. [1] directly addresses this by training a unified "Human Expertise Model" (HEM). By conditioning the model output on both the instance features and expert unique identity, it learns a single, shared representation of team behavior from these sparse annotations.

Other works have turned to more formal probabilistic models to handle missingness during the learning process. Nguyen et al. [59] frame the problem of missing expert annotations within a latent variable model, using the Expectation-Maximization (EM) algorithm to jointly infer the missing labels while simultaneously learning the L2D system. This provides a principled, end-to-end framework for training with incomplete data. In a different vein, the decoupled Bayesian approach of Strong et al. [67] avoids the need for expert annotations on the main training set entirely. It constructs an explicit, statistical representation of an expert's class-wise performance from a small, separate "context set." This expert model, which can be informed by priors, is then used directly at test-time to make deferral decisions, decoupling the expert modeling from the primary task training.

7.2 Handling Dynamic Expert Pools

A significant limitation of classic L2D frameworks is their assumption of a *fixed expert* pool. These systems are trained on data from a specific, well-identified set of expert(s), and their deferral policies are inherently customized to this exact group. This is often untenable in practice. For example, in a hospital, clinicians have different shift patterns, and subspecialists may be intermittently available. To address this, a line of research has focused on enabling L2D systems to adapt to previously unseen experts at test-time. The central idea shared by works such as Tailor et al. [68] and Strong et al. [67] is to enable this adaptation by leveraging a small amount of data from the new expert, known as a *context set*. This set provides a few-shot demonstration of the new expert's behavior. While both frameworks leverage this concept, they diverge fundamentally in how they represent expert behavior and structure their deferral mechanisms.

Expert Representation and Deferral

The key distinction between these approaches lies in how they model an expert from the context set. Tailor et al. [68] use a meta-learning framework with Neural Processes [22] to encode expert behavior into a learned implicit latent vector. The deferral mechanism is then additionally conditioned on this embedding, learning to associate different latent expert profiles with optimal deferral actions.

The alternative approach by Strong et al. [67] uses a Bayesian statistical model to construct an explicit, interpretable representation of expert performance. This representation consists of metrics, such as the estimated per-class accuracy and the associated uncertainty. This enables an *expert-agnostic* deferral mechanism, where the deferral logic is learned based on the general, structural relationship between predictor confidence and expert quantified competence (via the aforementioned metrics), rather than being tied to specific expert identities or profiles seen during training.

Implications and Trade-offs

These methods represent a step towards making L2D systems practical and deployable in non-static, real-world environments. The choice between these approaches introduces a key trade-off for system designers. Learned latent representations are useful for modeling complex, holistic behavioral patterns, but may be less interpretable and potentially less robust to out-of-distribution experts. In contrast, structured statistical representations can offer greater robustness in low-data regimes, interpretability, and data efficiency, but might provide a more generalized, and thus potentially less detailed representation of expert behavior. Future work in this area will likely focus on bridging this gap, seeking methods that combine the expressive power of latent models with the robustness and interpretability of structured ones.

7.3 Integrating Policy Constraints into Deferral Frameworks

Real-world L2D systems must often operate under policy constraints, such as fairness or budget limits, rather than only optimizing a single objective like accuracy. This has motivated a shift in L2D research from surrogate minimization to constrained optimization as described next.

A key contribution is the unifying post-processing framework of Charusaie and Samadi [10], which provides a general, theoretically-grounded method for finding a Bayes-optimal deferral policy that maximizes a primary objective while satisfying arbitrary constraints. Their two-stage methodology first uses standard models to estimate the necessary scores for the objective and constraints (e.g., model confidence, expert accuracy, fairness metrics). In the second stage, it leverages a generalization of the Neyman-Pearson lemma [35] to derive an optimal decision rule. This rule takes the form of a simple linear combination of the pre-computed scores: $\text{decision}(x) = \text{argmax}\left(\psi_0(x) - \sum_{i=1}^{m} k_i \psi_i(x)\right)$ Here, $\psi_0(x)$ represents the objective, $\psi_i(x)$ are costs for the m constraints, and the trade-off parameters k_i are tuned on a validation set to meet the desired constraint tolerances.

The significance of this work is its generality, which moves the field beyond designing task-specific surrogate losses. The authors also prove that finding a deterministic solution is NP-Hard, providing strong justification for their use of a tractable and provably optimal randomized policy.

7.4 Enhancing L2D Safety

The overall reliability of an L2D system hinges on the integrity of its core deferral mechanism. While much of the literature to date focuses on optimizing this mechanism under ideal data conditions, its performance can degrade significantly when faced with real-world complexities, compromising system safety and effectiveness. This has motivated a line of research aimed at enhancing the robustness of the deferral. These efforts address three primary failure modes: (1) the system's vulnerability to deliberate, adversarial manipulation; (2) its inability to express uncertainty when the optimal allocation is ambiguous; and (3) the lack of interpretability or rationale for deferral decisions needed for effective oversight.

Robustness Against Adversarial Attacks

The first study of adversarial robustness in L2D, by Montreuil et al. [50], reveals that systems are susceptible to attacks targeting the core *allocation mechanism*. Focusing on the practical two-stage setup (cf. §4.2) where only the rejector function r is learned, an adversary's goal is to apply a small perturbation to an input x to create x', manipulating the rejector's output $r(x')$ to cause a

suboptimal allocation. This can lead to consequences like increased costs or denial of service. The authors formalize two attack strategies:

1. **Untargeted Attacks:** The adversary crafts a perturbation to disrupt the *optimal allocation*, forcing the system to select *any* suboptimal agent and thereby degrading overall performance.
2. **Targeted Attacks:** The adversary forces the system to route a query to a *specific, predetermined* agent, which could enable fraud in a pay-per-query system.

A Robust Deferral Framework To defend against these threats, the same work proposes **SARD** (Smooth Adversarial Regularized Deferral), an adversarial training algorithm for the two-stage task. The approach learns a robust rejector r by minimizing the worst-case deferral loss within a small perturbation ball around each training input. This objective is the *adversarial true deferral loss*:

$$\tilde{\ell}_{\text{def}}(r, g, m, z) = \sup_{x' \in B_p(x,\gamma)} \left[\sum_{j=0}^{J} c_j(g(x), m_j(x), z) \cdot \mathbb{1}_{r(x')=j} \right] \tag{20}$$

A key subtlety is that the costs c_j are evaluated on the clean input x, while the rejector's decision is based on the perturbed input x'. As this loss is intractable, the SARD algorithm optimizes a smooth, regularized version of a tractable surrogate from a novel family of *adversarial margin deferral surrogates*. This framework has strong theoretical backing; the surrogates are proven to be both Bayes-consistent and ($\mathcal{R}$,$\mathcal{G}$)-consistent, providing a formal guarantee that the learned policy is robust against these allocation attacks.

Handling Uncertainty in the Deferral Decision

Beyond robustness to external attacks, the safety of an L2D system also depends on the reliability of its rejector. As a predictive model, the rejector can be wrong, especially on ambiguous inputs, making a forced allocation decision a safety risk.

To address this, Fang and Nalisnick [21] quantify rejector uncertainty using **conformal prediction** [2]. They construct a *deferral set* $\mathcal{S}(x)$ that can return $\{h\}$, $\{\mathcal{E}\}$, or $\{h, \mathcal{E}\}$ in the single-expert L2D setting (classifier h vs. expert $\mathcal{E}$). The set is calibrated (via split-conformal prediction) to marginally cover the indicator of expert correctness with level $1 - \alpha$, so that $\mathbb{P}\big(\mathbb{1}\{\mathcal{E}(x) = y\} \in \mathcal{S}(x)\big) \geq 1 - \alpha$ under the exchangeability assumption. When $\mathcal{S}(x) = \{h\}$ the system acts with h; when $\mathcal{S}(x) = \{\mathcal{E}\}$ it defers; and when $\mathcal{S}(x) = \{h, \mathcal{E}\}$ the rejector expresses uncertainty about the allocation.

This uncertainty signal supports safer downstream protocols when allocation is ambiguous:

1. **Abstention.** If $\mathcal{S}(x) = \{h, \mathcal{E}\}$, the system abstains; otherwise it follows the singleton in $\mathcal{S}(x)$ (act with h if $\{h\}$, defer if $\{\mathcal{E}\}$).
2. **Consensus prediction.** If $\mathcal{S}(x) = \{h, \mathcal{E}\}$, query both h and $\mathcal{E}$ and output a prediction only when they agree; otherwise abstain.

Interpretability in Deferral Decisions

Beyond robustness to attacks and uncertainty, a third pillar of L2D safety is interpretability. For a human-AI team to function effectively, it is often not enough for the human to know that a case has been deferred; they must also understand *why*. This can be important in high-stakes domains, as an explanation for deferral can build trust, reveal model blind spots, and guide the expert attention to the most challenging aspects of a problem. Addressing this, work has

focused on integrating L2D with inherently interpretable model architectures, most notably **Concept Bottleneck Models (CBMs)** [33].

CBMs are models designed to be transparent by forcing them to first predict a set of human-understandable intermediate "concepts" before making a final prediction. Pugnana et al. [64] introduce **Deferring Concept Bottleneck Models (DCBMs)**, a framework that equips CBMs with an L2D mechanism. The key innovation is that the system can learn to defer at two distinct levels: it can defer on the final task, or it can defer on one of the intermediate concepts that it finds ambiguous. This provides a granular, concept-level explanation for the deferral decision (e.g., "I am deferring this medical image case because I am uncertain about the 'Glandular' concept"). By modeling the problem as a composition of deferring systems, DCBMs can boost predictive performance while providing transparent reasons for human intervention, even when the human experts themselves are imperfect.

Similarly, the **DeCoDe** framework by He et al. [26] also leverages CBMs to create an interpretable, concept-driven deferral system. DeCoDe extends this framework by introducing a third collaborative mode beyond the standard AI or human only decision: a "complementarity" mode where AI and human inputs are fused for a joint decision. The choice between these three modes is governed by a gating network that operates on the concept representations, ensuring that the entire decision-making process (whether to predict, defer, or collaborate) remains transparent.

7.5 Controlling Workload Distribution to Experts

Table 4: Comparison of methods for controlling workload distribution to experts (L2D).

Approach (Citation)	Core Mechanism	Benefits	Limitations
Cost-Regularized Training Zhang et al. [76]	Additive penalty on collaboration cost (e.g., number of users engaged) in the training objective; a selector learns among modes (AI only, defer, or collaborate with J experts).	End-to-end, supports multi-user deferral/complementarity and trades off accuracy vs. human involvement.	No hard coverage/workload guarantees; tuning λ does not map directly to a target deferral rate; behavior depends on validation-time operating point.
MILP (Offline) Mozannar et al. [56]	Exact mixed-integer linear program that jointly optimizes a (linear) classifier and rejector with linear constraints (e.g., coverage).	Provably optimal for the training objective; straightforward to impose coverage and other linear constraints (e.g., fairness).	Computationally expensive; designed for linear (halfspace) models; typically used on modest n, d with precomputed features.
EM (Train-Time) Nguyen et al. [59]	Mixture-of-experts with constraints enforced in the E-step by bounding posterior assignment rates per expert (incl. AI).	Principled, integrates with minibatch deep learning, allows explicit per-expert workload budgets.	Requires specifying feasible per-expert lower/upper bounds; not an exact 0–1 optimizer like MILP.
Post-Hoc Calibration Ponomarev [63]	Calibrate a deferral score threshold on a held-out set to meet a required coverage, then apply per-sample at inference.	Fast, model-agnostic, and supports "on-the-fly" decisions without batching.	Heuristic with no formal guarantees; small budget violations can occur; performance depends on the calibration set.

Early L2D systems primarily maximized human-AI team accuracy, but real deployments often face bounded human capacity. Unconstrained L2D can overload a small subset of experts while underutilizing others, motivating methods that explicitly control workload distribution. Broadly, these either treat workload as a *soft penalty* learned end-to-end or impose a *hard budget* that must be met.

Cost-based approaches: workload as a soft penalty

A natural approach is to encode the *cost of collaboration* within the training loss. Zhang et al. [76] extend L2D to multiple users and joint complement/deferral (LECODU), defining a collaboration cost as the number of users engaged and optimizing $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \cdot \text{cost}(g_\phi(x))$, where $g_\phi(x)$ selects a collaboration mode (AI alone, defer, or collaborate with J experts). Varying λ smoothly trades accuracy for human effort, but there is no direct guarantee on coverage or per-expert workload, and the achieved operating point is typically chosen by validation. Related work has also noted that surrogate losses can be brittle in how well they complement the human, even without hard budgets, and may require careful tuning [56].

Budget-based approaches: workload as a hard constraint

When operations require strict capacity limits, budget-based methods are preferable. These fall into: (i) exact offline optimization, (ii) stochastic train-time optimization, and (iii) post-hoc calibration.

Exact offline optimization Mozannar et al. [56] show the linear L2D problem is computationally hard in general but give an exact mixed integer linear programming (MILP) solution for (linear) classifier and rejector, to which linear constraints are easily added. Let $r(x) \in \{0, 1\}$ denote deferral; a coverage constraint can be imposed as $\frac{1}{N}\sum_i r(x_i) \leq \beta$, i.e., AI covers at least $1 - \beta$ of inputs. On variables $d_i = \mathbf{1}\{r(x_i) = 1\}$ this is equivalent to $\frac{1}{N}\sum_i (1 - d_i) \geq C_r$, where C_r is the required AI coverage. The MILP yields globally optimal training solutions under the stated assumptions but is computationally heavier and targeted to halfspaces.

Stochastic train-time optimization via expectation–maximization (EM) Nguyen et al. [59] model L2D as a constrained mixture of multiple human experts plus the AI. Workload control is enforced during the E-step of EM by constraining posterior assignment rates $q(z_i)$ to lie within per-expert bounds $\boldsymbol{\varepsilon_\ell} \leq \frac{1}{N}\sum_i q(z_i) \leq \boldsymbol{\varepsilon_u}$. This integrates naturally with deep learning, supports missing annotations, and trains a gate that respects budgeted workloads without requiring post-hoc thresholding on the test set.

Post-hoc calibration heuristics. Ponomarev [63] propose a simple three-step recipe: (1) train any L2D model, (2) build a deferral score that ranks "AI vs. human," and (3) on a held-out set, pick a threshold that maximizes accuracy subject to the desired AI coverage C_r. At inference, defer iff the score falls below the calibrated threshold. This is lightweight and online, but lacks guarantees; empirical results show small but nonzero budget violations in some settings.

Discussion

Method choice hinges on operational needs (Table 4). Soft-penalty methods (e.g., 76) are attractive for multi-expert collaboration but provide no strict budget guarantees. MILP [56] offers exact control and provable optimality (in the stated linear setting), at higher computation. EM-based training [59] brings explicit workload budgets into end-to-end learning and avoids post-hoc dependence on test batches. Post-hoc calibration [63] is the most deployment-friendly when you only need an operating point, accepting approximate budget satisfaction. A promising direction is to combine EM-style train-time constraints with calibrated post-hoc tuning, or to pair MILP-style budget guarantees with richer (non-linear) models via learned surrogates.

8 Future Directions, Open Challenges and Concluding Remarks

The theory for L2D is well developed, but (to the best of our knowledge) there are no published real-world case studies or deployments. Because L2D is a systems problem (routing among models and experts under budgets, latency, and fairness), a priority should be operational validation: rigorously designed studies that test L2D end-to-end under real constraints. This complements ongoing theoretical work and establishes whether and when L2D delivers practical benefit. These studies will also surface real failure modes (e.g., distribution shift, expert drift, budget mis-specification, miscalibrated expert-correctness, handoff/latency bottlenecks), strongly motivating the next generation of L2D theory and methods.

Targeted agenda by taxonomy.

- **Methodological frameworks (§4).** Default to *two-stage* when the predictor is frozen/API; train the rejector from weak/bandit feedback—no logits or dense expert labels required (§4.2); mitigate one-stage brittleness with "anti-specialization" regularizers and advisor-compatible heads that preserve competence on deferred regions.
- **Optimization & theory (§5).** Make guarantees *shift-robust* (covariate/label/expert-population) while preserving calibration, budgets, and fairness (§5.2). Relax strong separability/modeling assumptions in multi-expert two-stage deferral (§5.3) (e.g., unique perfect expert or ≤ 2 correct experts), and develop surrogates/analyses that remain valid with overlapping experts, richer cost structures, and non-unique optima. Develop guarantees under *model–expert dependence* (§5.4): prove consistency/Bayes-consistency and calibration for dependence-aware surrogates (e.g., DCE), characterize identifiability and sample complexity for estimating joint events, and extend to multi-expert/two-stage, budgeted, and fairness-constrained settings with robustness to misspecification and selection bias.
- **Task generalizations (§6).** Develop realistic task generalizations that enable real-world deployments of L2D in healthcare, finance, and other decision-critical settings.
- **Real-world adaptations (§7).** Model the dynamics of human behavior and the operational context in which L2D operates, and design frameworks that adapt to these complexities.

Conclusion

L2D reframes reliability from *confidence estimation* to *decision allocation*: deciding who should act, for which input, under what costs and constraints. The field now has clear frameworks, robust objectives with meaningful guarantees, extensions beyond multiclass prediction, and an emerging toolkit for deployment under real-world constraints. We hope the taxonomy, synthesis, and guidance in this survey help researchers target the most impactful gaps and help practitioners build human-AI systems that are not only accurate, but *calibrated, accountable, and safe*.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council. JS is funded by the EPSRC Center for Doctoral Training in Health Data Science [EP/S02428X/1]. ES is funded by an EPSRC Doctoral Training Partnership [EP/W524311/1]. HR and AN acknowledge EPSRC Turing AI Fellowship: Ultra Sound Multi-Modal Video-based Human-Machine Collaboration [EP/X040186/1].

Contributions

JS conceived the survey, defined its scope and taxonomy, and led the writing of the manuscript. ES contributed substantially through literature curation, drafting and revising sections of the manuscript, and designed and produced the figures. HR contributed through review and editing of the manuscript. HH and AN provided supervision, domain guidance, and critical revisions. All authors approved the final version of the manuscript.

References

[1] Jean Vieira Alves, Diogo Leitão, Sérgio Jesus, Marco O. P. Sampaio, Javier Liébana, Pedro Saleiro, Mario A. T. Figueiredo, and Pedro Bizarro. 2024. Cost-Sensitive Learning to Defer to Multiple Experts with Workload Constraints. *Transactions on Machine Learning Research* (2024). https://openreview.net/forum?id=TAvGZm2Rqb

[2] Anastasios N Angelopoulos and Stephen Bates. 2021. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511* (2021).

[3] Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2022. H-Consistency Bounds for Surrogate Loss Minimizers. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 1117–1174. https://proceedings.mlr.press/v162/awasthi22c.html

[4] Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2022. Multi-Class *H*-Consistency Bounds. *Advances in neural information processing systems* 35 (2022), 782–795.

[5] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. 2021. Is the most accurate ai the best teammate? Optimizing AI for teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11405–11414.

[6] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S Weld, Walter S Lasecki, and Eric Horvitz. 2019. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 2429–2437.

[7] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. 2006. Convexity, classification, and risk bounds. *J. Amer. Statist. Assoc.* 101, 473 (2006), 138–156.

[8] Yuzhou Cao, Hussein Mozannar, Lei Feng, Hongxin Wei, and Bo An. 2023. In defense of softmax parametrization for calibrated and consistent learning to defer. *Advances in Neural Information Processing Systems* 36 (2023), 38485–38503.

[9] Mohammad-Amin Charusaie, Hussein Mozannar, David Sontag, and Samira Samadi. 2022. Sample efficient learning of predictors that complement humans. In *International Conference on Machine Learning*. PMLR, 2972–3005.

[10] Mohammad-Amin Charusaie and Samira Samadi. 2024. A unifying post-processing framework for multi-objective learn-to-defer problems. *Advances in Neural Information Processing Systems* 37 (2024), 23705–23755.

[11] Haoqing Chen, Bo Jin, and Xiangfeng Wang. 2024. Semi-supervised Learning to Defer Algorithm for Lung Disease Diagnosis. In *2024 IEEE International Conference on Big Data (BigData)*. 4474–4481. doi:10.1109/BigData62323.2024.10825864

[12] Xin Cheng, Yuzhou Cao, Haobo Wang, Hongxin Wei, Bo An, and Lei Feng. 2023. Regression with cost-based rejection. *Advances in Neural Information Processing Systems* 36 (2023), 45172–45196.

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

[14] C. K. Chow. 1957. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers* EC-6, 4 (1957), 247–254. doi:10.1109/TEC.1957.5222035

[15] Oscar Clivio, Divyat Mahajan, Perouz Taslakian, Sara Magliacane, Ioannis Mitliagkas, Valentina Zantedeschi, and Alexandre Drouin. 2025. Learning to Defer for Causal Discovery with Imperfect Experts. *arXiv preprint arXiv:2502.13132* (2025).

[16] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. 2016. Learning with rejection. https://cs.nyu.edu/~mohri/pub/rej.pdf

[17] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (Sept. 1995), 273–297. doi:10.1007/bf00994018

[18] Giulia DeSalvo, Mehryar Mohri, and Umar Syed. 2015. Learning with deep cascades. In *International Conference on Algorithmic Learning Theory*. Springer, 254–269.

[19] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. Springer, 1–15.

[20] Ran El-Yaniv et al. 2010. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research* 11, 5 (2010).

[21] Yizirui Fang and Eric Nalisnick. 2024. Learning to Defer with an Uncertain Rejector via Conformal Prediction. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*. https://openreview.net/forum?id=TWb9y4PNSW

[22] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. 2018. Neural processes. *arXiv preprint arXiv:1807.01622* (2018).

[23] Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in neural information processing systems* 30 (2017).

[24] Yonatan Geifman and Ran El-Yaniv. 2019. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*. PMLR, 2151–2159.

[25] Marah Ghoummaid and Uri Shalit. 2024. When to act and when to ask: policy learning with deferral under hidden confounding. *Advances in Neural Information Processing Systems* 37 (2024), 56108–56135.

[26] Chengbo He, Bochao Zou, Junliang Xing, Jiansheng Chen, Yuanchun Shi, and Huimin Ma. 2025. DeCoDe: Defer-and-Complement Decision-Making via Decoupled Concept Bottleneck Models. *arXiv preprint arXiv:2505.19220* (2025).

[27] Patrick Hemmer, Sebastian Schellhammer, Michael Vössing, Johannes Jakubik, and Gerhard Satzger. 2022. Forming Effective Human-AI Teams: Building Machine Learning Models that Complement the Capabilities of Multiple Experts. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2478–2484. doi:10.24963/ijcai.2022/344 Main Track.

[28] Patrick Hemmer, Lukas Thede, Michael Vössing, Johannes Jakubik, and Niklas Kühl. 2023. Learning to defer with limited expert predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 6002–6011.

[29] Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. 2024. Machine learning with a reject option: A survey. *Machine Learning* 113, 5 (2024), 3073–3110.

[30] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive Mixtures of Local Experts. *Neural Computation* 3, 1 (1991), 79–87. doi:10.1162/neco.1991.3.1.79

[31] Michael I Jordan and Robert A Jacobs. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural computation* 6, 2 (1994), 181–214.

[32] Shalmali Joshi, Sonali Parbhoo, and Finale Doshi-Velez. 2023. Learning-to-defer for sequential medical decision-making under uncertainty. *Transactions on Machine Learning Research* (2023). https://openreview.net/forum?id=0pn3KnbH5F

[33] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International conference on machine learning*. PMLR, 5338–5348.

[34] Maksim Lapin, Matthias Hein, and Bernt Schiele. 2017. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. *IEEE transactions on pattern analysis and machine intelligence* 40, 7 (2017), 1533–1554.

[35] Erich Leo Lehmann and Joseph P Romano. 2005. *Testing statistical hypotheses*. Springer.

[36] Diogo Leitão, Pedro Saleiro, Mário A. T. Figueiredo, and Pedro Bizarro. 2022. Human-AI Collaboration in Decision-Making: Beyond Learning to Defer. arXiv:2206.13202 [cs.LG] https://arxiv.org/abs/2206.13202

[37] Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. 2024. A survey on deep active learning: Recent advances and new frontiers. *IEEE Transactions on Neural Networks and Learning Systems* 36, 4 (2024), 5879–5899.

[38] Shuqi Liu, Yuzhou Cao, Qiaozhen Zhang, Lei Feng, and Bo An. 2024. Mitigating underfitting in learning to defer with consistent losses. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4816–4824.

[39] Phil Long and Rocco Servedio. 2013. Consistency versus Realizable H-Consistency for Multiclass Classification. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 28)*, Sanjoy Dasgupta and David McAllester (Eds.). PMLR, Atlanta, Georgia, USA, 801–809. https://proceedings.mlr.press/v28/long13.html

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

[41] Palak Mahajan, Shahadat Uddin, Farshid Hajati, and Mohammad Ali Moni. 2023. Ensemble learning for disease prediction: A review. In *Healthcare*, Vol. 11. MDPI, 1808.

[42] Anqi Mao. 2025. *Theory and Algorithms for Learning with Multi-Class Abstention and Multi-Expert Deferral*. Ph. D. Dissertation. New York University. https://www.proquest.com/dissertations-theses/theory-algorithms-learning-with-multi-class/docview/3172888387/se-2

[43] Anqi Mao, Christopher Mohri, Mehryar Mohri, and Yutao Zhong. 2023. Two-stage learning to defer with multiple experts. *Advances in neural information processing systems* 36 (2023), 3578–3606.

[44] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on Machine learning*. PMLR, 23803–23828.

[45] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2024. Predictor-rejector multi-class abstention: Theoretical analysis and algorithms. In *International Conference on Algorithmic Learning Theory*. PMLR, 822–867.

[46] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2024. Principled approaches for learning to defer with multiple experts. In *International Workshop on Combinatorial Image Analysis*. Springer, 107–135.

[47] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2024. Realizable H-Consistent and Bayes-Consistent Loss Functions for Learning to Defer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=0cO2XakUUK

[48] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2024. Regression with Multi-Expert Deferral. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 34738–34759. https://proceedings.mlr.press/v235/mao24d.html

[49] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2025. Mastering Multiple-Expert Routing: Realizable H-Consistency and Strong Guarantees for Learning to Defer. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=2KlxjR6lsd

[50] Yannis Montreuil, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. 2025. Adversarial Robustness in Two-Stage Learning-to-Defer: Algorithms and Guarantees. *arXiv preprint arXiv:2502.01027* (2025).

[51] Yannis Montreuil, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. 2025. One-Stage Top-k Learning-to-Defer: Score-Based Surrogates with Theoretical Guarantees. *arXiv preprint arXiv:2505.10160* (2025).

[52] Yannis Montreuil, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. 2025. Why Ask One When You Can Ask k? Two-Stage Learning-to-Defer to the Top-k Experts. *arXiv preprint arXiv:2504.12988* (2025).

[53] Yannis Montreuil, Shu Heng Yeo, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. 2024. Two-stage Learning-to-Defer for Multi-Task Learning. *arXiv preprint arXiv:2410.15729* (2024).

[54] Yannis Montreuil, Shu Heng Yeo, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. 2025. Optimal Query Allocation in Extractive QA with LLMs: A Learning-to-Defer Framework with Theoretical Guarantees. arXiv:2410.15761 [cs.CL] https://arxiv.org/abs/2410.15761

[55] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. 2023. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review* 56, 4 (2023), 3005–3054.

[56] Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David Sontag. 2023. Who should predict? exact algorithms for learning to defer to humans. In *International conference on artificial intelligence and statistics*. PMLR, 10520–10545.

[57] Hussein Mozannar and David Sontag. 2020. Consistent estimators for learning to defer to an expert. In *International conference on machine learning*. PMLR, 7076–7087.

[58] Harikrishna Narasimhan, Wittawat Jitkrittum, Aditya K Menon, Ankit Rawat, and Sanjiv Kumar. 2022. Post-hoc estimators for learning to defer to an expert. *Advances in Neural Information Processing Systems* 35 (2022), 29292–29304.

[59] Cuong C Nguyen, Thanh-Toan Do, and Gustavo Carneiro. 2025. Probabilistic learning to defer: Handling missing expert annotations and controlling workload distribution. In *The Thirteenth International Conference on Learning Representations*.

[60] Chenri Ni, Nontawat Charoenphakdee, Junya Honda, and Masashi Sugiyama. 2019. On the calibration of multiclass classification with rejection. *Advances in neural information processing systems* 32 (2019).

[61] Nastaran Okati, Abir De, and Manuel Rodriguez. 2021. Differentiable learning under triage. *Advances in Neural Information Processing Systems* 34 (2021), 9140–9151.

[62] Filippo Palomba, Andrea Pugnana, Jose Manuel Alvarez, and Salvatore Ruggieri. 2024. A causal framework for evaluating deferring systems. *arXiv preprint arXiv:2405.18902* (2024).

[63] Andrew Ponomarev. 2024. A Simple Heuristic for Controlling Human Workload in Learning to Defer. In *International Conference on Pattern Recognition*. Springer, 120–130.

[64] Andrea Pugnana, Riccardo Massidda, Francesco Giannini, Pietro Barbiero, Mateo Espinosa Zarlenga, Roberto Pellungrini, Gabriele Dominici, Fosca Giannotti, and Davide Bacciu. 2025. Deferring Concept Bottleneck Models: Learning to Defer Interventions to Inaccurate Experts. *arXiv preprint arXiv:2503.16199* (2025).

[65] Sahana Rayan and Ambuj Tewari. 2025. Learning to Partially Defer for Sequences. *arXiv preprint arXiv:2502.01459* (2025).

[66] Pouria Salehi, Erin K. Chiou, Michelle Mancenido, Ahmadreza Mosallanezhad, Myke C. Cohen, and Aksheshkumar Shah. 2021. Decision Deferral in a Human-AI Joint Face-Matching Task: Effects on Human Performance and Trust. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 65, 1 (2021), 638–642.

[67] Joshua Strong, Pramit Saha, Yasin Ibrahim, Cheng Ouyang, and Alison Noble. 2025. Expert-Agnostic Learning to Defer. *arXiv preprint arXiv:2502.10533* (2025).

[68] Dharmesh Tailor, Aditya Patra, Rajeev Verma, Putra Manggala, and Eric Nalisnick. 2024. Learning to defer to a population: A meta-learning approach. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3475–3483.

[69] Rajeev Verma, Daniel Barrejón, and Eric Nalisnick. 2023. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 11415–11434.

[70] Rajeev Verma and Eric Nalisnick. 2022. Calibrated learning to defer with one-vs-all classifiers. In *International Conference on Machine Learning*. PMLR, 22184–22202.

[71] Zixi Wei, Yuzhou Cao, and Lei Feng. 2024. Exploiting Human-AI Dependence for Learning to Defer. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 52484–52499. https://proceedings.mlr.press/v235/wei24a.html

[72] Forest Yang and Sanmi Koyejo. 2020. On the consistency of top-k surrogate losses. In *International Conference on Machine Learning*. PMLR, 10727–10735.

[73] Alessio Zanga, Elif Ozkirimli, and Fabio Stella. 2022. A survey on causal discovery: theory and practice. *International Journal of Approximate Reasoning* 151 (2022), 101–129.

[74] Tong Zhang. 2004. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research* 5, Oct (2004), 1225–1251.

[75] Xu-Yao Zhang, Guo-Sen Xie, Xiuli Li, Tao Mei, and Cheng-Lin Liu. 2023. A survey on learning to reject. *Proc. IEEE* 111, 2 (2023), 185–215.

[76] Zheng Zhang, Wenjie Ai, Kevin Wells, David Rosewarne, Thanh-Toan Do, and Gustavo Carneiro. 2024. Learning to complement and to defer to multiple users. In *European Conference on Computer Vision*. Springer, 144–162.