# CAREFUL JUDGE: SAFE AND EFFICIENT HUMAN–AI COLLABORATIVE DECISION MAKING

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

## ABSTRACT

In human–AI collaborative decision making, human review can prevent unsafe AI decisions, but each human judgment is costly. Treating human intervention as a one-off fallback misses the opportunity to improve future AI decisions, yet adaptively learning from human feedback changes the model and can invalidate previously calibrated safety guardrails. We approach this challenge with CARE—calibrated adaptive rectification and escalation—an end-to-end pipeline that combines AI models and human reviewers to guarantee safe, human-aligned decisions, while continuously learning from human feedback to achieve greater automation with fewer human queries. CARE is principled, general, modular, and works with any black-box AI model. We equip it with a rectification module that improves AI decisions by learning the structure of human–AI misalignment. Our novel adaptive calibration module guarantees risk control at every time step for any rectification module. Experiments on four safety-critical real-world datasets spanning language, vision, driving, and robotics demonstrate that CARE achieves human-aligned decisions while reducing human queries by 25–81% relative to baselines.

## 1 INTRODUCTION

Human–AI collaboration is increasingly used in safety-critical applications, including data triage and evaluation (Hemmer et al., 2025; Mozannar & Sontag, 2020), autonomous driving (Luo et al., 2025; Sinha et al., 2024), robot monitoring (Sermanet et al., 2025), and model alignment (Christiano et al., 2017; Lee et al., 2024; Bai et al., 2022). In these settings, AI predictions represent an inexpensive, but potentially misaligned, way to scale reliable, but costly, human judgment. Deciding when to trust AI and when to call for human review creates a safety–efficiency trade-off (El-Yaniv & Wiener, 2010).

Existing methods address different parts of this trade-off. Some provide safety guarantees but may require frequent human intervention (Jung et al., 2025; Ren et al., 2023); others learn from human feedback to improve efficiency without providing such guarantees (Angelopoulos et al., 2023; Christiano et al., 2017). Naively combining the two directions may amplify their respective weaknesses and worsen the trade-off—splitting already scarce human feedback between model improvement and safety checks could make the former noisier and the latter more conservative, as is evident in Section 7. Still, we ask for the best of both worlds: *how can a human–AI system continue to make safe decisions while learning from human feedback to reduce future intervention?*

A positive answer requires true collaboration: a human provides feedback when AI is uncertain, and AI becomes more confident when facing similar scenarios in the future. Such dynamic interaction should uncover the structure of human–AI misalignment and close this gap, making AI decisions safer and human intervention less frequent. However, this structure may be difficult to learn when human review is sparse and selective. Moreover, safety guarantees must keep pace with each model update. Specifically, queries selectively reviewed by a human can differ statistically from future ones, and earlier safety checks may no longer be valid after model updates, so calibration must efficiently track shifts in both the data distribution and the model.

These challenges call for model-agnostic risk control for every decision to be made, and model-aware adaptive learning after every decision is made. Model awareness enables learning *efficiency*, while model agnosticism ensures *safety* in the presence of model misspecification. Driven by these two objectives, we propose *calibrated adaptive rectification and escalation* (CARE), a principled, general, and modular framework that combines AI models and human reviewers into an ensemble



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

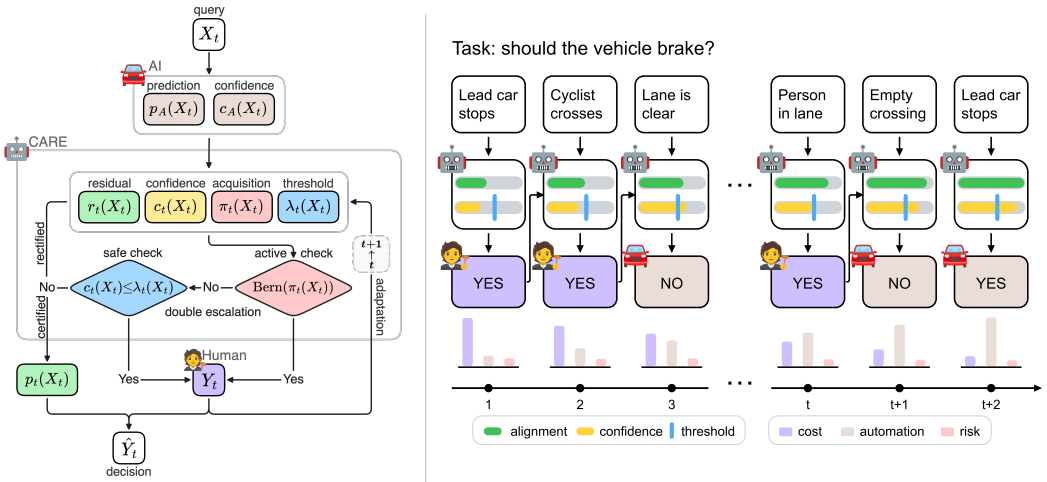



Figure 1: **(left)** Per-sample decision pipeline. Upon a query, CARE rectifies AI's prediction and confidence by correcting the human–AI residual and reducing uncertainty, based on past human feedback. The rectified quantities are then passed through two checks: an active acquisition rule that escalates informative queries, and a confidence gate that ensures the risk of the final human–AI decision is below the tolerance. Once a query is escalated, the human's feedback is used as the final decision and to update CARE's modules. **(right)** Adaptive learning from human feedback. CARE learns to adapt to the revealed human preference over time and across scenarios, continuously increasing alignment and reducing the uncertainty of the rectified AI model, and thus improving efficiency.

decision-maker. CARE guarantees safe, human-aligned decisions and continuously learns from human feedback to achieve greater automation while reducing human queries.

We instantiate CARE in binary classification, using an exact risk decomposition (Section 2) to motivate its four modules: (Section 3) *Residual Gauss* rectifies AI predictions and uncertainty estimates by fitting a Gaussian process to the residual between human and AI decisions; (Section 4) *Confident Jack* calibrates the rectified model with statistical and computational efficiency; (Section 5) *Exacting Gates* escalates queries to a human using a scenario-specific threshold informed by calibration; and (Section 6) *Active Max* adaptively selects informative human queries. All modules are designed for flexible offline–online learning schedules and share two objectives: safety and efficiency. An overview of CARE is provided in Figure 1.

Specifically, our contributions are:

- We formalize human–AI collaboration as a general statistical decision problem with a flexible offline–online schedule, covering different scenarios and model lifecycles.
- We derive an exact conditional risk decomposition into four sources, each motivating a distinct component of CARE—an end-to-end human–AI collaboration pipeline.
- We develop a correlation-aware Bayesian residual rectifier that combines a learned kernel with AI-imputed uncertainty. It propagates human feedback across similar scenarios, enabling faster adaptation and reducing future escalation frequency.
- We develop adaptive weighted confidence-based cross-conformal calibration that reuses labels collected for learning and accommodates arbitrary model updates under history-dependent acquisition. It provides model-agnostic marginal risk control at every time step.
- We evaluate CARE on four safety-critical real-world datasets spanning language, vision, driving, and robotics, demonstrating human-aligned decisions with substantially fewer human queries.

### 1.1 Related Work

**AI-accelerated human feedback.** As AI models are applied to increasingly complex tasks, high-quality supervision is a key bottleneck. To alleviate this, some works consider using learned models to scale supervision. Reinforcement learning from human feedback (RLHF) distills human preference data into a reward model (Christiano et al., 2017), while other techniques bring in human feedback at

a higher level, in rubric or constitution design (Bai et al., 2022; Lee et al., 2024; Zheng et al., 2023). Such techniques have been instrumental in advancing open-world physical AI, with LLM/VLM rewards scoring decision-making in robotics and autonomous vehicles (Ma et al., 2024; Sima et al., 2023; Huang et al., 2024; Sermanet et al., 2025; Rocamonde et al., 2024). However, ensuring alignment between the AI feedback and what a human expert would provide remains critical, especially in safety-critical settings. Recent literature applies ideas from selective classification and abstention (El-Yaniv & Wiener, 2010; Vardhan et al., 2026; Vishwakarma et al., 2024) to applications of AI feedback, allowing AI to escalate hard cases to human experts (Jung et al., 2025). We extend these ideas to allow for online adaptation, improving efficiency while maintaining alignment guarantees.

**Model-assisted inference.** A related line of research explores combining cheap proxy measurements and limited expensive measurements in the context of variance-reduced statistical inference, where the cheap proxy can be a prediction model (Angelopoulos et al., 2023; Zrnic & Candès, 2026) or a simulator (Luo et al., 2025; 2026; Badithela et al., 2025). Angelopoulos et al. (2023) propose prediction-powered inference (PPI) that uses empirical risk minimization to correct the model's systematic bias. While its fitted rectifier can capture some local structure, we model potential residual correlations explicitly and update each prediction's confidence through its posterior. PPI instead provides a global confidence bound for its bias correction. Greater model capacity can reduce approximation error but may increase estimation variability, motivating prior refitting as human feedback accumulates.

**Conformal calibration.** Sharing the goal of model-agnostic risk control, cross-conformal prediction (Vovk, 2012) and Jackknife+ (Barber et al., 2021) improve data efficiency over split conformal prediction (Papadopoulos et al., 2002; Lei et al., 2018; Jung et al., 2025), which splits the available human feedback into disjoint sets for training and calibration. To correct known or tractable covariate shifts, Tibshirani et al. (2020) introduce *weighted exchangeability*, and Prinster et al. (2022) apply it to obtain weighted Jackknife+. These methods do not directly handle the history-dependent covariate shifts induced by active acquisition in our setting, especially during the online period, when the weight calculation is intractable. Online model adaptation adds another layer of distribution shift. Existing online conformal methods (Gibbs & Candès, 2021; 2022; Bhatnagar et al., 2023; Feldman et al., 2023) often treat distribution shifts as adversarial and focus on time-averaged regret or coverage. Our goal is risk control at every time step without conservative fallbacks: calibration should track model improvement and reduce escalation as the model learns.

### 1.2 Problem setup

We instantiate human–AI collaborative decision making in binary classification. Consider an unlabeled dataset $(X_t)_{t\geq 1}$ from the feature space $\mathcal{X}$, with corresponding human oracle labels $(Y_t)_{t\geq 1}$ from the label space $\mathcal{Y} = \{0, 1\}$, revealed upon query escalation; we write $Z_t = (X_t, Y_t)$ for the full data. We assume the dataset is drawn from an i.i.d. stream and the human oracle provides correct labels. The inexpensive AI model provides a probabilistic classifier with an attached *confidence* score: $(p_A, c_A) : \mathcal{X} \mapsto [0, 1]^2$, which is not assumed to be accurate or calibrated.

Given these resources, we consider an *end-to-end* task: a framework must use them for all required stages, including but not limited to training, validation, calibration, and deployment. Any labels needed at these stages must be obtained through human queries. We consider two metrics: *risk* and *cost*. For a loss function $L$, e.g., zero–one loss $L_{01} : \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}, (\hat{y}, y) \mapsto \mathbf{1}\{\hat{y} \neq y\}$ and Brier score $L_2 : [0, 1]^2 \to \mathbb{R}_{\geq 0}, (\hat{p}, y) \mapsto (\hat{p} - y)^2$, and a classifier $f$, the risk is $R_L(f) = \mathbb{E}[L(f(X), Y)]$. Our default risk is the zero–one risk, i.e., the misclassification probability $R_{01}(f) = P(f(X) \neq Y)$. Cost is measured by the human query complexity, i.e., the end-to-end total number of human queries. We say a classifier is *safe* if its risk is below a user-specified tolerance $\epsilon \in (0, 1)$. We say an algorithm is more (cost-)*efficient* than another if it has lower human query complexity. Distinguish this from computational efficiency. Our goal is to maximize efficiency subject to the safety constraint.

## 2 Careful Judge: A framework for risk-controlled adaptive AI-assisted decision making

We motivate our method by examining the risk of an AI decision.

**Proposition 1** (Risk decomposition; informal)**.** *The AI model's Brier risk has an exact decomposition:*

$$\begin{aligned} R_{2,x}(p_A) =& \mathbb{E}_Y[(p_A(X) - Y)^2 \mid X = x] \\ =& \underset{\updownarrow}{\textit{aleatoric ambiguity}} + \underset{\updownarrow}{\textit{excess risk}} + \underset{\updownarrow}{\textit{epistemic uncertainty}} + \underset{\updownarrow}{\textit{systematic bias}}. \\ & \quad \textit{escalation} \qquad \textit{calibration} \qquad \textit{adaptation} \qquad \textit{rectification} \end{aligned}$$

The formal statement and proof are given in Appendix B. Proposition 1 identifies four sources of risk to be addressed to align the AI model with human preferences and deploy it safely:

- Aleatoric ambiguity encodes the intrinsic hardness of the query, which is the irreducible risk of the Bayes-optimal human-aligned predictor. When this risk exceeds the tolerance, we *escalate* the query to the human oracle to reveal the correct zero-loss decision.
- Excess risk measures the gap between a learned predictor and the Bayes-optimal predictor due to finite data and model misspecification. We *calibrate* the escalation rule to control the ensemble's risk despite imperfect learning.
- Epistemic uncertainty captures training-sample-specific information, including the correlation structure revealed by the queried data. We *adapt* to this structure and prioritize informative queries that can improve future decisions.
- Systematic bias is the population-level misalignment of the AI model. We *rectify* AI predictions by learning to correct this bias from human feedback.

This motivates calibrated adaptive rectification and escalation (CARE), a pipeline that reuses every queried human label to adaptively rectify AI's systematic bias and epistemic uncertainty, calibrate the rectified AI model, and escalate to human when the rectified AI decision is still risky. This pipeline improves efficiency while controlling the risk of the resulting ensemble. Formally, human queries inform a rectifier $r$ in the *score* space (the range of a strictly increasing link function $g\colon [0,1] \to \bar{\mathbb{R}}$), an uncertainty-reduced confidence measure $c$, a calibrated escalation threshold $\lambda$, and an active acquisition function $\pi$. At time step $t$, these quantities assemble into a probabilistic classifier:

$$\begin{cases} p_t(X_t) \coloneqq g^{-1}(g(p_A(X_t)) + r_t(X_t)), & \text{if } c_t(X_t) > \lambda_t(X_t) \text{ and } 0 = A_t \sim \mathrm{Bern}(\pi_t(X_t)); \\ \mathbf{1}\{Y_t = 1\}, & \text{otherwise.} \end{cases} \tag{1}$$

In words, the rectified AI decision is accepted only when its confidence exceeds the calibrated threshold and the input is not selected for active acquisition. Otherwise, the human oracle supplies the final decision, and the queried label becomes available for subsequent learning and calibration. Figure 1 illustrates this decision flow.

CARE is modular, as one can plug in any rectifier $r$, confidence measure $c$, escalation threshold $\lambda$, and acquisition function $\pi$ into the pipeline. Nonetheless, in the next four sections, we develop these four modules jointly to pursue:

- *Safety.* Adaptive rectification continuously improves AI predictions, while adaptive calibration and escalation provide model-agnostic risk control as the model evolves.
- *Efficiency.* We reuse each human query to update rectifier, quantify uncertainty, and calibrate the escalation threshold, maximizing query efficiency. Active acquisition and adaptive learning aim to reduce future escalation frequency.

*Remark* 1. Proposition 1 offers a triage of gaps left by existing tools. Classical selective classification (El-Yaniv & Wiener, 2010) and calibrated threshold-based escalation (Jung et al., 2025) do not learn to improve the base model, so a biased or uncertain model may escalate every query under a small risk tolerance, leading to huge cost. Population-level rectifier regression (Angelopoulos et al., 2023) targets systematic bias, but its uncertainty quantification does not adapt to locally revealed correlation structure. Finally, active acquisition (Zrnic & Candès, 2026) can target epistemically informative points, but does not by itself learn a rectifier or an acquisition function, nor does it certify the risk of accepted decisions. CARE closes all four gaps to achieve safety efficiently.

Furthermore, we design CARE to support flexible offline–online model lifecycles. At offline checkpoints, CARE revisits accumulated human feedback in an order-invariant manner, refits rectification and rebuilds calibration, prioritizing query efficiency by making fuller use of queried labels. Between checkpoints, CARE processes samples sequentially and updates rectification and calibration online.

## 3 RESIDUAL GAUSS: LOCALITY-AWARE BAYESIAN RECTIFICATION

To close reducible gaps in Proposition 1, we use queries to learn the human–AI residual and rectify AI's decision to align with human preference. AI's uncertainty about human preference can be correlated across the input space. For example, an autonomous driving system that misjudges a pedestrian's intention at one intersection may make similar errors at similar intersections (Sinha et al., 2024). These correlations allow feedback on one query to improve decisions on related inputs, improving alignment—and thereby reducing escalation frequency—over time. On the other hand, treating residuals as independent can lead to a query-all degradation when the initial uncertainty level is above the risk tolerance.

Therefore, Proposition 1 suggests a learning model to capture both the global systematic bias and local epistemic uncertainty of the human–AI residual. We model this with a Gaussian process (GP) (Rasmussen & Williams, 2008). On the score scale of the link $g$, we place a $\mathrm{GP}(m, k)$ prior on the residual $r(x) = g(p_H(x)) - g(p_A(x))$, where $p_H$ is the Bayes-optimal human-aligned predictor, and use human feedback to obtain the Bayesian rectifier $r_t$. The mean $m$ captures global systematic bias, while the kernel $k$ captures correlations that let feedback transfer across inputs.

**Initialization: parametric prior regression.** AI's raw confidence $c_A$ can offer useful information about uncertainty, even when it is not calibrated. We incorporate it into a learnable GP prior:

$$r \sim \mathrm{GP}(m_\beta, k_{\tau,\rho,\phi}), \quad \text{with} \quad m_\beta(x) = x^T\beta, \quad k_{\tau,\rho,\phi}(x,x') = \tau^2(\rho k_\phi(x,x') + (1-\rho)k_A(x,x')),$$

where the linear mean captures systematic bias, parameterized kernel $k_\phi$ captures residual correlations, and $k_A$ encodes the AI's uncertainty, such as an ensemble's empirical covariance or a diagonal kernel based on $c_A$.[1] When $\tau \to 0$, our model reduces to the parametric linear regression model in Angelopoulos et al. (2023). We refit the prior parameters at offline checkpoints, using, e.g., maximum likelihood estimation and expectation maximization. This adaptive prior lets human feedback determine how much to trust the AI's confidence and how much additional residual structure to learn.

**Offline: nonparametric posterior inference.** Given the queried labels, the GP posterior applies corrections at each input. At time step $t$, let $R_t$ follow the (approximate) posterior distribution of the residual. The rectified posterior predictive probability $P_t = g^{-1}(g(p_A) + R_t)$ gives a rectified point prediction $p_t$ as its posterior mean. We also rectify the confidence using the posterior risk estimate:

$$1-c_t = \mathbb{E}_{Y\sim\mathrm{Bern}(P_t)}[L(p_t, Y)] = \mathbb{E}_{P_t}[P_t L(p_t,1) + (1-P_t)L(p_t,0)] = p_t L(p_t,1) + (1-p_t)L(p_t,0).$$

Thus, in binary classification, the same posterior mean provides both the rectified prediction and confidence measure. For example, $c_t = \max\{p_t, 1-p_t\}$ for the 0–1 loss and $c_t = 1 - p_t(1-p_t)$ for the Brier loss. Since binary labels give a non-Gaussian posterior, we approximate it using expectation propagation (EP) or a Laplace approximation (Rasmussen & Williams, 2008).

**Online: assumed-density filtering.** Between offline checkpoints, we keep the prior fixed and update the residual posterior after each instance of actively acquired human feedback using assumed-density filtering (ADF), the online counterpart of EP. ADF obtains the Gaussian approximation of $R_{t+1}$ by matching its first two moments with those of the conditional posterior tilted by the new observation. Each update requires only a one-dimensional integral and a rank-one covariance correction; the probit link gives closed-form updates. An online maximum a posteriori (MAP) update provides an alternative online counterpart to the Laplace approximation. See Appendix C for more details.

When the GP model is well-specified and the human–AI residual is small, in the sense of the $L^2(P_X)$ norm, our rectified inference enjoys the variance reduction standard in model-assisted inference (Angelopoulos et al., 2023; Zrnic & Candès, 2026; Luo et al., 2025). This formally characterizes the efficiency gain from using an aligned AI model.

## 4 CONFIDENT JACK: CALIBRATING ADAPTIVE ESCALATION AND LEARNING

Rectification alone does not guarantee safety; we need an adaptive safety check that keeps pace with rectifier learning. We develop *Confident Jack* to control the risk of trusting the rectified model at any

[1] We normalize $k_\phi$ and $k_A$ to have comparable scale. For simplicity, we use the same feature space $\mathcal{X}$ for the bias functional and both kernels.

fixed time, regardless of the underlying AI model or rectifier. It uses the same queried labels used for rectifier learning for calibration, while accounting for how those labels were adaptively selected.

Confident Jack is a *co-evolving weight-score particle system* $\{(w_t^{(k)}, s_t^{(k)})\}_{k=1}^K$ that attends to the test input $X_t = x$, giving a prediction set of the following form for any risk tolerance $\epsilon \in (0, 1)$:

$$\mathcal{C}_{t,\epsilon}(x) = \left\{ y \in \mathcal{Y} : \sum_{k=1}^K w_t^{(k)}(x, y) \cdot \mathbf{1}\left\{ s_t^{(k)}(z^{(k)}) < s_t^{(k)}(x, y) \right\} < 1 - \epsilon/2 \right\}, \tag{2}$$

where $\{z^{(k)}\}$ are past human queries to be specified together with the particle system. Intuitively, scores help check if a hypothetical label $y$ is consistent with past queries, which may have different relevance to the current test point due to selective escalation and adaptive model updates, and thus are weighted accordingly. The prediction set retains labels that pass this weighted consistency check. This prediction set is translated into a confidence threshold for escalation in Section 5.

**Initialization: confidence-based nonconformity score.** To incorporate the model $\mathsf{M}_t$'s own uncertainty estimation, we define the score using its rectified confidence:

$$s(x, y; \mathsf{M}_t) \coloneqq c_t(x)\mathbf{1}\{f_t(x) \neq y\}, \tag{3}$$

where $f_t(x) = \mathbf{1}\{p_t(x) \geq 1/2\}$ is the rectified model prediction. The score is formally derived as the generalized inverse of the decision set of (1) in Appendix D. Intuitively, when $y$ is the true label, a confident mistake receives a higher nonconformity score than an uncertain one. With $c_t(x) = \max\{p_t(x), 1 - p_t(x)\}$ in binary classification, this recovers the prediction-based score in Romano et al. (2020). Our construction is tailored to our escalation rule and accommodates model-provided confidence measures.

**Offline: weighted cross-conformal calibration.** A score is associated with a model. Prioritizing query efficiency, Confident Jack constructs these scores by fitting leave-one-fold-out models $\{\mathsf{M}_t^{(k)}\}_{k=1}^K$. Each queried label can then support learning in some folds and calibration in another, avoiding a permanently separate calibration set. Let $s_t^{(k)} \coloneqq s(\cdot; \mathsf{M}_t^{(k)})$ and let $z^{(k)}$ be a designated point in the $k$th left-out dataset. With leave-one-out folds, this recovers leave-one-out cross-conformal prediction (Romano et al., 2020) and Jackknife+ (Barber et al., 2021) for binary classification.

The weights restore the symmetry lost through adaptation (history-dependent selective escalation and model improvement). For a candidate $(x, y)$, a weight answers how likely each held-out query would be to occupy the test slot when we do not know the order of the offline dataset plus the test point. We find this probability by *replaying* the logged acquisition policy under possible permutations of the held-out groups and the test point. Let $G$ be the allowed permutations and $\Lambda_t^\sigma(x, y)$ the likelihood of the observed acquisition decisions under permutation $\sigma \in G$; the required restrictions on $G$ and the replay are detailed in Appendix D. Then, the exact conditional probability that the designated point $z^{(k)}$ occupies the test slot is

$$w_t^{(k)}(x, y) \propto \textstyle\sum_{\{\sigma \in G : \sigma(k) = K+1\}} \Lambda_t^\sigma(x, y) \qquad k = 1, \ldots, K, \tag{4}$$

where $\sigma(k) = K + 1$ places the $k$th held-out group in the test slot $K + 1$. With a fixed, time-homogeneous query propensity $\pi_t(x) \equiv \pi(x)$, these become the usual normalized inverse-propensity weights $w_t^{(k)}(x) \propto 1/\pi(x^{(k)})$ (Tibshirani et al., 2020; Prinster et al., 2022).

**Online: co-evolving calibration particles.** We shift the focus to computational efficiency in the online period. Replaying every possible history in (4) would be costly online. We thus restrict the replay to the offline period while fixing the observed order of the subsequent online stream. Since the online period is order-invariant with respect to the offline dataset, we can factor the weight as:

$$w_t^{(k)}(x, y) \propto w_{t_0}^{(k)}(x, y) \cdot \Lambda_{t_0:t}^{(k)}, \tag{5}$$

where $t_0$ is the last offline checkpoint and $\Lambda_{t_0:t}^{(k)}$ is the likelihood of the observed online acquisition decisions for the $k$th held-out model; the derivation is deferred to Appendix D. This factorization places the main computational burden at the offline checkpoint to obtain $\{w_{t_0}^{(k)}\}$. During the online period, the particle system $\{\mathsf{M}_t^{(k)}\}$ co-evolves with the main model $\mathsf{M}_t$ using the same online stream, giving adaptive scores as in (3) and efficiently maintaining the weights by appending the online likelihood as in (5). A trade-off is that new online feedback improves the models immediately but

joins the pool of held-out calibration points only at the next offline checkpoint. Calibration therefore continues to use the offline feedback while its scores and weights adapt to the current model.

The co-evolution avoids treating model adaptation as adversarial drift. For example, as the model particles improve, nonconformity scores on past feedback become lower, making it easier for (2) to reject inconsistent test labels, yielding less conservative prediction sets and thus fewer escalations.

**Theorem 1** (Adaptive risk control). *Let $\mathcal{C}_{t,\epsilon}$ be the prediction set defined in (2) with the co-evolving particle system $\{(w_t^{(k)}, s_t^{(k)})\}_{k=1}^K$. Then, for any time $t$ (either an offline checkpoint or an online step) and any user-specified risk tolerance $\epsilon \in (0,1)$, we have*

$$P(Y_t \notin \mathcal{C}_{t,\epsilon}(X_t)) \leq \epsilon.$$

## 5 Exacting Gates: Safety Escalation Threshold on Confidence

To get fast and simple escalation decisions, we translate the prediction set (2) into confidence gates that adapt in both time and space, as used in (1). We have two observations: (i) stabilized learning closes the gap between the main model and the leave-one-out particles, and (ii) the exclusion of the opposite label $1 - f_t(x)$ from (2) indicates that it's *safe* to accept the model's chosen label $f_t(x)$, although it does not guarantee the inclusion of $f_t(x)$ (proved in Appendix E). Therefore, to test if the model is confident enough to reject its opposite label, the threshold is set to

$$\lambda_t(x) := \text{Quantile}_{\frac{1-\epsilon/2}{W_t(x)}}\left(\sum_{k\in\mathcal{W}_t(x)} \frac{w_t^{(k)}(x)}{W_t(x)}\,\delta\{s_t^{(k)}(z^{(k)})\}\right) + \gamma_t^c(x), \tag{6}$$

where weights $w_t^{(k)}(x) := w_t^{(k)}(x, 1-f_t(x))$ concern the opposite label, $\delta\{x\}$ is the Dirac measure at $x$, $W_t(x) = \sum_{k\in\mathcal{W}_t(x)} w_t^{(k)}(x)$ is the mass of the agreeing set $\mathcal{W}_t(x) := \{k \in [K] : f_t^{(k)}(x) = f_t(x)\}$, and $\gamma_t^c(x) = \max_{k\in\mathcal{W}_t(x)} \max\{0, c_t(x) - c_t^{(k)}(x)\}$ inflates the quantile to account for instability. We use the convention that max over empty $\mathcal{W}_t(x)$ is zero and $\text{Quantile}_c = \infty$ for $c > 1$.

Threshold (6) requires confidence to exceed a disagreement-inflated $(1-\epsilon/2)$-quantile of the agreeing particles' nonconformity scores. Intuitively, if most particles agree with the current prediction and are mostly correct, then $c_t(x)$ passes the threshold more easily; if few particles agree with the current prediction and those that agree are overconfident, then $c_t(x)$ must be very high to pass the threshold. Not only is $\lambda_t$ adaptive in time, but it also varies across the input space. For example, it's lower where the particle predictions are more stable (smaller $\gamma_t^c$ and larger $W_t$).

**Corollary 1** (Ensemble risk control). *Let $\hat{Y}_t$ be the decision output by CARE (1) with the safety escalation threshold in (6). Then, for any time $t$ (either an offline checkpoint or an online step) and any user-specified risk tolerance $\epsilon \in (0,1)$, we have $R_{01}(\hat{Y}_t) = P(\hat{Y}_t \neq Y_t) \leq \epsilon$.*

The proof is deferred to Appendix E, where we also relax the weight and score calculations to allow for approximation errors and accommodate computational constraints, offline or online.

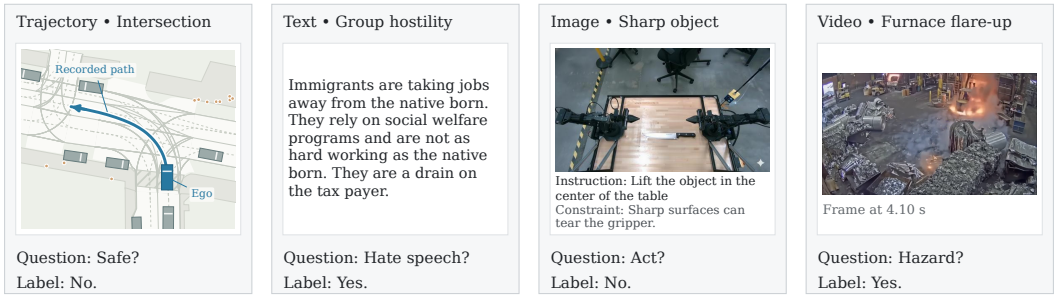
## 6 Active Max: Active Acquisition for Long-Term Efficiency

An active acquisition module $\pi$ in (1) independent of safety escalation is motivated by two needs. First, calibration requires replaying the acquisition policy on counterfactual histories to compute the weights in (4). If this acquisition policy coincides with the safety escalation policy, which is only known after calibration, then it creates a nested dependency and potentially exponential computational complexity. Second, active acquisition can potentially improve long-term data efficiency by prioritizing the human queries that can reduce future escalation frequency, which cannot be achieved by the safety escalation rule alone. This shares the same spirit as active statistical inference (Zrnic & Candès, 2026), but with a different objective: instead of maximizing inference precision, we aim to maximize efficiency.

This double escalation mechanism separates the input and output of the calibration particle system and lets human review protect current decisions and guide learning for future ones. Specifically, between offline checkpoints, actively escalated queries ($1 = A_t \sim \text{Bern}(\pi_t(X_t))$) update the rectifier and calibration particles; queries escalated only by the safety threshold ($1 = \mathbf{1}c_t(X_t) \leq \lambda_t(X_t)$) resolve the current decision, while their use for learning waits until the next checkpoint. We refer the reader to Appendix F for our active acquisition design.

## 7 EXPERIMENTS

We evaluate our method across four real-world safety-critical tasks, domains, and data modalities: collision detection in autonomous driving on the trajectory dataset nuPlan (Caesar et al., 2022), hate speech detection in LLM alignment on the text dataset HateBench (Shen et al., 2025), semantic safety in robotic tool use on the image dataset ASIMOV (Gemini Robotics Team, 2026), and hazard detection in the workplace on the video dataset iSafetyBench (Abdullah et al., 2025). Representative queries are shown in Figure 2. In all tasks, "AI" is an inexpensive model and "human" is a high-fidelity oracle. Results are reported over ten random seeds; detailed setups and full results are in Appendix A.




(a) nuPlan (b) HateBench (c) ASIMOV (d) iSafetyBench

Figure 2: Sample queries and benchmark labels. nuPlan shows recorded motion; its label refers to the benchmark simulation result. iSafetyBench shows one video frame.

We compare with three baselines: prediction-powered inference (PPI) (Angelopoulos et al., 2023), which learns a linear rectifier without calibration; selective escalation (SE) (Jung et al., 2025), which calibrates AI confidence using conformal prediction without rectification; and PPI+SE, which combines the two and uses separate data for rectification and calibration. We also include three ablations: GPC+SE replaces the PPI rectifier in PPI+SE with a GP classification rectifier; static CARE shares offline labels between rectification and calibration without online adaptation; and CARE with PPI replaces our GP rectifier with the PPI rectifier. The full CARE uses GP rectification with an offline–online learning and calibration schedule.

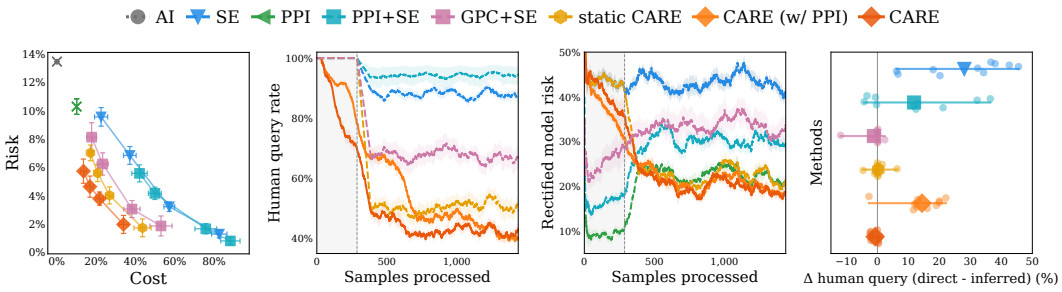
Here we present some representative results demonstrating four properties that are observed consistently across all experiments.

**Cost efficiency under risk control.** CARE achieves the lowest query complexity while keeping empirical risk below each tested tolerance. On iSafetyBench (Figure 3a), at the 5% risk tolerance, CARE escalates 21.7% of queries to humans, compared with 57.1% for SE, giving a 62% reduction.

**Fast learning.** CARE quickly learns useful residual structure with fewer human queries. On ASIMOV (Figures 3b and 3c), CARE variants achieve significantly higher efficiency through early-stage learning. Residual Gauss enables faster adaptation than PPI, and Confident Jack overcomes the data inefficiency exhibited by PPI+SE.

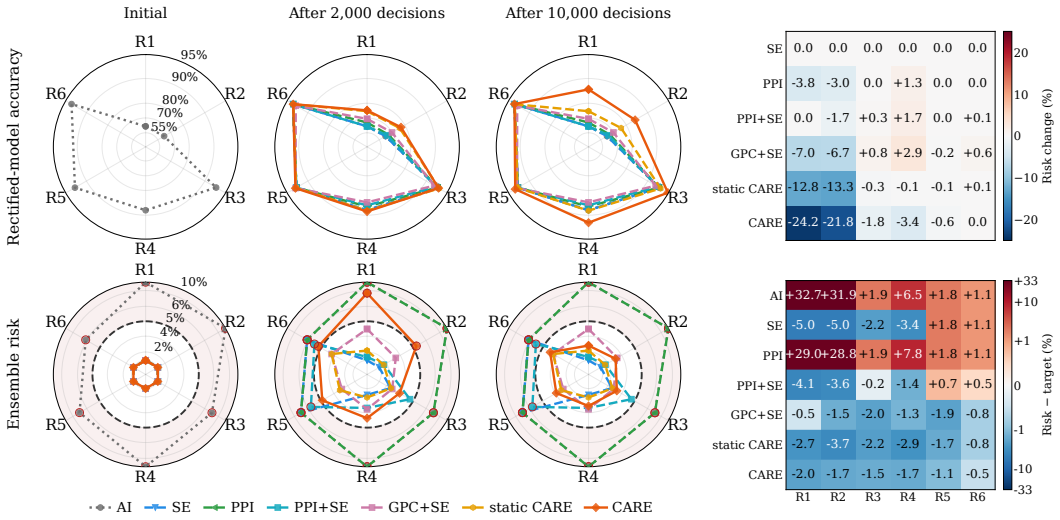
**Local adaptation.** While calibration controls marginal risk (Theorem 1), our GP model is designed to adapt locally, powering CARE to achieve more balanced performance across the input space. We examine this behavior across six physically defined regions in nuPlan (Figure 4). At the final snapshot, CARE achieves the largest model-risk reduction in every region, especially R1 and R2, where the original AI struggles. Its mean ensemble risk stays below the target in all six regions and is closest to the target among methods below it in R5 and R6.

**Robustness and generality.** CARE performs consistently across different domains, AI models, and confidence measures. On HateBench (Table 5), CARE reduces human queries and rectified-model error across all eight black-box AI models, whose raw error rates range from 14.5% to 44.7%. Query savings are 49–66% w.r.t. SE and 26–48% w.r.t. PPI+SE, with relative error reductions of 15.5–70.5%. On iSafetyBench (Figure 3d), CARE maintains similar query efficiency under direct (AI-provided) confidence, which is overconfident, and confidence inferred from the AI's predictive probability. For SE and PPI variants, however, the query rate increases by 28.0 pp and 11.8+ pp when switching from inferred to direct confidence. CARE gains this robustness by also rectifying the confidence measure.




(a) Risk–cost. (b) Human query rate. (c) Rectified-model risk. (d) Confidence sensitivity.

Figure 3: **(a)** iSafetyBench with AI-provided confidence and 10% preparation data for static methods. The $x$-axis is the total human query rate, and the $y$-axis is the empirical mean risk. Lines connect nominal risk targets of 2.5%, 5%, 7.5%, and 10%. Crosses mark risk exceeding the corresponding target. A Pareto frontier would lower-bound all methods from the lower left. **(b)–(c)** ASIMOV with inferred confidence, 5% risk target, and 20% preparation data for static methods. The $x$-axis covers all samples regardless of escalation, and the $y$-axis uses a rolling window of 100 samples. **(d)** iSafetyBench query-rate differences at the 2.5% risk target and 10% static preparation. Positive deviations indicate efficiency degradation.




(a) Regional adaptation. (b) Final regional differences.

Figure 4: Regional model improvement and ensemble risk at the 5% target, with 20% preparation data for static methods. Radars show successive snapshots; heatmaps summarize the final snapshot on the same held-out panel of 2,000 scenarios. The risk target is highlighted as a dashed ring in the lower row, and the red band and outlines indicate exceedance. Region definition: open-loop time to collision below 1 s for R1–R2, finite and at least 1 s for R3–R4, and infinite for R5–R6; average displacement error below 5 m for R1, R3, R5, and at least 5 m for R2, R4, R6.

## 8 Conclusion

We introduced CARE, an end-to-end framework for safe and efficient human–AI collaborative decision making. By combining model-aware residual learning with model-agnostic adaptive calibration, CARE uses human feedback to improve future predictions while maintaining control of the final system's risk. Experiments on real-world safety-critical tasks demonstrate substantial query savings across domains, AI models, and confidence measures. Together, these results provide a theoretical and empirical foundation for safeguarding continuously improving AI models towards greater autonomy.

### AI USE STATEMENT

In this work, we used generative AI tools for polishing the language of the manuscript, and for writing the code of the experiments. We have not used generative AI tools for method or theory development, and generating synthetic data sets is not applicable to this work. We have reviewed all AI-assisted writing. LLM-generated code was verified and tested for correctness by all authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

Datasets and benchmarks may contain sensitive content, including hate speech and unsafe scenarios. Please refer to the original dataset papers for their ethical considerations and data collection procedures. No new data or human subject experiment is introduced in this project.

### REPRODUCIBILITY STATEMENT

The code for reproducing the experiments is in the supplementary material, and will be released publicly upon acceptance.

## REFERENCES

Raiyaan Abdullah, Yogesh Singh Rawat, and Shruti Vyas. iSafetyBench: A video-language benchmark for safety in industrial environment. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 1444–1453. IEEE, 2025.

Anastasios N. Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. Prediction-Powered Inference, 2023.

Apurva Badithela, David Snyder, Lihan Zha, Joseph Mikhail, Matthew O'Kelly, Anushri Dixit, and Anirudha Majumdar. Reliable and scalable robot policy evaluation with imperfect simulators. In *Conference on Robot Learning (CoRL)*, 2025.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1), 2021. ISSN 0090-5364. doi: 10.1214/20-AOS1965.

Aadyot Bhatnagar, Huan Wang, Caiming Xiong, and Yu Bai. Improved Online Conformal Prediction via Strongly Adaptive Online Learning, 2023.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles, 2022.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Ran El-Yaniv and Yair Wiener. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Shai Feldman, Liran Ringel, Stephen Bates, and Yaniv Romano. Achieving Risk Control in Online Learning Settings, 2023.

Gemini Robotics Team. Gemini Robotics 2: Safety Evaluations. Technical report, Google DeepMind, 2026.

Isaac Gibbs and Emmanuel Candès. Adaptive Conformal Inference Under Distribution Shift, 2021.

Isaac Gibbs and Emmanuel Candès. Conformal Inference for Online Prediction with Arbitrary Distribution Shifts, 2022.

Patrick Hemmer, Max Schemmer, Niklas Kühl, Michael Vössing, and Gerhard Satzger. Complementarity in human-ai collaboration: Concept, sources, and evidence. *European Journal of Information Systems*, 34(6):979–1002, 2025.

Zilin Huang, Zihao Sheng, Yansong Qu, Junwei You, and Sikai Chen. VLM-RL: A unified vision language models and reinforcement learning framework for safe autonomous driving. *arXiv preprint arXiv:2412.15544*, 2024.

Jaehun Jung, Faeze Brahman, and Yejin Choi. Trust or escalate: Llm judges with provable guarantees for human agreement. In *International Conference on Learning Representations*, volume 2025, pp. 3101–3125, 2025.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Lu, Thomas Mesnard, Colton Bishop, Victor Carvalho, Felix Jiang, Doina Precup, Antoine Fleury, et al. RLAIF: Scaling reinforcement learning from human feedback with AI feedback. In *International Conference on Learning Representations (ICLR)*, 2024.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2017.1307116.

Rachel Luo, Heng Yang, Michael Watson, Apoorva Sharma, Sushant Veer, Edward Schmerling, and Marco Pavone. Sim2val: Leveraging correlation across test platforms for variance-reduced metric estimation. *arXiv preprint arXiv:2506.20553*, 2025.

Rachel Luo, Michael Watson, Apoorva Sharma, Heng Yang, Han Qi, Edward Schmerling, Sushant Veer, Boris Ivanovic, and Marco Pavone. X4Val: Learning Neural Surrogates for Variance-Reduced Policy Evaluation, 2026.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Thomas P. Minka. Expectation Propagation for approximate Bayesian inference, 2013.

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

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive Confidence Machines for Regression. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Tapio Elomaa, Heikki Mannila, and Hannu Toivonen (eds.), *Machine Learning: ECML 2002*, volume 2430, pp. 345–356. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-540-44036-9 978-3-540-36755-0. doi: 10.1007/3-540-36755-1_29.

Drew Prinster, Anqi Liu, and Suchi Saria. JAWS: Auditing Predictive Uncertainty Under Covariate Shift, 2022.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. ISBN 978-0-262-18253-9.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners, 2023.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with Valid and Adaptive Coverage, 2020.

Pierre Sermanet, Anirudha Majumdar, Alex Irpan, Dmitry Kalashnikov, and Vikas Sindhwani. Generating robot constitutions & benchmarks for semantic safety. *arXiv preprint arXiv:2503.08663*, 2025.

Xinyue Shen, Yixin Wu, Yiting Qu, Michael Backes, Savvas Zannettou, and Yang Zhang. {HateBench}: Benchmarking Hate Speech Detectors on {LLM-Generated} Content and Hate Campaigns. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 221–240, 2025.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

Aman Sinha, Payam Nikdel, Supratik Paul, and Shimon Whiteson. Rate-Informed Discovery via Bayesian Adaptive Multifidelity Sampling, 2024.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candes, and Aaditya Ramdas. Conformal Prediction Under Covariate Shift, 2020.

Harsh Vardhan, Sunav Choudhary, Natwar Modani, and Arya Mazumdar. Improving Selective Classification with Pairwise Queries for Binary Classification, 2026.

Harit Vishwakarma, Heguang Lin, Frederic Sala, and Ramya Korlakai Vinayak. Promises and Pitfalls of Threshold-based Auto-labeling, 2024.

Vladimir Vovk. Cross-conformal predictors, 2012.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Tijana Zrnic and Emmanuel J. Candès. Active Statistical Inference, 2026.

# A EXTENDED EXPERIMENTAL SETUPS AND RESULTS

**Decisions, queries, and metrics.** Each dataset comes with a binary decision task, with a fixed inexpensive predictor as "AI", and a benchmark reference label as the correct "human" response. Ensemble risk divides final errors by all processed examples, including queried examples which contribute no error. Query rate divides all oracle calls by the same denominator and includes initial offline static preparation dataset (training and/or calibration) and online adaptive escalations. Reusing a purchased label for fitting or calibration does not incur a second query. Rectified-model risk instead evaluates the predictor before oracle intervention, including examples that are ultimately queried.

**Paired evaluation.** Each study uses ten randomized trials, pairing the data stream within trial across methods, risk targets, and preparation budgets. Means and standard errors across the ten trials are reported. nuPlan and HateBench have reserved evaluation panels for assessing the rectified model. This evaluation/test dataset does not enter the data stream, is not used for training or calibration, and thus is not counted in the query complexity. Feature design may use covariate information, but no label information is revealed in this preparation stage.

**Methods.**

- AI uses the original prediction without queries.
- PPI fits a linear rectifier on the static preparation data and then predicts without safety escalation.
- SE calibrates the original AI confidence.
- PPI+SE splits the static preparation data approximately equally between fitting the linear rectifier and calibrating escalation.
- GPC+SE replaces the linear rectifier in PPI+SE with a GP classifier, but otherwise uses the same static preparation and calibration.
- Static `CARE` uses GP rectification and cross-fitted calibration on the shared static preparation data, then freezes the fitted policy.
- (Adaptive) `CARE` does not have static preparation data. It starts with a conservative confidence threshold and learns from labels obtained through escalation.
- `CARE` with PPI replaces the GP rectifier in `CARE` with a linear one, but otherwise uses the same adaptive calibration and escalation.

**Prepare–offline–online schedule.** Static methods reserves a preparation fraction of the entire data stream. This fraction is thus a hyperparameter for static methods, but not for (adaptive) `CARE`. Offline checkpoints are a hyperparameter for adaptive methods, and follow batches of 32, 64, 128, 256, 512, 1,024, and 2,048 newly queried labels, with the final interval repeated.

**Gap between theory and practice.** Active escalation coincides with safety escalation in our experiments. Computational approximations include sampled counterfactual replays using truncated horizons and logged acquisition decisions, and shorter EP iterations for the particle system. Reported empirical risk only approximates the marginal risk guaranteed by Theorem 1 and Corollary 1.

Table 1 summarizes the principal settings.

Table 1: Shared protocol and dataset-specific settings. The default operating point uses 20% preparation and a 5% risk target.

| | nuPlan | HateBench | iSafetyBench | ASIMOV |
|---|---|---|---|---|
| Stream dataset size | 10,000 | 7,054 | 1,100 | 1,437 |
| Reserved test panel | 2,000 | 784 | — | — |
| Rolling-window size | 1,000 | 500 | 100 | 100 |
| GP feature dimension | 256 | 64 | 256 | 256 |
| Particle system size | 24 | 5 | 20 | 20 |
| Preparation fractions (%) | [10, 20, 30, 40] | [10, 20, 30, 40] | [10, 20, 30, 40] | [10, 20, 30, 40] |
| Risk targets (%) | [1, 3, 5, 7.5, 10] | [1, 3, 5, 7.5, 10] | [2.5, 5, 7.5, 10] | [5, 10, 15, 20] |

## A.1 NUPLAN

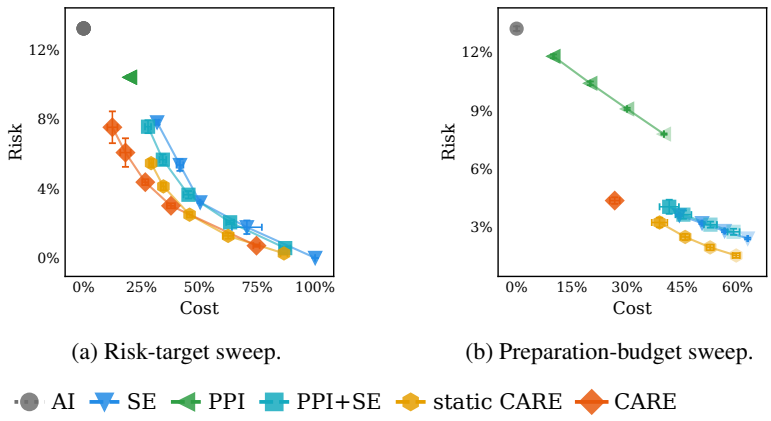
**Task and evaluation.** We use nuPlan driving scenarios (Caesar et al., 2022) with its inexpensive "AI" predictor as the open-loop simulator and its "human" oracle as the closed-loop oracle. The binary task

is to collision prediction, with the label being 1 if the closed-loop simulation has a time-to-collision (TTC) of less than one second. The "AI" predictor's probability of safety is a probit transform of the open-loop TTC margin, with scale 0.5 seconds and probabilities clipped to $[10^{-6}, 1 - 10^{-6}]$.

The population contains 49,763 scenarios from six intersection-related scenario types and 63 driving logs. A fixed, label-blind split reserves 9,953 scenarios, stratified by the six physical regions in Table 3; 39,810 remain eligible for streaming. Each of ten paired trials randomly orders this pool and processes its first 10,000 scenarios. Regional evaluation uses the same region-stratified panel of 2,000 reserved scenarios for every trial and method, with population region weights used for pooled snapshot metrics. Driving logs occur in both partitions, so this evaluates within-population adaptation.

**Feature space.** The rectifier receives ten engineered open-loop scalars and eight joint principal components of 384 ego-, agent-, and road-embedding coordinates. The scalar and embedding blocks contribute 80% and 20% of normalized feature energy, respectively. Representations are fitted without labels on each trial's first 2,000 covariates, frozen, and shared across methods.

**Risk–cost tradeoffs.** Figure 5 and Table 2 show the tradeoff across all five risk targets and four preparation budgets. At the primary operating point, CARE achieves 4.36% ensemble risk with 26.6% queries, compared with 50.3% for SE, 45.3% for PPI+SE, and 45.8% for static CARE. These correspond to query reductions of 47%, 41%, and 42%, respectively. Mean CARE risk is below every tested target, while query cost falls from 74.6% at the 1% target to 12.4% at the 10% target. Larger static preparation budgets generally reduce ensemble risk but increase total query cost: buying more labels does not ensure that subsequent query savings repay their cost. Adaptive CARE avoids choosing a fixed preparation size and has lower mean query cost than these three comparators throughout the reported grid.




(a) Risk-target sweep.

(b) Preparation-budget sweep.

Figure 5: nuPlan risk–cost tradeoffs. (a) Targets of 1%, 3%, 5%, 7.5%, and 10%, with 20% static preparation. (b) 10%, 20%, 30%, and 40% preparation at the 5% target, shown with decreasing marker opacity.

**Adaptive learning.** Figure 6 presents adaptive learning through model improvement, escalation cost, and final-ensemble error. In the first 1,000 arrivals, CARE queries 43.8% and has 13.1% model risk; in the final 1,000 arrivals these become 25.6% and 8.6%. Static CARE ends at 32.0% queries and 11.1% model risk. Continued residual learning therefore improves decisions before intervention and reduces the amount of intervention needed. The final-window ensemble risk is 3.7% for CARE. Short-window risk may fluctuate around the nominal target; these trajectories describe successive deployed policies rather than a guarantee for every realized window.

**Local adaptation.** The frozen-model analysis in Figure 4 isolates spatial variation from changes in the arriving stream. Table 3 supplies the region definitions, evaluation support, and final CARE values. The strongest correction occurs in R1 and R2, where open-loop TTC is below one second: model risk falls from 37.7% to 13.6% and from 36.9% to 15.1%, respectively. These regions also receive the most oracle queries, reflecting their greater remaining difficulty. On the full weighted panel, CARE reduces model risk from 12.7% initially to 10.2% after 2,000 decisions and 7.9% after 10,000; static

Table 2: nuPlan risk–cost tradeoffs. Ensemble risk / oracle query rate (%) over the complete target and static-preparation grid.

| Target | Prep. | SE | PPI+SE | static CARE | CARE |
|---|---|---|---|---|---|
| 1% | 10% | 0.00 / 100.0 | 0.57 / 87.0 | 0.43 / 84.3 | 0.70 / 74.6 |
| 1% | 20% | 0.00 / 100.0 | 0.56 / 87.1 | 0.26 / 86.5 | 0.70 / 74.6 |
| 1% | 30% | 0.00 / 100.0 | 0.55 / 87.2 | 0.16 / 89.3 | 0.70 / 74.6 |
| 1% | 40% | 0.00 / 100.0 | 0.47 / 89.6 | 0.10 / 91.5 | 0.70 / 74.6 |
| 3% | 10% | 2.04 / 66.0 | 2.46 / 56.9 | 1.85 / 55.4 | 3.00 / 37.7 |
| 3% | 20% | 1.76 / 70.5 | 2.05 / 63.3 | 1.27 / 62.4 | 3.00 / 37.7 |
| 3% | 30% | 1.28 / 78.2 | 1.76 / 68.4 | 0.97 / 67.2 | 3.00 / 37.7 |
| 3% | 40% | 1.15 / 81.0 | 1.55 / 72.1 | 0.72 / 72.8 | 3.00 / 37.7 |
| 5% | 10% | 3.60 / 44.2 | 4.04 / 41.4 | 3.23 / 38.8 | 4.36 / 26.6 |
| 5% | 20% | 3.20 / 50.3 | 3.63 / 45.3 | 2.48 / 45.8 | 4.36 / 26.6 |
| 5% | 30% | 2.79 / 56.5 | 3.12 / 52.6 | 1.95 / 52.5 | 4.36 / 26.6 |
| 5% | 40% | 2.40 / 62.7 | 2.75 / 58.9 | 1.53 / 59.6 | 4.36 / 26.6 |
| 7.5% | 10% | 5.73 / 35.4 | 6.05 / 28.9 | 4.98 / 27.5 | 6.06 / 18.1 |
| 7.5% | 20% | 5.34 / 41.6 | 5.65 / 34.2 | 4.11 / 34.4 | 6.06 / 18.1 |
| 7.5% | 30% | 4.38 / 49.9 | 4.88 / 42.1 | 3.32 / 42.8 | 6.06 / 18.1 |
| 7.5% | 40% | 4.15 / 55.4 | 4.28 / 49.8 | 2.73 / 51.1 | 6.06 / 18.1 |
| 10% | 10% | 8.88 / 22.9 | 8.04 / 21.5 | 6.73 / 20.7 | 7.51 / 12.4 |
| 10% | 20% | 7.80 / 31.7 | 7.55 / 27.8 | 5.45 / 29.1 | 7.51 / 12.4 |
| 10% | 30% | 6.77 / 40.3 | 6.49 / 37.0 | 4.65 / 37.7 | 7.51 / 12.4 |
| 10% | 40% | 5.84 / 48.8 | 5.81 / 45.3 | 3.82 / 47.1 | 7.51 / 12.4 |

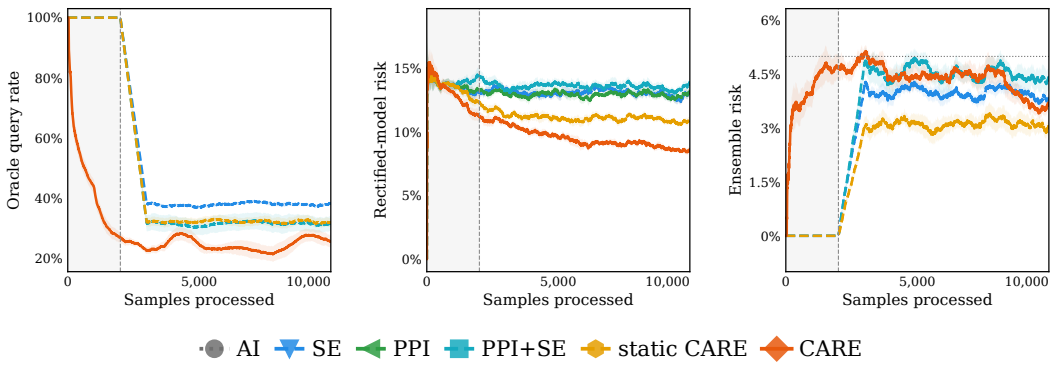



Figure 6: nuPlan learning trajectories at the 5% target and 20% static preparation. Lines average trailing windows of up to 1,000 arrivals over ten trials. Gray shading marks the 2,000-arrival static preparation period. AI and PPI are shown as uncalibrated references in the model-risk panel. The dotted horizontal line is the nominal ensemble-risk target.

CARE remains at 10.4% after preparation. Final mean regional ensemble risks for CARE range from 3.0% to 4.5%, all below the 5% target. As in the main text, GPC+SE also remains below the target in every region, whereas SE and PPI+SE exceed it in R5 and R6. This is an empirical regional balance result; the calibration target concerns marginal risk, and the smaller regions have correspondingly larger sampling uncertainty.

### A.2 HATEBENCH

**Task and evaluation.** We use HateBench responses (Shen et al., 2025) with eight cached detectors as the "AI" predictors and released human hate/non-hate annotations as the oracle labels. The population contains 7,838 responses; a fixed, label-blind split reserves 784 responses by normalized-text group, leaving 7,054 for each trial. The reserved panel and stream orders are shared across detectors. Prompt families may occur in both partitions, so this evaluates within-population adaptation. Perspective and both Detoxify variants use the identity-attack score at threshold 0.5; the other detectors retain

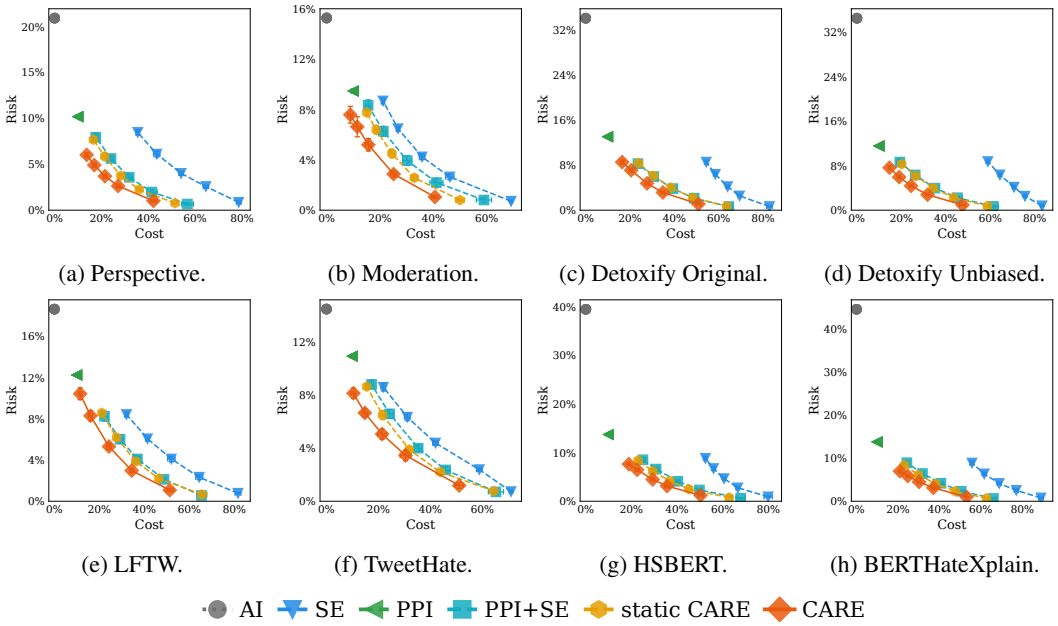
| | Region | Panel size | AI risk (%) | Final CARE (%) | | |
|---|---|---|---|---|---|---|
| | | | | Model risk | Ensemble risk | Queries |
| R1 | TTC $< 1$, ADE $< 5$ | 53 | 37.7 | $13.6 \pm 1.5$ | $3.0 \pm 1.1$ | $49.6 \pm 5.0$ |
| R2 | TTC $< 1$, ADE $\geq 5$ | 293 | 36.9 | $15.1 \pm 0.5$ | $3.3 \pm 0.4$ | $56.5 \pm 3.7$ |
| R3 | Finite TTC $\geq 1$, ADE $< 5$ | 320 | 6.9 | $5.1 \pm 0.2$ | $3.5 \pm 0.2$ | $13.8 \pm 1.9$ |
| R4 | Finite TTC $\geq 1$, ADE $\geq 5$ | 374 | 11.5 | $8.1 \pm 0.5$ | $3.3 \pm 0.3$ | $30.6 \pm 2.8$ |
| R5 | Infinite TTC, ADE $< 5$ | 468 | 6.8 | $6.3 \pm 0.2$ | $3.9 \pm 0.3$ | $11.9 \pm 1.4$ |
| R6 | Infinite TTC, ADE $\geq 5$ | 492 | 6.1 | $6.1 \pm 0.1$ | $4.5 \pm 0.3$ | $10.1 \pm 1.6$ |

Table 3: nuPlan regional definitions and final frozen-model results at the 5% target. TTC (time to collision) is in seconds and ADE (average displacement error) is in meters.

their native binary decisions with scores oriented toward the hate class. The confidence supplied to escalation is the absolute margin from 0.5, scaled to $[0, 1]$.

**Feature space.** The rectifier receives response-text TF–IDF features with unigrams and bigrams, reduced to 32 singular-vector coordinates, together with log response length and the selected detector's available category scores. Coordinates are standardized, clipped at six, and divided by the square root of the feature dimension. Representations are fitted without labels on each trial's first 20% of covariates, frozen, and shared across methods. The same fixed GP prior is used across all eight detectors.

**Risk–cost tradeoffs.** Figure 7 show the tradeoff across all five risk targets for all models, and selectively, Table 4 reports the tradeoff across four preparation budgets for Perspective. With Perspective, at the 5% target and 10% static preparation, CARE achieves 3.71% ensemble risk with 21.5% queries, compared with 54.0% for SE, 31.8% for PPI+SE, and 28.3% for static CARE. Query cost falls from 42.2% at the 1% target to 13.7% at the 10% target. Larger static preparation budgets generally reduce ensemble risk but increase total query cost. Adaptive CARE has lower mean query cost than these three comparators throughout the reported grid. Similar trends hold with other models.




(a) Perspective. (b) Moderation. (c) Detoxify Original. (d) Detoxify Unbiased.

(e) LFTW. (f) TweetHate. (g) HSBERT. (h) BERTHateXplain.

Figure 7: HateBench risk–cost tradeoffs with different models. Targets of 1%, 3%, 5%, 7.5%, and 10%, with 10% static preparation.

**Adaptive learning.** Figure 8 presents adaptive learning with Perspective through model improvement, escalation cost, and final-ensemble error. In the first 500 arrivals, CARE queries 41.4% and has 14.3% model risk; in the final 500 arrivals these become 16.8% and 9.4%. Continued residual learning

Table 4: HateBench risk–cost tradeoffs with Perspective. Ensemble risk / oracle query rate (%) over the complete target and static-preparation grid.

| Target | Prep. | SE | PPI+SE | static CARE | CARE |
|---|---|---|---|---|---|
| 1% | 10% | 0.86 / 78.6 | 0.67 / 56.7 | 0.77 / 51.4 | 1.02 / 42.2 |
| 1% | 20% | 0.81 / 80.3 | 0.54 / 57.5 | 0.69 / 53.8 | 1.02 / 42.2 |
| 1% | 30% | 0.70 / 82.8 | 0.57 / 59.8 | 0.60 / 58.6 | 1.02 / 42.2 |
| 1% | 40% | 0.59 / 85.3 | 0.53 / 64.1 | 0.53 / 64.0 | 1.02 / 42.2 |
| 3% | 10% | 2.58 / 64.6 | 1.97 / 41.5 | 2.26 / 36.1 | 2.64 / 27.1 |
| 3% | 20% | 2.31 / 68.2 | 2.01 / 41.8 | 2.02 / 40.7 | 2.64 / 27.1 |
| 3% | 30% | 1.98 / 72.5 | 1.66 / 48.5 | 1.73 / 47.3 | 2.64 / 27.1 |
| 3% | 40% | 1.73 / 76.0 | 1.58 / 54.8 | 1.62 / 54.0 | 2.64 / 27.1 |
| 5% | 10% | 4.01 / 54.0 | 3.59 / 31.8 | 3.75 / 28.3 | 3.71 / 21.5 |
| 5% | 20% | 3.70 / 58.3 | 3.56 / 34.2 | 3.50 / 33.8 | 3.71 / 21.5 |
| 5% | 30% | 3.20 / 63.6 | 3.20 / 41.2 | 3.04 / 41.4 | 3.71 / 21.5 |
| 5% | 40% | 2.87 / 68.1 | 2.64 / 49.7 | 2.78 / 48.9 | 3.71 / 21.5 |
| 7.5% | 10% | 6.11 / 43.6 | 5.63 / 24.1 | 5.86 / 21.3 | 4.91 / 16.9 |
| 7.5% | 20% | 5.50 / 49.2 | 5.45 / 28.1 | 5.36 / 27.8 | 4.91 / 16.9 |
| 7.5% | 30% | 4.79 / 55.6 | 4.94 / 35.7 | 4.67 / 36.1 | 4.91 / 16.9 |
| 7.5% | 40% | 4.22 / 61.4 | 4.24 / 44.8 | 4.29 / 44.4 | 4.91 / 16.9 |
| 10% | 10% | 8.48 / 35.5 | 7.95 / 17.6 | 7.68 / 16.5 | 6.03 / 13.7 |
| 10% | 20% | 7.68 / 42.3 | 7.36 / 23.2 | 6.84 / 24.0 | 6.03 / 13.7 |
| 10% | 30% | 6.75 / 49.4 | 6.77 / 31.4 | 5.94 / 32.9 | 6.03 / 13.7 |
| 10% | 40% | 5.82 / 56.5 | 5.73 / 41.3 | 5.19 / 42.2 | 6.03 / 13.7 |

therefore improves decisions before intervention and reduces the amount of intervention needed. The final-window ensemble risk is 3.7% for CARE.

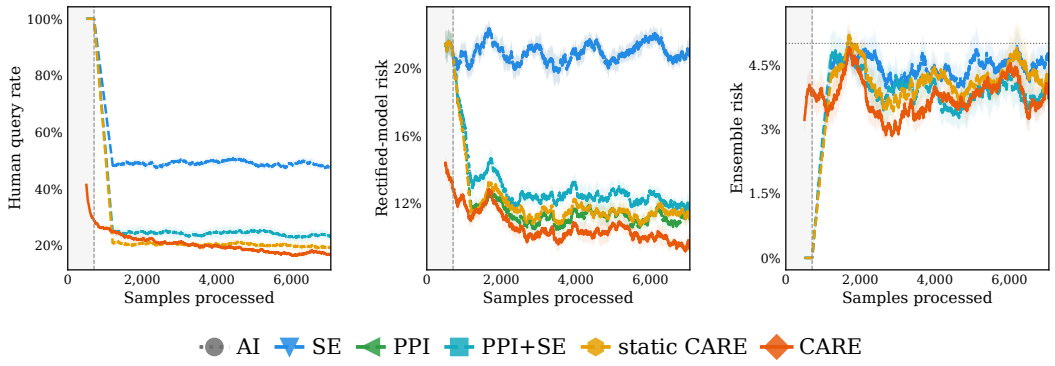



Figure 8: HateBench learning trajectories with Perspective at the 5% target and 10% static preparation. Lines average complete trailing windows of 500 arrivals over ten trials. Gray shading marks the 705-arrival static preparation period. AI and PPI are shown as uncalibrated references in the model-risk panel. The dotted horizontal line is the nominal ensemble-risk target.

**Robustness across AI models.** Table 5 summarize query cost and held-out model improvement across all eight detectors. At the 5% target and 10% static preparation, CARE uses 15.6–30.2% queries with mean ensemble risks of 3.71–5.31%. These correspond to query reductions of 49–66% relative to SE and 26–48% relative to PPI+SE. On the reserved panel, raw model risk ranges from 14.2% to 47.3%, while final CARE model risk ranges from 10.8% to 14.6%. The relative reductions of 15.5–70.5% show that residual learning improves substantially different initial predictors.

Table 5: HateBench comparison across AI models at the 5% target and 10% static preparation. All values are percentages. Raw AI error is measured on the stream; model-error reduction is measured on the reserved panel.

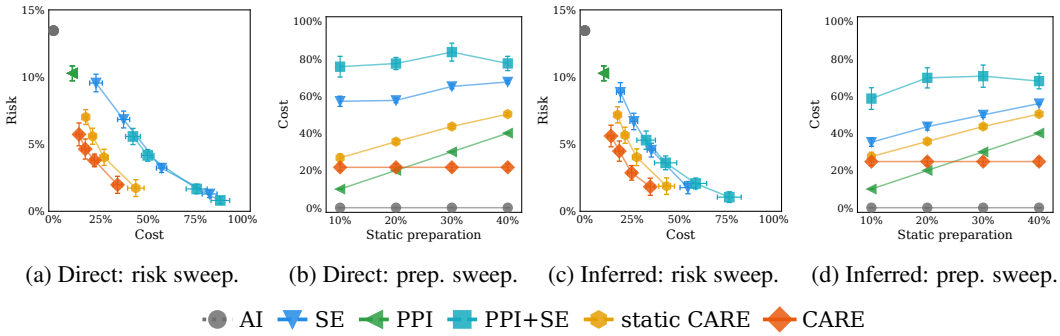
| AI model | Raw AI error | Human query rate | | | Relative improvement | | |
|---|---|---|---|---|---|---|---|
| | | SE | PPI+SE | CARE | Query saving vs SE ↑ | Query saving vs PPI+SE ↑ | Model-error reduction ↑ |
| Perspective | 20.9 | $54.0 \pm 0.9$ | $31.8 \pm 1.5$ | $21.5 \pm 0.7$ | 60.2 | 32.5 | 40.5 |
| Moderation | 15.3 | $35.7 \pm 1.0$ | $30.2 \pm 1.7$ | $15.6 \pm 1.2$ | 56.3 | 48.2 | 15.5 |
| Detoxify Original | 34.1 | $64.0 \pm 0.5$ | $39.3 \pm 1.5$ | $27.7 \pm 1.0$ | 56.7 | 29.6 | 58.6 |
| Detoxify Unbiased | 34.5 | $70.6 \pm 0.6$ | $35.1 \pm 1.3$ | $24.3 \pm 0.7$ | 65.5 | 30.6 | 65.8 |
| LFTW | 18.6 | $52.0 \pm 1.6$ | $36.9 \pm 1.2$ | $24.2 \pm 1.0$ | 53.4 | 34.3 | 27.2 |
| TweetHate | 14.5 | $41.9 \pm 1.1$ | $35.3 \pm 1.4$ | $21.4 \pm 1.6$ | 49.0 | 39.3 | 23.4 |
| HSBERT | 39.4 | $60.6 \pm 0.5$ | $40.3 \pm 0.9$ | $29.3 \pm 0.9$ | 51.6 | 27.2 | 65.7 |
| BERTHateXplain | 44.7 | $68.8 \pm 1.0$ | $40.7 \pm 1.2$ | $30.2 \pm 0.7$ | 56.2 | 25.9 | 70.5 |

### A.3 iSafetyBench

**Task and evaluation.** We use iSafetyBench videos (Abdullah et al., 2025) for binary workplace hazard detection, with the frozen Qwen3-VL-2B-Instruct model as the "AI" predictor and released category annotations as the oracle labels. The population contains 1,100 clips: 420 hazardous and 680 routine. The AI receives 4–32 frames sampled across each clip, without audio. Its hazard probability $p_A$ is obtained from the two answer logits, and a separate Yes/No follow-up supplies direct confidence $c_{\text{direct}} = P(\text{Yes})$. We compare this with inferred confidence $c_{\text{inferred}} = 2|p_A - 1/2|$, holding the video inputs and initial AI decisions fixed. Each trial processes all 1,100 clips from 151 source-video families, grouped within calibration folds.

**Feature space.** The rectifier receives the 2,048-dimensional final-layer prompt state, reduced to 32 principal components. Representations are fitted without labels on each trial's first 20% of covariates, frozen, and shared across methods and confidence measures.

**Risk–cost tradeoffs.** Figure 9 show the tradeoff across all four risk targets and four preparation budgets under both confidence measures. At the 5% target and 10% static preparation, direct-confidence CARE achieves 3.79% ensemble risk with 21.7% queries, compared with 57.1% for SE and 26.7% for static CARE. With inferred confidence, it achieves 2.86% risk with 24.8% queries, compared with 35.1% and 27.4% for these baselines. Mean CARE risk is below every tested target under both confidence measures. Larger static preparation budgets generally reduce ensemble risk but increase total query cost: static CARE's query rate rises from 26.7% to 50.2% across the direct-confidence budget sweep at the 5% target.

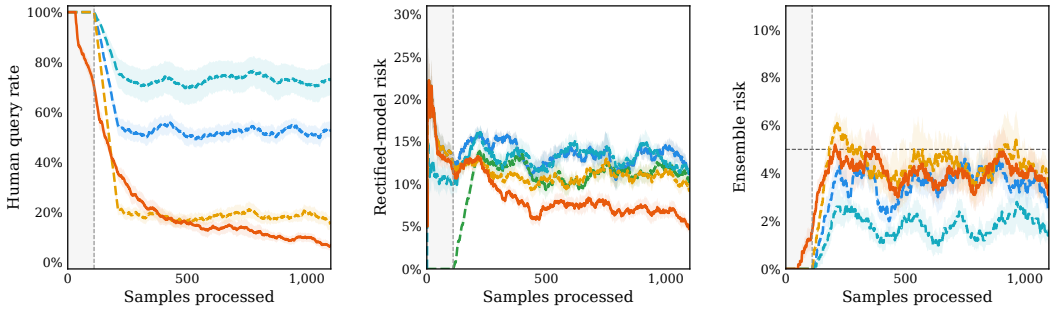



(a) Direct: risk sweep. (b) Direct: prep. sweep. (c) Inferred: risk sweep. (d) Inferred: prep. sweep.

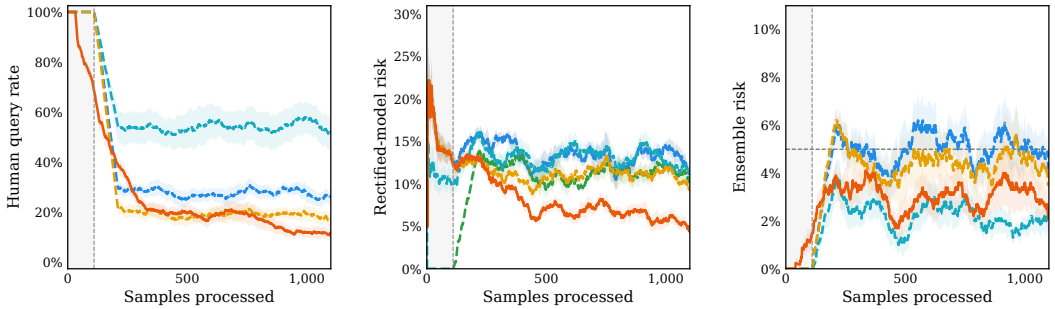
Figure 9: iSafetyBench risk–cost tradeoffs. (a, c) Targets of 2.5%, 5%, 7.5%, and 10%, with 10% static preparation. (b, d) 10%, 20%, 30%, and 40% preparation at the 5% target.

**Adaptive learning.** Figure 10 presents adaptive learning through model improvement, escalation cost, and final-ensemble error. With direct confidence, CARE queries 25.3% and has 9.3% model risk

in the window ending at 300 clips; in the final 100 clips these become 6.9% and 4.8%. Static CARE ends at 16.0% queries and 9.4% model risk. Continued residual learning therefore improves decisions before intervention and reduces the amount of intervention needed. The inferred-confidence results show a similar reduction in model risk, reaching 4.5% in the final window.




(a) Direct confidence.




(b) Inferred confidence.

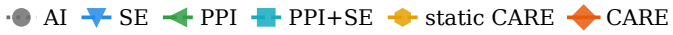



Figure 10: iSafetyBench learning trajectories at the 5% target and 10% static preparation. Lines average trailing windows of up to 100 arrivals over ten trials. Gray shading marks the 110-arrival static preparation period, where static model risks can include in-sample predictions. AI and PPI are shown as uncalibrated references in the model-risk panels. The dashed horizontal line is the nominal ensemble-risk target.

**Robustness to confidence choice.** Inferred confidence ranks the original AI's correct decisions more accurately than direct confidence (correctness AUROC 0.803 versus 0.642). Table 6 extends the paired comparison in Figure 3d to all four risk targets. At 2.5%, replacing inferred with direct confidence increases query rates by 28.0 percentage points for SE, 11.8 for PPI+SE, and 14.4 for CARE with PPI, while GP CARE changes by only $-0.7$ points. Across the four targets, GP CARE's absolute mean query-rate change is at most 3.1 points. Rectifying confidence together with predictions reduces reliance on the original confidence measure.

### A.4 ASIMOV

**Task and evaluation.** We use ASIMOV-Agentic queries (Gemini Robotics Team, 2026) for binary robot dispatch decisions, with the frozen Qwen3-VL-2B-Instruct model as the "AI" predictor and benchmark reference decisions as the oracle labels. The population contains 1,437 queries: 641 on physical constraints, 396 on instruction ambiguity, and 400 on VLA feasibility. The binary task is deciding whether to dispatch the instruction for immediate execution. Physical-constraint references are human annotated; the other subsets use the released benchmark references. The AI receives the image, instruction, and tool schemas, and its decision probability is obtained from the two answer logits. Direct and inferred confidence follow the same construction as in iSafetyBench, holding

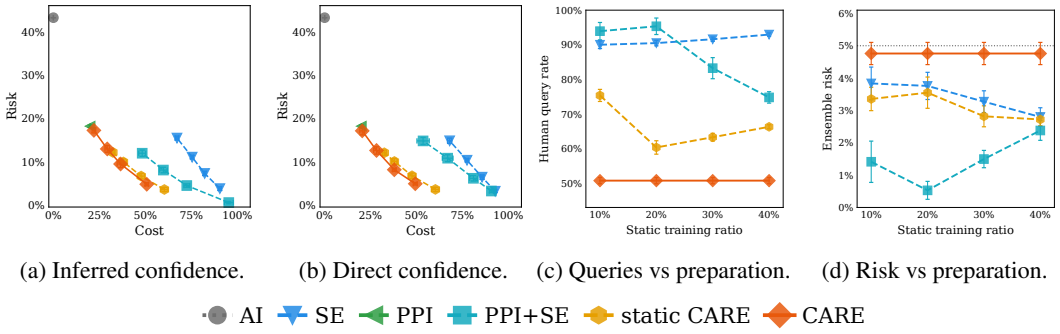
Table 6: Effect of confidence choice on iSafetyBench with 10% static preparation. Entries are paired mean differences (direct minus inferred), in percentage points, for ensemble risk ($\Delta R$) and human query rate ($\Delta Q$).

| | 2.5% target | | 5% target | | 7.5% target | | 10% target | |
|---|---|---|---|---|---|---|---|---|
| Method | $\Delta R$ | $\Delta Q$ | $\Delta R$ | $\Delta Q$ | $\Delta R$ | $\Delta Q$ | $\Delta R$ | $\Delta Q$ |
| SE | $-0.5$ | $+28.0$ | $-1.3$ | $+22.0$ | $+0.1$ | $+11.2$ | $+0.7$ | $+3.7$ |
| PPI+SE | $-0.2$ | $+11.8$ | $-0.4$ | $+17.1$ | $+0.6$ | $+7.1$ | $+0.3$ | $+9.6$ |
| GPC+SE | $+0.1$ | $-1.2$ | $-0.2$ | $+1.1$ | $-0.3$ | $+0.4$ | $+0.1$ | $-0.7$ |
| static CARE | $-0.1$ | $+0.3$ | $+0.0$ | $-0.7$ | $-0.1$ | $-0.6$ | $-0.2$ | $-0.3$ |
| CARE with PPI | $+0.2$ | $+14.4$ | $-0.6$ | $+5.6$ | $-0.5$ | $+2.1$ | $+0.0$ | $-3.0$ |
| CARE | $+0.1$ | $-0.7$ | $+0.9$ | $-3.1$ | $+0.2$ | $-1.5$ | $+0.1$ | $-0.4$ |

inputs and initial AI decisions fixed. Each trial processes all 1,437 queries from 722 distinct images, grouped within calibration folds.

**Feature space.** The rectifier receives 13,312 image-and-text features from decoder and vision hidden states, reduced to 64 principal components. These coordinates and the signed Yes-minus-No logit margin contribute 75% and 25% of normalized feature energy, respectively. Representations are fitted without labels on each trial's first 20% of covariates, frozen, and shared across methods and confidence measures.

**Risk–cost tradeoffs.** Figure 11 show the tradeoff across all four risk targets and four preparation budgets under both confidence measures. At the 5% target and 20% static preparation with inferred confidence, CARE achieves 4.76% ensemble risk with 50.8% queries, compared with 90.5% for SE, 95.3% for PPI+SE, and 60.3% for static CARE. Mean CARE risk is below every tested target under both confidence measures. Larger static preparation budgets do not always reduce total query cost: static CARE uses 75.4%, 60.3%, 63.3%, and 66.3% queries across the four budgets at the 5% target with inferred confidence. Adaptive CARE avoids choosing a fixed preparation size.




(a) Inferred confidence. (b) Direct confidence. (c) Queries vs preparation. (d) Risk vs preparation.

Figure 11: ASIMOV risk–cost tradeoffs. (a, b) Targets of 5%, 10%, 15%, and 20%, with 20% static preparation. (c, d) 10%, 20%, 30%, and 40% preparation at the 5% target with inferred confidence; the dotted horizontal line is the nominal ensemble-risk target.

**Adaptive learning.** Figure 12 presents adaptive learning through model improvement, escalation cost, and final-ensemble error. After 500 samples, GP-based CARE queries 47.4% and has 21.4% rolling model risk, compared with 65.6% and 23.1% for CARE with PPI. Both rectifiers improve with feedback and reach similar model risk by the end of the stream (18.7% for GP and 18.5% for PPI). Learning useful corrections earlier lowers GP CARE's average query rate over the full stream to 50.8%, compared with 57.9% for CARE with PPI.

**Robustness to confidence choice.** At the 5% target, replacing inferred with direct confidence changes CARE's query rate by only $-1.5$ percentage points (50.8% to 49.3%), with mean ensemble risks of 4.76% and 4.88%, respectively. For CARE with PPI, the same change increases queries by 28.3 points (57.9% to 86.2%); SE changes by 2.2 points at 20% preparation. As in iSafetyBench, rectifying confidence together with predictions reduces reliance on the original confidence measure.

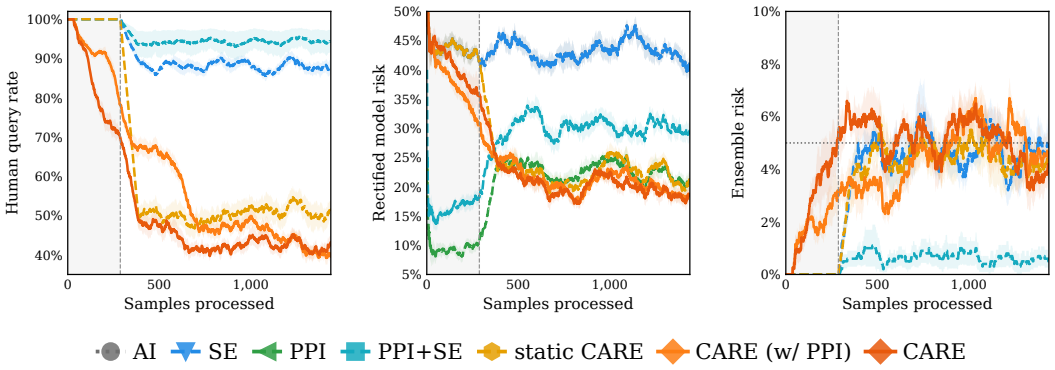



Figure 12: ASIMOV learning trajectories at the 5% target and 20% static preparation with inferred confidence. Lines average trailing windows of up to 100 arrivals over ten trials. Gray shading marks the 287-arrival static preparation period, where static model risks can include in-sample predictions. AI and PPI are shown as uncalibrated references in the model-risk panel. The dotted horizontal line is the nominal ensemble-risk target. Some initial model-risk transients are outside the plotted range.

## B RISK DECOMPOSITION

**Proposition 1** (Risk decomposition). *Let $p_H(x) = P(Y = 1 \mid X = x, H)$ be the human-aligned conditional probability, induced by human preference $H$. For a random training dataset $D$ drawn independently (conditioned on $H$) from the same distribution as $(X, Y)$, namely $P_X P_{Y|X,H}$, where $P_{Y|X,H} = \mathrm{Bern}(p_H(X))$, let $p_D = \mathbb{E}_H[p_H \mid D]$ be the training-conditional learned predictor, and let $p_B = \mathbb{E}_D[p_D]$ be the training-marginal counterpart. Then, we have*

$$R_{2,x}(p_A) = \mathbb{E}_Y[(p_A(X) - Y)^2 \mid X = x]$$
$$= \underbrace{\mathbb{E}_H[p_H(x)(1 - p_H(x))]}_{\textit{aleatoric}} + \underbrace{\mathbb{E}_{D,H}[(p_H(x) - p_D(x))^2]}_{\textit{excess}} + \underbrace{\mathbb{E}_D[(p_D(x) - p_B(x))^2]}_{\textit{epistemic}} + \underbrace{(p_B(x) - p_A(x))^2}_{\textit{bias}}.$$

*Proof.* By the tower property,

$$R_{2,x}(p_A) = \mathbb{E}_Y[(p_A(X) - Y)^2 \mid X = x]$$
$$= \mathbb{E}_H[\mathbb{E}_Y[(Y - p_H(X) + p_H(X) - p_A(x))^2 \mid X = x, H]]$$
$$= \mathbb{E}_H[p_H(x)(1 - p_H(x)) + (p_H(x) - p_A(x))^2] \tag{7}$$
$$= \mathbb{E}_H[p_H(x)(1 - p_H(x))] + \mathbb{E}_D[\mathbb{E}_H[(p_H(x) - p_D(x) + p_D(x) - p_A(x))^2 \mid D]]$$
$$= \mathbb{E}_H[p_H(x)(1 - p_H(x))] + \mathbb{E}_{D,H}[(p_H(x) - p_D(x))^2 + (p_D(x) - p_A(x))^2] \tag{8}$$
$$= \mathbb{E}_H[p_H(x)(1 - p_H(x))] + \mathbb{E}_{D,H}[(p_H(x) - p_D(x))^2] + \mathbb{E}_D[(p_D(x) - p_B(x) + p_B(x) - p_A(x))^2]$$
$$= \mathbb{E}_H[p_H(x)(1 - p_H(x))] + \mathbb{E}_{D,H}[(p_H(x) - p_D(x))^2] + \mathbb{E}_D[(p_D(x) - p_B(x))^2] + (p_B(x) - p_A(x))^2, \tag{9}$$

where the cross terms in (7–9) vanish, respectively, because $0 = p_H(x) - \mathbb{E}_Y[Y \mid X = x, H] = p_D(x) - \mathbb{E}_H[p_H(x) \mid D] = p_B(x) - \mathbb{E}_D[p_D(x)]$. □

**Example 1** (Linear regression). Suppose $p_H(x) = x^T(\beta + \epsilon_H(x))$, where $\epsilon_H(x)$ has zero mean and covariance $\Sigma_x$, and the AI model is null $p_A \equiv 0$. Consider perfect training-conditional and training-marginal learners recovering $\beta_D(x) := \mathbb{E}_H[\beta + \epsilon_H(x) \mid D]$ and $\beta_B := \mathbb{E}_D[\beta_D] = \mathbb{E}_H[\beta + \epsilon_H(x)] = \beta$. Let $p_D(x) = x^T\beta_D(x)$ and $p_B(x) = x^T\beta_B$ be their corresponding predictors. By the law of total variance,

$$\mathbb{E}[(p_H(x) - p_B(x))^2] = x^T \mathrm{Var}(\epsilon_H)x = x^T\mathbb{E}[\mathrm{Var}(\epsilon_H(x) \mid D)]x + x^T \mathrm{Var}(\mathbb{E}[\epsilon_H(x) \mid D])x$$
$$= \mathbb{E}[(p_H(x) - p_D(x))^2] + x^T \mathrm{Var}(\mathbb{E}[\epsilon_H(x) \mid D])x \geq \mathbb{E}[(p_H(x) - p_D(x))^2].$$

When $\epsilon_H$ is homoskedastic and independent, $\mathbb{E}[\epsilon_H(x) \mid D] = 0$, and thus the two predictors have the same Brier risk; when it's not, the training-conditional predictor $p_D$ can be strictly better by leveraging the richer information set $D$ to adapt to the local uncertainty structure.

# C Extended discussion of rectification

## C.1 Prior regression

At an offline checkpoint, write the available labeled data as $D = \{(X_i, Y_i)\}_{i=1}^n$, let $\mathbf{X}$ have rows $X_i^T$, and let $\mathbf{r} = (r(X_1), \dots, r(X_n))^T$. The prior in Section 3 gives

$$\mathbf{r} \sim \mathcal{N}(\mathbf{X}\beta, K), \qquad K = \tau^2\Sigma, \qquad \Sigma = \rho K_\phi + (1-\rho)K_A,$$

where $(K_\phi)_{ij} = k_\phi(X_i, X_j)$ and $(K_A)_{ij} = k_A(X_i, X_j)$, with $\tau > 0$ and $\rho \in [0, 1]$.

Conditional on the residual process, the labels are independent. By link symmetry, their log-likelihood contributions are $\ell_i(r) = \log g^{-1}((2Y_i - 1)(g(p_A(X_i)) + r))$. We fit $(\beta, \phi, \tau, \rho)$ by maximizing the marginal likelihood

$$\Lambda(\beta, \phi, \tau, \rho) = \int_{\mathbb{R}^n} \exp\left(\sum_{i=1}^n \ell_i(r_i)\right) \mathcal{N}(\mathbf{r}; \mathbf{X}\beta, K)\, \mathrm{d}\mathbf{r}.$$

The resulting posterior is

$$p(\mathbf{r} \mid D) = \Lambda^{-1} \exp\left(\sum_{i=1}^n \ell_i(r_i)\right) \mathcal{N}(\mathbf{r}; \mathbf{X}\beta, K). \tag{10}$$

Let $\mathbf{m}$ and $V$ denote its posterior mean vector and covariance matrix, and set $V_\beta = V + (\mathbf{m} - \mathbf{X}\beta)(\mathbf{m} - \mathbf{X}\beta)^T$. Differentiating the marginal likelihood under the integral gives

$$\nabla_\beta \log \Lambda = \mathbf{X}^T K^{-1}(\mathbf{m} - \mathbf{X}\beta),$$
$$\nabla_\alpha \log \Lambda = \frac{1}{2} \operatorname{trace}\left((K^{-1}V_\beta K^{-1} - K^{-1})\nabla_\alpha K\right), \qquad \alpha \in \{\tau^2, \rho, \phi_j\}.$$

In particular, setting the mean and amplitude derivatives to zero yields

$$\beta = (\mathbf{X}^T\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^T\Sigma^{-1}\mathbf{m}, \qquad \tau^2 = \frac{1}{n}\operatorname{trace}(\Sigma^{-1}V_\beta). \tag{11}$$

System (11) consists of coupled stationarity equations, as the posterior moments depend on the prior parameters. This gives a practical expectation–maximization (EM) procedure. The E-step computes $(\mathbf{m}, V)$ under the current prior. Holding these moments fixed, the M-step increases the expected Gaussian log prior $-\frac{1}{2}(\log|K| + \operatorname{trace}(K^{-1}V_\beta))$. Fixing $(\phi, \rho)$, update $\beta$ and then $\tau^2$ by (11), recomputing $V_\beta$ after updating $\beta$. Then update $\rho \in [0, 1]$ and $\phi$ numerically using $\nabla_\rho K = \tau^2(K_\phi - K_A)$ and $\nabla_\phi K = \tau^2\rho\nabla_\phi K_\phi$.

## C.2 Posterior inference

Fix the fitted prior, abbreviate its kernel as $k = k_{\tau,\rho,\phi}$, and write $k(x, \mathbf{X}) = (k(x, X_1), \dots, k(x, X_n))$. Gaussian conditioning gives

$$r(x) \mid \mathbf{r} \sim \mathcal{N}\Big(x^T\beta + k(x, \mathbf{X})K^{-1}(\mathbf{r} - \mathbf{X}\beta), k(x, x) - k(x, \mathbf{X})K^{-1}k(\mathbf{X}, x)\Big).$$

Integrating over (10) gives the (non-Gaussian) posterior process, whose mean and covariance follow from total expectation and covariance:

$$m_t(x) = x^T\beta + k(x, \mathbf{X})K^{-1}(\mathbf{m} - \mathbf{X}\beta), \tag{12}$$
$$k_t(x, x') = k(x, x') - k(x, \mathbf{X})K^{-1}k(\mathbf{X}, x') + k(x, \mathbf{X})K^{-1}VK^{-1}k(\mathbf{X}, x'). \tag{13}$$

On the labeled inputs, $\mathbf{m} = m_t(\mathbf{X})$ and $V = k_t(\mathbf{X}, \mathbf{X})$; $K$ remains the prior covariance matrix. The formulas are exact when $(\mathbf{m}, V)$ are exact posterior moments. For implementation, we approximate the latent posterior by $\mathcal{N}(\mathbf{m}, V)$ using EP or Laplace moments below; the same formulas then define the approximate process $R_t \sim \mathrm{GP}(m_t, k_t)$.

Under this Gaussian approximation, the rectified prediction is the posterior mean:

$$p_t(x) = \int_{\mathbb{R}} g^{-1}(g(p_A(x)) + r)\mathcal{N}(r; m_t(x), k_t(x, x))\, \mathrm{d}r. \tag{14}$$

For a general link this is a one-dimensional integral. For the probit link $g = \Phi^{-1}$, introduce an independent $Z \sim \mathcal{N}(0,1)$. We have $Z - R_t(x) \sim \mathcal{N}(-m_t(x), 1 + k_t(x,x))$ and thus

$$p_t(x) = \mathbb{P}(Z - R_t(x) \le g(p_A(x))) = \Phi\left(\frac{g(p_A(x)) + m_t(x)}{\sqrt{1 + k_t(x,x)}}\right), \tag{15}$$

### C.3 EP AND ADF APPROXIMATION

**Offline: expectation propagation.** EP approximates each likelihood factor by a Gaussian site (Minka, 2013):

$$\exp(\ell_i(r_i)) \approx \widetilde{t}_i(r_i) \propto \exp\Big(-\tfrac{1}{2}\widetilde{\lambda}_i r_i^2 + \widetilde{\nu}_i r_i\Big).$$

Multiplying the sites by the prior gives a Gaussian approximation with

$$V^{-1} = K^{-1} + \mathrm{diag}(\widetilde{\boldsymbol{\lambda}}), \qquad \mathbf{m} = V(K^{-1}\mathbf{X}\beta + \widetilde{\boldsymbol{\nu}}).$$

To update site $i$, remove it from the current marginal to form the Gaussian cavity $\mathcal{N}(b, v)$, where

$$v = (V_{ii}^{-1} - \widetilde{\lambda}_i)^{-1}, \qquad b = v(m_i/V_{ii} - \widetilde{\nu}_i).$$

Multiply this cavity by $\exp(\ell_i(r))$ and compute the normalized tilted mean $\tilde{b}$ and variance $\tilde{v}$. Moment matching gives the new site parameters

$$\widetilde{\lambda}_i = \tilde{v}^{-1} - v^{-1}, \qquad \widetilde{\nu}_i = \tilde{b}/\tilde{v} - b/v.$$

Recompute the Gaussian approximation and revisit sites until the updates stabilize.

**Online: assumed-density filtering.** ADF processes each new active label once. Let the current approximate residual process be $R_t \sim \mathrm{GP}(m_t, k_t)$.[2] After an escalated query $(X_t, Y_t)$, set $b = m_t(X_t)$, $v = k_t(X_t, X_t)$, form the one-dimensional tilted marginal $\tilde{R} \propto g^{-1}((2Y_t - 1)(g(p_A(X_t)) + \tilde{R}))\mathcal{N}(\tilde{R}; b, v)$, and calculate $\tilde{b} = \mathbb{E}[\tilde{R}]$ and $\tilde{v} = \mathrm{Var}(\tilde{R})$. For $v > 0$, Gaussian conditioning gives

$$\mathbb{E}[R_t(x) \mid R_t(X_t) = r] = m_t(x) + v^{-1}k_t(x, X_t)(r - b),$$
$$\mathrm{Cov}(R_t(x), R_t(x') \mid R_t(X_t) = r) = k_t(x, x') - v^{-1}k_t(x, X_t)k_t(X_t, x').$$

ADF matches the first two moments of the Gaussian approximation of $R_{t+1}$ to the above conditional moments using the laws of total expectation and covariance, giving the functional update:

$$m_{t+1}(x) = m_t(x) + v^{-1}k_t(x, X_t)(\tilde{b} - b), \tag{16}$$
$$k_{t+1}(x, x') = k_t(x, x') + v^{-2}k_t(x, X_t)k_t(X_t, x')(\tilde{v} - v). \tag{17}$$

If $v = 0$, positive semidefiniteness implies $k_t(x, X_t) = 0$ for every $x$, and the process is unchanged.

Each update requires only a one-dimensional integral for the tilted marginal and a rank-one covariance correction.

**Closed-form probit updates.** Differentiating the Gaussian density in tilted marginal with respect to $b$ yields

$$\tilde{b} = b + v\,\partial_b \log N_t, \qquad \tilde{v} = v + v^2\,\partial_b^2 \log N_t. \tag{18}$$

Indeed, $\partial_b \log \mathcal{N}(r; b, v) = (r - b)/v$, so $\partial_b \log N_t = (\tilde{b} - b)/v$ and a second differentiation gives $\partial_b^2 \log N_t = (\tilde{v} - v)/v^2$. For probit link $g = \Phi^{-1}$, let

$$\zeta = \frac{(2Y_t - 1)(g(p_A(X_t)) + b)}{\sqrt{1 + v}}, \qquad a(u) = \frac{\varphi(u)}{\Phi(u)}, \tag{19}$$

where $\varphi$ is the standard normal density. The same Gaussian convolution as in (15) gives $N_t = \Phi(\zeta)$. Since $a'(u) = -a(u)(u + a(u))$, (18) gives

$$\tilde{b} = b + \frac{(2Y_t - 1)v}{\sqrt{1 + v}}a(\zeta), \qquad \tilde{v} = v - \frac{v^2}{1 + v}a(\zeta)(\zeta + a(\zeta)).$$

[2] As will be clear from the context, $R_t$ denotes both the marginal rectifier for the next (test) input $X$ and the approximate residual process on the entire input space $\mathcal{X}$.

Substitution into (16) and (17) gives the explicit functional updates

$$m_{t+1}(x) = m_t(x) + \frac{(2Y_t - 1)k_t(x, X_t)}{\sqrt{1+v}} a(\zeta),$$
$$k_{t+1}(x, x') = k_t(x, x') - \frac{k_t(x, X_t)k_t(X_t, x')}{1+v} a(\zeta)(\zeta + a(\zeta)).$$

### C.4 Laplace and MAP approximation

**Offline: Laplace approximation.** Laplace approximation uses a mode and local curvature rather than matching posterior moments to determine the Gaussian approximation. Specifically, it replaces the log posterior by its quadratic expansion at the maximum a posteriori (MAP) residual vector (Rasmussen & Williams, 2008):

$$\widehat{\mathbf{r}}^{\mathrm{MAP}} = \operatorname*{argmax}_{\mathbf{r}} \left\{ \sum_{i=1}^{n} \ell_i(r_i) - \frac{1}{2}(\mathbf{r} - \mathbf{X}\beta)^T K^{-1}(\mathbf{r} - \mathbf{X}\beta) \right\}.$$

The gradient is $(\ell_i'(r_i))_{i=1}^n - K^{-1}(\mathbf{r} - \mathbf{X}\beta)$, and the negative Hessian is $K^{-1} + \mathrm{diag}(-\ell_i''(r_i))$, giving Newton iterations for the mode. At the mode, the Gaussian approximation has

$$\mathbf{m} = \widehat{\mathbf{r}}^{\mathrm{MAP}}, \qquad V = (K^{-1} + \mathrm{diag}(-\ell_i''(\widehat{r}_i^{\mathrm{MAP}})))^{-1}.$$

Substituting these into (12) and (13) gives the approximate posterior process. Predictions still integrate over this Gaussian process as in (14).

**Online: Newton step for maximum a posteriori.** The Gaussian log-density $\log \mathcal{N}(r; b, v)$ is already quadratic. Expand the log-likelihood at $r = b$:

$$\ell_t(r) \approx \ell_t(b) + \ell_t'(b)(r - b) + \tfrac{1}{2}\ell_t''(b)(r - b)^2$$

Set $d = \ell_t'(b)$ and $w = -\ell_t''(b)$. The local expansion gives a Gaussian tilted approximation with mean $b + vd/(1 + vw)$ and variance $v/(1 + vw)$. Substituting these for $(\tilde{b}, \tilde{v})$ in (16) and (17) gives

$$m_{t+1}(x) = m_t(x) + \frac{k_t(x, X_t)d}{1 + vw}, \tag{20}$$

$$k_{t+1}(x, x') = k_t(x, x') - \frac{w\, k_t(x, X_t)k_t(X_t, x')}{1 + vw}. \tag{21}$$

**Closed-form logistic and probit updates.** For the logit link, let $p = g^{-1}(g(p_A(X_t)) + b)$. Direct differentiation gives $d = Y_t - p$ and $w = p(1 - p)$, which yields the usual recursive logistic Newton update through (20) and (21). For the probit link, use the ratio $a$ from (19) and set $\zeta = (2Y_t - 1)(g(p_A(X_t)) + b)$. Differentiating $\ell_t(b) = \log \Phi(\zeta)$ gives $d = (2Y_t - 1)\, a(\zeta)$ and $w = a(\zeta)(\zeta + a(\zeta))$. Here $\zeta$ has no factor $\sqrt{1+v}$ because we evaluate the likelihood derivatives at $b$.

## D Extended analysis of calibration

**Initial: confidence-based nonconformity score.** It's natural to use the confidence measure $c_t$ (provided by the AI model and rectified by the rectifier) to construct our consistency check, referred to as a *nonconformity score*. Formally, for a (rectified) model consisting of a classifier and a confidence function $\mathsf{M}_t = (f_t, c_t)$, its *decision set* is given by (1):

$$\mathcal{S}(\lambda; \mathsf{M}_t, x) := \begin{cases} \{f_t(x)\}, & \text{if } c_t(x) > \lambda(x); \\ \mathcal{Y}, & \text{o.w.} \end{cases}$$

That is, when the decision set is a singleton, the model is confident enough to make a decision; otherwise, it is ambiguous and requires human review. The corresponding nonconformity score is the generalized inverse of this decision set:

$$s(x, y; \mathsf{M}_t) := \inf\{\lambda \in [0, 1] : y \in \mathcal{S}(\lambda; \mathsf{M}_t, x)\} = c_t(x)\mathbf{1}\{f_t(x) \neq y\},$$

which is zero if $y$ agrees with the prediction and equals the model confidence when it disagrees. Intuitively, when $y$ is the true label, a confident error receives a higher score than an uncertain one. For $c_t(x) = \max\{p_t(x), 1 - p_t(x)\}$ in binary classification, this recovers the prediction-based score of Romano et al. (2020). Our construction is tailored to our escalation rule and accommodates model-provided confidence measures.

### STEP 1. AUGMENTED PARTICLES

We first reformulate our method more concisely as follows. Fix $t_0 \le t$, with the last offline checkpoint preceding arrival $X_{t_0}$. Let $D_0$ collect data available at that checkpoint and $D_{t_0:t-1}$ collect actively escalated data from the online stream at times $t_0, \dots, t-1$. For each particle, let $D^{(k)}$ be the corresponding left-out data set. For simplicity, we suppose all left-out data sets have the same size $m$. Let $D^{(K+1)}$ augment the incoming test input $X_t$ with the test label $y$ plus another $m-1$ data points in $D_0$ to match the size of the left-out data sets.[3] These left-out datasets should be chosen by a fixed rule independent of the data values. When $m = 1$, we recover the leave-one-out setting. Let $D^{-(k,k')} := (D_0 \cup D^{(K+1)}) \setminus (D^{(k)} \cup D^{(k')})$, and define augmented model particles as

$$\mathsf{M}^{-(k,k')} = \mathsf{M}_{\mathsf{on}}\left(\mathsf{M}\left(D^{-(k,k')}\right), D_{t_0:t-1}\right), \quad k, k' \in [K+1].$$

The $k$-th actual model particle is then $\mathsf{M}^{(k)} = \mathsf{M}^{-(k,K+1)}$. For each left-out dataset, we designate a representative data point $z^{(k)}$ by a fixed slot rule, and designate $z^{(K+1)} := (X_t, y)$ for $D^{(K+1)}$. We associate each particle with the nonconformity scores evaluated on their designated data points:

$$s^{-(k,k')} = s(\mathsf{M}^{-(k,k')}, z^{(k)}), \quad k \neq k' \in [K+1],$$

or $s^{-(k,k')} = \infty$ when $k = k'$. Note that the actual nonconformity score of the $k$-th particle at the test input is $s(\mathsf{M}^{(k)}, (X_t, y)) = s^{-(K+1,k)}$.

### STEP 2. WEIGHTED CONFORMITY

When all left-out datasets are exchangeable and the learning method is permutation-invariant, the scores $s^{-(k,k')}$ are equivariant and thus conformal with the test score $s^{-(K+1,k)}$. However, in our setting, the left-out datasets are drawn from different distributions due to the adaptive data stream, and thus they have different impact on the model particles. We thus need to weigh their conformity w.r.t. the test score.

To do so, we need to delineate two things each data point specifies: a time slot and a realized sample value. Then, we consider the following conditioning: $K+1$ groups of left-out time slots are realized, but the mapping from each slot group to the sample group is not revealed. Under this conditioning, each sample group has some probability of occupying the test slot group, and thus the scores should be weighed by the corresponding probability.

Formally, define the following conditioning information $\mathcal{D}$: the full redacted arrival slot log which records the escalation decisions is known; the grouped slots of all left-out datasets are known; all collected samples are observed; the slot-to-sample mapping of slots outside the union of the left-out datasets, including the online slots, is known; and the grouping and within-group order of the left-out samples, including the test point, are known. The only thing we mask—the mapping from left-out slot groups to left-out sample groups—is then an element of the $(K+1)$-permutation group $S_{K+1}$. We can further augment $\mathcal{D}$ by restricting the mapping to be in a transitive subgroup $G$ of $S_{K+1}$, e.g., the $(K+1)$-cyclic group. Thus, the remaining assignments are exactly those obtained by applying $G$ to the realized assignment. Our results hold for any permutation subgroup. A smaller group reduces the number of permutations to consider, hence making likelihood calculation easier; however, it may produce more concentrated weights and reduce the diversity of our particle system. For example, if the subgroup is non-transitive and never maps a fold to the test slot, then the corresponding particle has zero weight and will not contribute.

Let $D^{\circ}_{t,z}$ be the observed data stream $((X_i, A_i, A_i Y_i))_{i=1}^{t-1}$ appended with the test point $z = (X_t, y)$, where unqueried labels are redacted. With a slight abuse of notation, we also denote $G$ as the extended group of permutations on the entire data stream, which is the identity on the slots outside the union of the left-out datasets, including the unqueried slots and online stream, and keeps the relative order of the slots inside each left-out dataset. An immediate result is that the permutations in $G$ do not change

[3] A simple mental model is that $\{D^{(k)}\}_{k=1}^{K+1}$ are disjoint. For example, for $K$-fold cross-conformal, we can first reserve $(n_0+1)/(K+1) - 1$ data points for test fold augmentation, and then partition the remaining data points into $K$ disjoint sets. Leave-one-out cross-conformal is another example where no test augmentation is needed. However, all of our analysis goes through without requiring the disjointness, via the multiset formalism.

our collection of model particles. Specifically, for any $\sigma \in G$ applied on the entire data stream, the new model particles are

$$\begin{aligned}
&\mathsf{M}^{-(k,k')}(\sigma(D^{\circ}_{t,z})) \\
&= \mathsf{M}_{\mathsf{on}}\left(\mathsf{M}\left(\sigma((D_0 \cup \{z\}) \setminus (D^{(\sigma^{-1}(k))} \cup D^{(\sigma^{-1}(k'))}))\right), \sigma(D_{t_0:t-1})\right) && (22)\\
&= \mathsf{M}_{\mathsf{on}}\left(\mathsf{M}\left(\sigma(D^{-(\sigma^{-1}(k),\sigma^{-1}(k'))})\right), D_{t_0:t-1}\right) && (23)\\
&= \mathsf{M}_{\mathsf{on}}\left(\mathsf{M}\left(D^{-(\sigma^{-1}(k),\sigma^{-1}(k'))}\right), D_{t_0:t-1}\right) && (24)\\
&= \mathsf{M}^{-(\sigma^{-1}(k),\sigma^{-1}(k'))}(D^{\circ}_{t,z}), \quad k,k' \in [K+1],
\end{aligned}$$

where (22) is because $\sigma$ is separable between the online stream and the rest of the data stream, and when we leave out the $k$-th batch of slots under sample permutation $\sigma$, we are actually removing the original $\sigma^{-1}(k)$-th left-out dataset; (23) is because $D_0 \cup D^{(K+1)} = D_0 \cup \{z\}$ and $\sigma$ preserves the online stream; and (24) is because $\mathsf{M}$ is permutation-invariant.

As a result, under permutation $\sigma$, the original score $s^{-(k,k')}$ moves to position $(\sigma(k), \sigma(k'))$, for any $k, k' \in [K+1]$, and thus the score array, indexed by sample groups, is fixed under $\mathcal{D}$.[4] Therefore, the probability that the $k$-th sample group supplies the test-score row is exactly its probability of occupying the test slot group under $\mathcal{D}$. This need not be true when the learning method is permutation-variant w.r.t. elements in $G$, as the scores may change under each permutation. Our method resolves this by restricting the permutation group to a subgroup $G$ under which the learning method is permutation-invariant.

### Step 3. Risk control

Let the probability of the $k$-th sample group occupying the test slot under $\mathcal{D}$ be $w^{(k)}$; note that $w^{(k)}$ depends on $y$ and is a functional of $X_t$, but here we suppress this dependence for notational simplicity. Then, the probability that group $k$ wins its comparison with the random test group is

$$\begin{aligned}
&\sum_{k'=1}^{K+1} w^{(k')} \mathbf{1}\left\{s(\mathsf{M}^{-(k,k')}, z^{(k)}) > s(\mathsf{M}^{-(k,k')}, z^{(k')})\right\} \\
&= \sum_{k'=1}^{K+1} w^{(k')} \mathbf{1}\{s^{-(k,k')} > s^{-(k',k)}\}.
\end{aligned}$$

We see that each summand puts $k'$ at the test slot and is scaled with its likelihood. Following Barber et al. (2021), we call a group *strange* if the above probability is unusually large, giving the *strange set*:

$$\mathcal{U} := \left\{k \in [K+1] : \sum_{k'=1}^{K+1} w^{(k')} \mathbf{1}\left\{s^{-(k,k')} > s^{-(k',k)}\right\} \geq 1 - \epsilon/2\right\}.$$

Recall that each score has some probability to be the test score. Thus, the probability of the test group being strange is $u = \sum_{k=1}^{K+1} w^{(k)} \mathbf{1}\{k \in \mathcal{U}\}$. We are left to show that this probability is small. By definition, we have

$$\begin{aligned}
u(1-\epsilon/2) &= \sum_{k=1}^{K+1} w^{(k)} \mathbf{1}\{k \in \mathcal{U}\} \cdot (1-\epsilon/2) \\
&\leq \sum_{k=1}^{K+1} w^{(k)} \mathbf{1}\{k \in \mathcal{U}\} \sum_{k'=1}^{K+1} w^{(k')} \mathbf{1}\{s^{-(k,k')} > s^{-(k',k)}\} \\
&= \sum_{k \in \mathcal{U}, k' \in [K+1]} w^{(k)} w^{(k')} \mathbf{1}\{s^{-(k,k')} > s^{-(k',k)}\} \\
&= \left(\sum_{k \in \mathcal{U}, k' \in \mathcal{U}} + \sum_{k \in \mathcal{U}, k' \in [K+1] \setminus \mathcal{U}}\right) w^{(k)} w^{(k')} \mathbf{1}\{s^{-(k,k')} > s^{-(k',k)}\}.
\end{aligned}$$

[4] We also require that elements of $G$ map designated data points to designated data points.

For the first summation, we have

$$\begin{aligned}
&\sum_{k\in\mathcal{U},k'\in\mathcal{U}} w^{(k)}w^{(k')}\mathbf{1}\{s^{-(k,k')} > s^{-(k',k)}\} \\
=&\frac{1}{2}\sum_{k\in\mathcal{U},k'\in\mathcal{U}} w^{(k)}w^{(k')}(\mathbf{1}\{s^{-(k,k')} > s^{-(k',k)}\} + \mathbf{1}\{s^{-(k,k')} < s^{-(k',k)}\}) \\
\leq&\frac{1}{2}\sum_{k\in\mathcal{U},k'\in\mathcal{U}} w^{(k)}w^{(k')}\left(1-\mathbf{1}\left\{k=k'\right\}\right) \\
=&\frac{1}{2}\left(u^2-\sum_{k\in\mathcal{U}}(w^{(k)})^2\right),
\end{aligned}$$

where the inequality is because there are cases when $s^{-(k,k')} = s^{-(k',k)}$ but $k \neq k'$. The second summation can be bounded by

$$\sum_{k\in\mathcal{U},k'\notin\mathcal{U}} w^{(k)}w^{(k')}\mathbf{1}\{s^{-(k,k')} > s^{-(k',k)}\} \leq \sum_{k\in\mathcal{U},k'\notin\mathcal{U}} w^{(k)}w^{(k')} = \sum_{k\in\mathcal{U}} w^{(k)}(1-u) = u(1-u).$$

Combining both gives

$$u(1-\epsilon/2) \leq \frac{1}{2}u^2 + u(1-u) = u - \frac{1}{2}u^2 \implies u \leq \epsilon.$$

Connecting back to our prediction set

$$\mathcal{C}_{t,\epsilon}(x) = \left\{y \in \mathcal{Y} : \sum_{k=1}^{K} w^{(k)}(x,y)\mathbf{1}\left\{s(\mathsf{M}^{(k)}, z^{(k)}) < s(\mathsf{M}^{(k)}, (x,y))\right\} < 1-\epsilon/2\right\}$$

gives

$$P(Y_t \notin \mathcal{C}_{t,\epsilon}(X_t)) = \mathbb{E}[u] \leq \epsilon.$$

STEP 4. COUNTERFACTUAL LIKELIHOOD

We are left to calculate the weight functional $w^{(k)}$ under $\mathcal{D}$. Writing $X_i^\sigma = X_{\sigma^{-1}(i)}$, we have

$$\begin{aligned}
w^{(k)}(x,y) &= \frac{\sum_{\sigma\in G:\sigma(k)=K+1}\Lambda_t^\sigma(x,y)}{\sum_{\sigma\in G}\Lambda_t^\sigma(x,y)}, \qquad (25)\\
\text{where}\quad \Lambda_t^\sigma(x,y) &= \prod_{i=1}^{t-1}\ell(X_i^\sigma, A_i \,|\, \mathcal{F}_{i-1}^\sigma), \\
\text{where}\quad \ell(x,a\,|\,\mathcal{F}) &= \pi(x\,|\,\mathcal{F})^a(1-\pi(x\,|\,\mathcal{F}))^{1-a},
\end{aligned}$$

where $(\mathcal{F}_i^\sigma)_{i=0}^{t-1}$ is the observed filtration generated by replaying the learning and acquisition algorithms on the counterfactual stream $\sigma(D_{t,z}^\circ)$; $\pi(x\,|\,\mathcal{F})$ is the algorithm's probability to escalate $x$ given the filtration $\mathcal{F}$;[5] and (25) follows because the i.i.d. covariate and conditional-label density factors in the unconditioned stream likelihood are invariant under $G$ and cancel upon conditioning on $\mathcal{D}$. The product includes unqueried arrivals but no acquisition factor for the test point.

We see that to calculate the weight functionals, we have to replay the learning algorithm on each counterfactual stream $\sigma(D_{t,z}^\circ)$ for all $\sigma \in G$. For the online period, we have a simplifying factoriza-

[5] In this section, $A$ denotes the general escalation indicator, and $\pi$ its corresponding escalation probability.

tion:

$$
\begin{aligned}
w^{(k)}(x,y) &\propto \sum_{\sigma\in G:\sigma(k)=K+1} \Lambda^{\sigma}_{t_0}(x,y)\cdot\prod_{j=t_0}^{t-1}\ell(X_j,A_j\,|\,\mathcal{F}^{\sigma}_{t_0-1},(Z_{j'})_{j'=t_0}^{j-1})\\
&=\left(\sum_{\sigma\in G:\sigma(k)=K+1}\Lambda^{\sigma}_{t_0}(x,y)\right)\cdot\prod_{j=t_0}^{t-1}\ell(X_j,A_j\,|\,\mathcal{F}^{(k)}_{t_0-1},(Z_{j'})_{j'=t_0}^{j-1})\\
&\propto w^{(k)}_{t_0}(x,y)\cdot\prod_{j=t_0}^{t-1}\ell(X_j,A_j\,|\,\mathcal{F}^{(k)}_{t_0-1},(Z_{j'})_{j'=t_0}^{j-1}),
\end{aligned}
$$

where $\mathcal{F}^{(k)}_{t_0-1}$ denotes the counterfactual history obtained by switching only the designated data point $z^{(k)}$ in the $k$-th left-out dataset with the test point $z$ and no other permutation. We use the fact that permutations in $\{\sigma\in G:\sigma(k)=K+1\}$ generate the same set of offline data points $(D_0\cup\{z\})\setminus\{z^{(k)}\}$, and that the online period is order-invariant with respect to the offline dataset. Therefore, after obtaining the weights $w^{(k)}_{t_0}$ after an offline checkpoint, we only need to co-evolve the particles to maintain $\mathcal{F}^{(k)}_{t-1}$ without doing any replay, enabling efficient online adaptation of the weights. Again, this does not hold if either the offline learning is order-variant or the permutation changes the online stream.

# E Extended analysis of safety escalation

## E.1 Proof of Corollary 1

The risk of our ensemble in (1) reads

$$
\begin{aligned}
R_{01}(\hat{Y}_t) =& P(\hat{Y}_t\neq Y_t)\\
=& P((\hat{Y}_t\neq Y_t)\wedge(A_t\vee E_t))+P((\hat{Y}_t\neq Y_t)\wedge\neg(A_t\vee E_t))\\
=& P(A_t\vee E_t)P(Y_t\neq Y_t\,|\,A_t\vee E_t)+P((\hat{Y}_t\neq Y_t)\wedge\neg A_t\wedge\neg E_t)\\
\leq& P((\hat{Y}_t\neq Y_t)\wedge\neg E_t)\\
\leq& P((1-f_t(X_t)=Y_t)\wedge(c_t(X_t)>\lambda_t(X_t))).
\end{aligned}
\tag{26}
$$

Therefore, we only need to bound the probability of the true label being the opposite label when we accept the current prediction. By definition (6),

$$
\begin{aligned}
& c_t(X_t)>\lambda_t(X_t)\\
\iff & c_t(X_t)-\gamma^{c}_t(X_t)>\text{Quantile}_{\frac{1-\epsilon/2}{W_t(x)}}\left(\sum_{k\in\mathcal{W}_t(x)}\frac{w^{(k)}_t(x)}{W_t(x)}\delta\{s^{(k)}_t\}\right)\\
\implies & \min_{k\in\mathcal{W}_t(X_t)}c^{(k)}_t(X_t)>\text{Quantile}_{\frac{1-\epsilon/2}{W_t(x)}}\left(\sum_{k\in\mathcal{W}_t(x)}\frac{w^{(k)}_t(x)}{W_t(x)}\delta\{s^{(k)}_t\}\right)\\
\iff & \min_{k\in\mathcal{W}_t(X_t)}c^{(k)}_t(X_t)>\text{Quantile}_{1-\epsilon/2}\left(\sum_{k=1}^{K}w^{(k)}_t\delta\{\iota_k s^{(k)}_t\}\right),
\end{aligned}
$$

where we use shorthand $s^{(k)}_t:=s^{(k)}_t(z^{(k)})$ and $\iota_k$ is the characteristic function that equals 1 if $k\in\mathcal{W}_t(X_t)$ and $+\infty$ otherwise. Further, this condition is equivalent to

$$
\begin{aligned}
&\sum_{k\in\mathcal{W}_t(X_t)}w^{(k)}_t\mathbf{1}\left\{\min_{k'\in\mathcal{W}_t(X_t)}c^{(k')}_t(X_t)>s^{(k)}_t\right\}\geq 1-\epsilon/2\\
\implies &\sum_{k\in\mathcal{W}_t(X_t)}w^{(k)}_t\mathbf{1}\left\{c^{(k)}_t(X_t)>s^{(k)}_t\right\}\geq 1-\epsilon/2.
\end{aligned}
$$

Recall that $\{w_t^{(k)}\}$ concern the opposite label $1 - f_t(X_t)$, and $f_t^{(k)}(X_t) = f_t(X_t)$ for all agreeing particles $k \in \mathcal{W}_t(X_t)$. Thus, the above condition is equivalent to

$$\sum_{k \in \mathcal{W}_t(X_t)} w_t^{(k)}(X_t, 1 - f_t(X_t)) \mathbf{1}\left\{c_t^{(k)}(X_t) > s_t^{(k)}(z^{(k)})\right\} \geq 1 - \epsilon/2$$

$$\iff \sum_{k=1}^{K} w_t^{(k)}(X_t, 1 - f_t(X_t)) \mathbf{1}\left\{c_t^{(k)}(X_t) \mathbf{1}\{f_t^{(k)}(X_t) \neq 1 - f_t(X_t)\} > s_t^{(k)}(z^{(k)})\right\} \geq 1 - \epsilon/2$$

$$\iff \sum_{k=1}^{K} w_t^{(k)}(X_t, 1 - f_t(X_t)) \mathbf{1}\left\{s_t^{(k)}(X_t, 1 - f_t(X_t)) > s_t^{(k)}(z^{(k)})\right\} \geq 1 - \epsilon/2.$$

By definition (2), the above condition is equivalent to

$$1 - f_t(X_t) \notin \mathcal{C}_{t,\epsilon}(X_t).$$

Plugging this indication back into (26) gives

$$\begin{aligned} R_{01}(\hat{Y}) \leq & P((1 - f_t(X_t) = Y_t) \wedge (1 - f_t(X_t) \notin \mathcal{C}_{t,\epsilon}(X_t))) \\ = & P((1 - f_t(X_t) = Y_t) \wedge (Y_t \notin \mathcal{C}_{t,\epsilon}(X_t))) \\ \leq & P(Y_t \notin \mathcal{C}_{t,\epsilon}(X_t)) \\ \leq & \epsilon., \end{aligned}$$

where the final inequality follows from Theorem 1.

We remark that with equal weights, excluding the opposite label indicates the inclusion of the predicted label in the prediction set $\mathcal{C}_{t,\epsilon}$. However, this is no longer true under covariate shift. Nonetheless, (26) bounded by $\epsilon$ implies that it is still safe to accept the prediction.

### E.2 Approximate weights and scores

During the offline period, calculating (4) can involve numerous policy replays, even with a simple subgroup $G$ as discussed in Appendix D. During the online period, although no counterfactual replays are needed, prioritizing computational efficiency may still call for approximation. As long as we have a high-probability bound on the approximation error, safety guarantee in Theorem 1 still holds.

Consider approximate weights $\{\hat{w}^{(k)}\}_{k=1}^K$ and scores $\{\hat{s}^{(k)}\}_{k=1}^K$. For notational simplicity, we omit the time index $t$ in this subsection. Following (6), we denote the *exact* calibrated threshold using accurate weights and scores as

$$\lambda^*(x) = \lambda(x; \{w^{(k)}\}_{k=1}^K, \{s^{(k)}\}_{k=1}^K) := \text{Quantile}_{\frac{1-\epsilon}{W(x)}}\left(\sum_{k=1}^{K} \frac{w^{(k)}(x)}{W(x)} \delta\{s^{(k)}\}\right) + \gamma^c(x).$$

Then, if we know with high probability $1 - O(\epsilon)$ that the exact $(w, e)$ is in an uncertainty set $\Omega$, we can set

$$\lambda(x) = \sup_{(w,e) \in \Omega} \lambda(x; w, e),$$

which guarantees an excess risk of $O(\epsilon)$.

We give an example with two simple approximation bounds. Suppose that, for all $k \in [K]$, we have $s^{(k)} \leq \hat{s}^{(k)} + \gamma_k^s$ for scores and $\|w - \hat{w}\|_{\text{TV}} \leq \gamma^w$ for weights. Then, we can set

$$\lambda(x) = \text{Quantile}_{1-\epsilon/2+\gamma^w}\left(\sum_{k=1}^{K} \hat{w}^{(k)}(x) \delta\{\hat{s}^{(k)} + \gamma_k^s\}\right) + \gamma^c(x),$$

which satisfies Corollary 1.

## F Extended discussion of active acquisition

We first remark that thanks to its modular design, CARE can be plugged with any acquisition strategy in the active learning/inference literature (Zrnic & Candès, 2026). For simplicity, our experiments

implement safe escalation only ($\pi_t(X_t) = \mathbf{1}\{c_t(x_t) \le \lambda_t(x_t)\}$). Below, we discuss a more general active acquisition strategy tailored to our goal of minimizing query complexity.

Let $A_t$ and $E_t$ be the active and safe escalation decisions at time $t$, respectively. Concerning short-term efficiency, $A_t$ should escalate a query that is likely to be escalated by $E_t$; concerning long-term efficiency, $A_t$ should escalate a query that minimizes the future escalation frequency of $E \vee A$. Ideally, active escalation optimizes

$$\min_{\pi_t \in [0,1]} \mathbb{E}_{A_t \sim \mathrm{Bern}(\pi_t)} \left[ A_t(1 - E_t) + \mathbb{E}\left[ \sum_{t' \ge t+1} E_{t'} \vee A_{t'} \middle| A_t \right] \right],$$

where the first term is the additional cost of active escalation at this step, and the second term is the expected future escalation cost. However, this objective is hard to evaluate as (i) the future is hard to estimate due to adaptation; (ii) $E_t$ is unknown as we need to avoid $\pi_t$ depending on $\lambda_t$; and (iii) the decision indicators are not smooth. Therefore, we consider a myopic and approximate objective:

$$\begin{aligned}
&\mathbb{E}_{A_t \sim \mathrm{Bern}(\pi_t)} \left[ A_t(1 - \hat{E}_t) + \eta \mathbb{E}\left[ \hat{E}_{t+1} \middle| A_t \right] \right] \\
=&\mathbb{E}_{A_t \sim \mathrm{Bern}(\pi_t)} \left[ A_t(1 - \mathbf{1}\{c_t(X_t) < 1 - \epsilon\}) + \eta \mathbb{E}\left[ \mathbf{1}\{\hat{c}_{t+1}(X_{t+1}) < 1 - \epsilon\} | A_t \right] \right] \\
=&\pi_t \mathbf{1}\{c_t(X_t) \ge 1 - \epsilon\} + \eta(1 - \pi_t) P_X(c_t(X) < 1 - \epsilon) \\
&+ \eta \pi_t \left( p_t P_X(c_{t+1}^{(X_t,1)}(X) < 1 - \epsilon) + \pi_t(1 - p_t) P_X(c_{t+1}^{(X_t,0)}(X) < 1 - \epsilon) \right),
\end{aligned}$$

where $p_t = p_t(X_t)$ and $X_{t+1} \sim P_X$; besides the one-step lookahead scaled by the effective horizon parameter $\eta$, we also approximate the safe escalation threshold by $1 - \epsilon$ and the label distribution of $Y_t$ by the model prediction $\mathrm{Bern}(p_t)$. Specifically, $\hat{E}_t(c) \coloneqq \mathbf{1}\{c < 1 - \epsilon\}$ and $\hat{c}_{t+1} = \mathbb{E}_{\hat{Y}_t \sim \mathrm{Bern}(p_t)}[c_{t+1}^{(X_t, \hat{Y}_t)}]$ where $c_{t+1}^{(X_t, \hat{Y}_t)}$ is the rectifier confidence at $t + 1$ after updating the rectifier with the new query $(X_t, \hat{Y}_t)$.

The functional $c \mapsto P_X(c(X) < 1 - \epsilon)$ may be hard to evaluate. Additionally, the above objective gives a hard decision $\pi_t \in \{0, 1\}$. We thus further smooth the objective, giving a *confidence-maximizing* active escalation rule

$$\pi_t = g^{-1} \circ h^{(1)}(p_t h_\epsilon^{(2)}(c_{t+1}^{(X_t,1)}) + (1 - p_t) h_\epsilon^{(2)}(c_{t+1}^{(X_t,0)}) - h_\epsilon^{(2)}(c_t) - \eta^{-1} \mathbf{1}\{c_t(X_t) \ge 1 - \epsilon\}),$$

where both $h^{(1)}$ and $h_\epsilon^{(2)}$ are strictly increasing function, with $h^{(1)}$ being affine onto $\mathbb{R}$ and $h_\epsilon^{(2)}(c)$ approximates $P_X(c(X) \ge 1 - \epsilon)$.