Trace-Aware Routing for Cost-Effective Human–AI Collaborative Labeling

Anonymous Authors[1]

Abstract

Large-scale generative AI systems, such as text-to-image models, require reliable labels to evaluate whether outputs satisfy intended specifications (e.g., prompt fidelity or rubric-based quality). In practice, AI labelers (e.g., large language model (LLM)/vision-language model (VLM) judges) are efficient but may exhibit systematic errors on subtle or fine-grained aspects of text–image alignment, whereas human labels can be more reliable but costly. This raises a natural question: How can we coordinate AI and human labelers so that only instances likely to be mislabeled are escalated to humans, under a fixed human-labeling budget? Furthermore, in many modern AI labeling workflows that rely on LLM/VLM judges, labels are produced together with an explicit reasoning trace prior to finalization, allowing early human intervention and potential savings in AI labeling computational cost. Yet, a key challenge is *when to escalate*, as intervening too late wastes AI computation and may fail to prevent incorrect labels, while intervening too early incurs unnecessary human effort. Existing deferral and routing methods typically make one-shot decisions and do not exploit trace information.

Here, we construct a trace-aware AI router for cost-effective human–AI collaboration with three key features: (i) it conditions routing decisions on the evolving reasoning trace of the AI labeler; (ii) it performs stepwise monitoring to determine the earliest point at which human review is needed; and (iii) it incorporates human budget control through a feature-based disagreement scoring model, prioritizing hard instances where AI and human judgments are more likely to differ. Empirical results across multiple baselines show that our method consistently improves labeling accuracy under fixed human budgets, demonstrating the value of reasoning traces for sequential, budget-aware routing.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

Reliable image-text labels are a key bottleneck for evaluating whether large-scale multimodal generative AI systems (e.g., text-to-image models and vision-language assistants) satisfy intended specifications such as prompt fidelity, caption correctness, or rubric-based quality. Concretely, given an image–text pair, the labeling task asks whether the image satisfies the textual specification.

Producing these labels at scale can be costly in two dimensions: human effort and AI computation. Human labels are often more reliable on subtle or fine-grained cases, but they are expensive and capacity-limited. AI labelers (e.g., large language model (LLM)/vision-language model (VLM) judges) are typically faster and cheaper per instance, yet they are not free in practice: generating a label can require substantial inference with long prompts and multi-step reasoning, leading to nontrivial token usage, latency, and serving cost (Han et al., 2025; Agrawal et al., 2024; Yao et al., 2024). Moreover, AI judges remain imperfect and can exhibit systematic failure modes on subtle attributes (e.g., counting, relations, negation, or fine-grained objects), motivating selective use of human review.

This setting raises a natural question: *How can we coordinate AI and human labelers so that only instances likely to be mislabeled are escalated to humans, under fixed budgets for both human labeling and AI computation?* Classical selective prediction and learning-to-defer frameworks address related problems by learning one-shot abstention policies that defer uncertain instances to a human labeler and otherwise rely on the model prediction (Madras et al., 2018; Mozannar & Sontag, 2020). However, these approaches typically make the deferral decision only after the AI label has been fully produced, and thus do not directly control AI inference effort.

In contrast, many modern LLM/VLM labeling pipelines produce labels together with an explicit *reasoning trace* prior to finalization. This creates two opportunities that are largely absent in one-shot deferral: First, the system can perform trace-prefix monitoring and intervene early to avoid

wasting additional AI computation. Second, the evolving trace can provide richer diagnostic signals (e.g., missing evidence, speculative claims, internal inconsistencies) that help identify likely AI mistakes, improving the quality of escalation decisions and ultimately the final (post-routing) label accuracy under the same human budget. The core challenge is therefore not only *which* instances to escalate, but also *when* to escalate during the AI labeler's stepwise reasoning: intervening too late wastes AI compute and may still allow incorrect labels, while intervening too early increases unnecessary human workload and reduces throughput.

Compared to our trace-aware, stepwise routing formulation, most existing deferral/routing methods are one-shot: they make a single abstain/defer decision after observing a fixed feature vector or a final model score (Madras et al., 2018; Mozannar & Sontag, 2020). To our knowledge, no existing method leverages the *reasoning trace of an LLM/VLM-based labeler* as a signal to decide *when* to escalate an instance to human review or to halt AI inference mid-generation. Training-free heuristics (e.g., thresholding image–text compatibility scores) similarly provide simple abstention rules but do not leverage the richer diagnostic information revealed in an AI labeler's reasoning trace (Hessel et al., 2021). Recent LLM-based selective answering and risk-rating approaches show that language models can express uncertainty and abstain, yet they typically operate as single-step policies and do not perform trace-prefix monitoring or enable early stopping during trace generation (Strong et al., 2025; Machcha et al., 2025; Mao et al., 2025). As a result, existing approaches offer limited control over *when* escalation happens and do not capitalize on the sequential evidence exposed by reasoning traces.

Our work makes three contributions:

- We formulate a sequential routing framework that monitors prefixes of an AI judge's reasoning trace and decides *when* to `ESCALATE` or `CONTINUE`. By leveraging diagnostic signals in intermediate reasoning (rather than only a final score), the router makes more accurate escalation decisions and yields more accurate *post-routing* labels under the same human-review budget; in streaming deployments, it can also early-stop the judge to avoid wasted AI computation.
- We instantiate the stepwise router as a VLM policy (optionally with retrieval-augmented in-context demonstrations for robustness across error modes) and pair it with a lightweight MLP-based disagreement predictor that estimates AI–human disagreement scores. The disagreement score prioritizes instances so that limited human effort is spent on the most error-prone cases.
- By leveraging semantically rich reasoning traces produced by VLMs, our trace-aware routing system achieves higher post-routing accuracy at fixed escalation budgets and catches a larger fraction of AI labeling mistakes than one-shot deferral heuristics and standard learning-to-defer baselines, both under i.i.d. evaluation and *distributional shift*. The stepwise policy also yields an auditable stopping point that records the trace evidence available at the moment escalation is triggered.

2. Problem setup

We consider a dataset of n independent image-text pair instances $\{(W_i, X_i)\}_{i=1}^n$, where W_i is a text prompt (such as an image caption) and X_i is the corresponding image. We write $(W, X) \sim \mathcal{P}$ for the population distribution over image-text pairs. For each pair (W_i, X_i), the evaluation question is:

"Does W_i accurately describe X_i?"

In this case, a label refers to a binary response (e.g. yes/no) that evaluates whether the text W_i accurately describes the image X_i. We denote the label space as $\mathcal{Y}$. We consider two types of labels: *AI label, and human label*.

AI label can be generated by an LLM/VLM. Specifically, for each instance (W_i, X_i), the LLM generates the following:

$$\{\boldsymbol{Y}_{AI,i} \in \mathcal{Y}, R_i\},$$

where $\boldsymbol{Y}_{AI,i}$ denotes an AI label that answers the evaluation question, and R_i is an unstructured reasoning justifying the label. Let $R_{i,\leq s}$ denote the raw text generated by the judge up to stream position s (e.g., emitted tokens or emitted text chunks). We define a deterministic *streaming* parser $\textsc{LabelStep}(\cdot)$ that extracts the completed reasoning steps available so far:

$$(r_{i,1}, \ldots, r_{i,t(s)}) \leftarrow \textsc{LabelStep}(R_{i,\leq s}),$$

where $t(s)$ is the number of completed steps detected after observing the prefix $R_{i,\leq s}$. The router is queried only when $t(s)$ increases, i.e., when a new completed step is emitted. If generation proceeds to completion, the final AI output is $(Y_{AI,i}, R_i)$, where R_i is the full raw trace. In that case,

$$(r_{i,1}, r_{i,2}, \ldots, r_{i,T_i}) \leftarrow \textsc{LabelStep}(R_i)$$

denotes the completed step sequence used for logging and offline analysis.

Human labels are generated by a human labeler who answers the same evaluation questions as the AI labeler, denoted as

$$\boldsymbol{Y}_{H,i} \in \mathcal{Y}.$$

Generally, because human labeling is costly, human labels are only obtained for an index set $I_L \subseteq \{1, \ldots, n\}$. We

write $I_U = \{1, \dots, n\} \setminus I_L$ for the remaining deployment indices. We use $\mathcal{D}_L$ and $\mathcal{D}_U$ to denote datasets, I_L and I_U for index sets. Note that in all the labeled data, we can obtain an AI label at the same time for all $i \in I_L$. Therefore, on the labeled instances $i \in I_L$, we define an AI labeling error as

$$\delta_i := \mathbf{1}\{\boldsymbol{Y}_{AI,i} \neq \boldsymbol{Y}_{H,i}\} \in \{0, 1\}.$$

Because human labels are considered as the "gold labels", $\delta_i = 1$ when there is a disagreement between the AI and the human labeler, and $\delta_i = 0$ when AI and the human labeler agree. Our goal is to design an AI router that is trained to identify, as early as possible, instances with $\delta_i = 1$.

Formally, we define a router as an online stopping policy that reads the AI labeler's reasoning one step at a time and decides whether to escalate the instance to a human reviewer. After observing the first t completed reasoning steps for instance i, the router has access to

$$H_{i,t} := (W_i, X_i, r_{i,1:t}), \qquad Z_{i,t} := \Phi_\psi(H_{i,t}).$$

Here $\Phi_\psi(\cdot)$ formats $H_{i,t}$ into router input features under design choices ψ, such as prompt formatting, demonstration selection, and retrieval strategy. We consider a family of routers indexed by $\psi \in \Psi$. Each ψ induces a policy π_ψ that maps the step-t features to a binary decision:

$$A_{i,t}^{\pi_\psi} := \pi_\psi(Z_{i,t}) \in \{0, 1\},$$

where $A_{i,t}^{\pi_\psi} = 1$ means `ESCALATE` (stop and route to human), and $A_{i,t}^{\pi_\psi} = 0$ means `CONTINUE`. The induced stopping time under π_ψ is

$$\tau_i^{\pi_\psi} := \inf\{t \in \{1, \dots, T_i\} : A_{i,t}^{\pi_\psi} = 1\},$$
$$\text{where } \tau_i^{\pi_\psi} \in \{1, \dots, T_i\} \cup \{\text{NA}\},$$

with $\tau_i^{\pi_\psi} = \text{NA}$ if the router never escalates. We denote the induced policy class by $\Pi := \{\pi_\psi : \psi \in \Psi\}$.

Let $E_i \in \{0, 1\}$ denote whether instance i is actually escalated to a human after budget enforcement. In the population-level policy formulation without finite-sample budget exhaustion, $E_i = \mathbf{1}\{\tau_i^{\pi_\psi} \neq \text{NA}\}$. The final post-routing label is

$$\widetilde{Y}_i := \begin{cases} Y_{H,i}, & E_i = 1, \\ Y_{AI,i}, & E_i = 0. \end{cases}$$

AI router trades off (i) leaving AI errors uncorrected and (ii) spending human effort. Therefore, we optimize over $\psi \in \Psi$ (equivalently, over $\pi_\psi \in \Pi$), and consider a budget-constrained objective. In the following population objective, $\delta = \mathbf{1}\{Y_{AI} \neq Y_H\}$ and τ^{π_ψ} denote the AI-error indicator and stopping time for a generic draw from the population.

$$\min_{\psi \in \Psi} \mathbb{E}[\delta\, \mathbf{1}\{\tau^{\pi_\psi} = \text{NA}\}] \quad \text{s.t.} \quad \mathbb{E}[\mathbf{1}\{\tau^{\pi_\psi} \neq \text{NA}\}] \leq \rho, \tag{1}$$

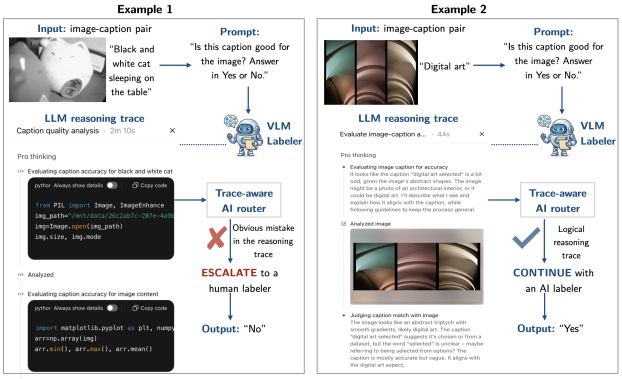

Figure 1. Trace-aware AI router illustration. Example 1 illustrates a case where the AI labeler's reasoning trace exhibits clear errors, indicating that the instance should be escalated to a human labeler. Example 2 shows a contrasting case in which the reasoning trace remains coherent and aligned with the correct label, suggesting that no human escalation is necessary.

Our primary focus is selecting ψ (prompt design and demonstration construction) to induce a policy π_ψ that decides *when* to escalate during stepwise reasoning.

3. Our proposed method

In this section, we introduce a budgeted human–AI labeling workflow that combines a disagreement predictor with a trace-aware, stepwise router. First, using a labeled set with AI-judge outputs, reasoning traces, and human gold labels, we train an MLP to estimate AI–human disagreement as an explicit measure of instance disagreement probability. Second, we design a trace-aware router that monitors progressively longer prefixes of each reasoning trace and decides whether to `ESCALATE` to a human or `CONTINUE`, with router designs selected on a held-out development split under either a hard budget or a cost-penalized objective. Third, at deployment, unlabeled instances are prioritized by predicted disagreement probability and processed online, escalating only when warranted and while budget remains, otherwise deferring to the AI label. Together, these components form an end-to-end procedure that enables early, informed escalation while enforcing system-level budget control. We summarize our method in Algorithms 1 and 2, and provide a detailed illustration below.

We begin by introducing the inputs to our algorithm. We consider a labeled dataset, defined as $\mathcal{D}_L := \{(W_i, X_i, Y_{AI,i}, R_i, Y_{H,i})\}_{i \in I_L}$, and define the unlabeled deployment dataset as $\mathcal{D}_U := \{(W_i, X_i)\}_{i \in I_U}$. We fix the human escalation budget $\rho \in (0, 1)$, interpreted as the maximum fraction of D_U that can be escalated to human labelers (equivalently, an absolute budget $B := \lfloor \rho |\mathcal{U}| \rfloor$). We define a trace pre-processor as LABELSTEP$(\cdot)$ that converts R_i into stepwise reasoning. Split $\mathcal{D}_L$ into a demonstration

Algorithm 1 Trace-Aware Budgeted AI Router

Input: $\mathcal{D}_L, \mathcal{D}_U, \rho, \Psi$. Split $\mathcal{D}_L$ into $\mathcal{D}_{\text{risk}}$, $\mathcal{D}_{\text{demo}}$ and $\mathcal{D}_{\text{dev}}$.
$f_\theta \leftarrow \textsc{FitDisagreement}\big(\{(g(W_i, X_i), \mathbf{1}\{Y_{AI,i} \neq Y_{H,i}\})\}_{i \in I_{\text{risk}}}\big)$.
$\widehat{\psi} \leftarrow \textsc{SelectRouter}(\Psi, \mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{demo}}, \rho)$.
$B \leftarrow \lfloor \rho |I_U| \rfloor$, $b \leftarrow 0$; compute $\widehat{p}_i = f_\theta(g(W_i, X_i))$ for $i \in I_U$. Let $\mathcal{O}$ sort I_U by decreasing $\widehat{p}_i$.
for all $i \in \mathcal{O}$ **do**
  $\widehat{\tau}_i \leftarrow \textsc{OnlineRouter}(W_i, X_i, \mathcal{D}_{\text{demo}}, \widehat{\psi})$ if $b < B$, otherwise NA.
  if $\widehat{\tau}_i \neq \text{NA}$ **then**
    Stop generation; obtain $Y_{H,i}$; $E_i \leftarrow 1$, $b \leftarrow b + 1$, $\widetilde{Y}_i \leftarrow Y_{H,i}$.
  else
    Complete the AI judge; obtain $Y_{AI,i}$; $E_i \leftarrow 0$, $\widetilde{Y}_i \leftarrow Y_{AI,i}$.
  end if
end for

set $\mathcal{D}_{\text{demo}}$ and a development set $\mathcal{D}_{\text{dev}}$, with corresponding index sets I_{demo} and I_{dev}. We also specify a router family $\{\pi_\psi : \psi \in \Psi\}$, where ψ indexes prompt design choices (e.g., demonstration format, retrieval method, number of demos m, instruction wording/strictness).

Step 1: Disagreement scoring on image-text instances. Our routing problem is fundamentally budgeted: we cannot send every AI-produced label to a human, so we need a principled way to prioritize which instances deserve human attention when capacity is limited. To this end, we first learn an instance-level disagreement score that estimates how likely the AI labeler's output label will diverge from a human label. This score serves two purposes: (i) it operationalizes "difficulty" as predictable AI–human disagreement rather than ad hoc heuristics, and (ii) it provides a global ranking over instances that we later use to allocate limited human effort where it has the highest expected value. Concretely, for each labeled instance $i \in I_L$ we construct a pre-routing feature representation $z_i = g(W_i, X_i)$, where $g(\cdot)$ can include vision-language embeddings, such as CLIP text/image embeddings, their similarity, and caption/image metadata. The feature map g is deliberately restricted to quantities available before running the AI judge to completion. We then fit an MLP risk model f_θ to predict the probability of disagreement,

$$\widehat{p}_i := f_\theta(z_i) \approx \mathbb{P}(\delta_i = 1 \mid z_i), \qquad i \in I_L.$$

Because $\hat{p}_i$ is trained on observed AI-human disagreements, it provides a learned estimate of which image-caption pairs are most prone to AI mistakes, enabling systematic budgeting and evaluation rather than relying on subjective "hardness" cues.

Step 2: Develop trace-aware AI router. In the second step, our core objective is to learn a router that (i) leverages the reasoning trace as incremental evidence, and (ii) can stop the AI labeler early when a mistake becomes likely, reducing wasted computation and enabling timely human correction. We implement this as a trace-aware, stepwise decision policy and tune it on a development split.

Figure 2. Example of high disagreement score and low disagreement scores.

Specifically, for each candidate router design $\psi \in \Psi$ (capturing choices such as prompt format, strictness wording, number of demonstrations m, and retrieval style), we evaluate it on I_{dev}. For each $j \in I_{\text{dev}}$, we first convert the raw trace R_j into stepwise evidence using the deterministic preprocessor LABELSTEP: $(r_{j,1:T_j}) \leftarrow \textsc{LabelStep}(R_j)$. Next, we provide *in-context calibration* by retrieving m similar labeled examples from the demonstration set and formatting them as demonstrations. Each demonstration includes $(W, X, r_{1:T})$, together with supervision indicating whether the completed AI label agreed with the human label. The router itself is queried only on the current instance state $H_{j,t} = (W_j, X_j, r_{j,1:t})$, and outputs either `ESCALATE` or `CONTINUE`. Starting at $t = 1$, the router observes $H_{j,t} = (W_j, X_j, r_{j,1:t})$ and decides whether the available evidence is already sufficient to justify human review. If it outputs `ESCALATE`, we record the stopping time $\hat{\tau}_j^\psi = t$; otherwise we advance to the next reasoning step. If escalation never occurs, we set $\hat{\tau}_j^\psi = \text{NA}$. This yields not only an escalation decision, but also an *intervention time*, which directly captures the "when to escalate" problem. The stepwise routing procedure is summarized in Algorithm 2.

On the development split, full traces are already logged. Therefore, for model selection we reconstruct the complete step sequence once via $(r_{j,1}, \ldots, r_{j,T_j}) \leftarrow \textsc{LabelStep}(R_j)$, and replay prefixes offline. This offline replay is used only for development-time evaluation of candidate router designs; deployment uses the same parser incrementally on the live generation stream.

We summarize each design ψ by three empirical quantities on $\mathcal{D}_{\text{dev}}$. First, the missed-error loss is

$$\widehat{\text{Loss}}(\psi) := \frac{1}{|I_{\text{dev}}|} \sum_{j \in I_{\text{dev}}} \delta_j \, \mathbf{1}\{\hat{\tau}_j^\psi = \text{NA}\}.$$

Algorithm 2 Online Stepwise Trace Router

Input: instance (W, X), demo pool $\mathcal{D}_{\text{demo}}$, router design ψ.
Output: stopping time $\widehat{\tau} \in \{1, \dots, T\} \cup \{\text{NA}\}$.
$\mathcal{M}_\psi \leftarrow \text{RETRIEVE}(\mathcal{D}_{\text{demo}}, W, X, \psi)$; $t \leftarrow 0$.
while LABELSTEP emits a new step r_{t+1} **do**
  $t \leftarrow t + 1$; $H_t \leftarrow (W, X, r_{1:t})$; $Z_t \leftarrow \Phi_\psi(H_t; \mathcal{M}_\psi)$.
  if $\pi_\psi(Z_t) = \text{ESCALATE}$ **then**
    return $\widehat{\tau} = t$.
  end if
end while
return $\widehat{\tau} = \text{NA}$.

Second, the normalized AI-effort proxy is

$$\widehat{\text{Effort}}(\psi) := \frac{1}{|I_{\text{dev}}|} \sum_{j \in I_{\text{dev}}} c(\widehat{\tau}_j^\psi; T_j),$$

where

$$c(\widehat{\tau}_j^\psi; T_j) = \begin{cases} \widehat{\tau}_j^\psi / T_j, & \widehat{\tau}_j^\psi \neq \text{NA}, \\ 1, & \widehat{\tau}_j^\psi = \text{NA}. \end{cases}$$

Third, the escalation rate is

$$\widehat{\text{Esc}}(\psi) := \frac{1}{|I_{\text{dev}}|} \sum_{j \in I_{\text{dev}}} \mathbf{1}\{\widehat{\tau}_j^\psi \neq \text{NA}\}.$$

We then select $\hat{\psi}$ by directly enforcing the human budget on the development set,

$$\hat{\psi} \in \arg\min_{\psi \in \Psi} \widehat{\text{Loss}}(\psi) \quad \text{s.t.} \quad \widehat{\text{Esc}}(\psi) \leq \rho, \tag{2}$$

with ties broken by smaller $\widehat{\text{Effort}}(\psi)$. This development step is what turns trace monitoring into a controlled routing policy: it ensures that the router is calibrated to the desired escalation behavior (budget) while explicitly trading off missed AI errors versus intervention cost.

Step 3: Budget-constrained deployment. At deployment time, we combine disagreement-based prioritization with online trace monitoring under a fixed human budget. For each deployment instance $i \in I_U$, we first compute a pre-routing disagreement score using only quantities available before AI completion: $z_i = g(W_i, X_i), \widehat{p}_i = f_\theta(z_i)$. We sort I_U by decreasing $\widehat{p}_i$, set $B = \lfloor \rho |I_U| \rfloor$, and initialize the budget counter $b = 0$.

We then process instances in this order. If $b = B$, the budget is exhausted, so the AI judge runs to completion and we set $E_i = 0$. If $b < B$, we stream the AI judge online and apply the selected router $\widehat{\psi}$ to each newly completed trace prefix $H_{i,t} = (W_i, X_i, r_{i,1:t})$. When the router returns ESCALATE, we stop generation, request the human label $Y_{H,i}$, set $E_i = 1$, and increment b. If the router never escalates, the AI judge completes normally, producing $Y_{AI,i}$, and we set $E_i = 0$. The final post-routing label is

$$\widetilde{Y}_i := \begin{cases} Y_{H,i}, & E_i = 1, \\ Y_{AI,i}, & E_i = 0. \end{cases}$$

Thus, the disagreement score determines which instances receive priority for the limited human budget, while the trace-aware router determines when escalation should occur for each prioritized instance.

4. Experiments

In this section, we describe the experimental setup used to evaluate trace-aware, budgeted human–AI routing for image–caption labeling and the experiment results. We first introduce the underlying image–caption benchmark and how we treat its human judgments as gold supervision, then define a suite of comparison baselines that span judge-only prompting, trace-based routing, and classical learning-to-defer risk scoring under the same escalation budget. To stress-test robustness beyond i.i.d. evaluation, we also construct topic-shifted train/test splits by clustering caption embeddings and holding out entire latent caption topics for testing, and we quantify the resulting covariate shift using both distributional statistics and embedding-space two-sample tests. Finally, we report evaluation metrics that jointly capture agreement with human labels and the quality of escalation decisions (e.g., how many cases are sent to humans, how many true AI errors are caught, and how many escalations are wasted), enabling a direct comparison of accuracy–effort tradeoffs across all methods.

4.1. Experiment setup

Dataset. We use Open Images–based Rated Image Captions (v2), a Google-released benchmark of image–caption pairs with large-scale human quality judgments (Levinboim et al., 2021). The images are randomly sampled from the Open Images Dataset (OID), while the captions are automatically generated by transformer-based captioning models trained on Conceptual Captions. Each image–caption pair is evaluated by up to 10 human raters who answer a single binary question: "Is this caption good for the image?" The dataset provides the raw counts and a binned average rating, which maps agreement into discrete rating bins. The release includes a total of 144,046 image-caption pairs plus accompanying metadata files that contain fields needed to obtain the images (e.g., URLs, licensing, and rotation information). We treat the human judgments as ground truth supervision for whether the caption matches the image, and we additionally attach an AI judge label to each pair to define an AI–human disagreement signal for routing experiments.

Benchmark methods. We compare our proposed trace-aware, budgeted routing procedure against a suite of benchmarks designed both for fair comparison and for ablation-style diagnosis of which components matter (trace conditioning, retrieval, learned risk scoring, and the choice of underlying VLM/LLM). We group baselines into the following categories.

(i) Prompting-only baselines (no routing). These baselines measure the intrinsic labeling quality of an AI judge when used as a standalone labeler, i.e., without any human intervention. We include standard zero-shot and few-shot prompting, which directly output AI labels for each image–caption pair; by construction, these methods have 0% escalation. In addition, we include two stronger *AI-labeler-only* benchmarks, `AI-only (OpenAI)` and `AI-only (Qwen)`, which run the same VLM models used in our pipeline (respectively `GPT-4o-mini` and `Qwen2-VL-7B-Instruct`) to produce labels (and full reasoning traces) but never escalate to humans. These AI-only variants are important because they isolate the value of human routing from improvements that might simply come from using a stronger judge model: they represent the natural alternative of "just label everything with a better VLM/LLM" and therefore provide a direct point of comparison for cost–accuracy tradeoffs.

(ii) Trace-based router baselines (trace ablations). To test whether routing benefits from observing the judge's intermediate reasoning, we implement two trace-aware routers: a `Reasoning trace only` baseline that routes based solely on the unfolding trace, and a `Reasoning trace + LLM retriever` baseline that augments the router with retrieval-selected in-context demonstrations. Both run an LLM router that repeatedly outputs `ESCALATE` or `CONTINUE` as it reads an increasing trace prefix. This category ablates our framework's use of trace information and retrieval augmentation, without introducing an explicit learned risk model.

(iii) No-trace budgeted router baseline (trace removal). To isolate the specific contribution of reasoning traces while holding fixed the rest of our system design, we add a no-trace variant of our method that still uses the same learned MLP difficulty scoring model to rank instances and enforce the same budgeted routing procedure, but makes routing decisions without access to the judge's step-by-step reasoning. Concretely, this baseline uses the same features to prioritize high-risk examples and then applies the same router logic and budget constraint, but the router only observes the image, caption, and the final AI label. This baseline is a direct ablation that answers whether our gains are driven by cost-aware prioritization and budget control alone, or the additional information contained in intermediate reasoning traces.

(iv) Learning-to-defer (L2D) benchmarks. To compare against established deferral literature, we include standard one-shot L2D policies that learn or define a scalar risk score and defer the top-ρ fraction to humans (Madras et al., 2018; Mozannar & Sontag, 2020). Concretely, `L2D-CLIP-Rule` uses CLIP-based image–text compatibility as a training-free deferral score (Hessel et al., 2021); `L2D-LogReg(CLIP)` and `L2D-MLP(CLIP)` learn supervised deferral scores from vision–language features and tune a threshold to meet the human budget (Madras et al., 2018; Mozannar & Sontag, 2020); and `L2D-LLM-Risk(vision LLM)` replaces the learned score with an LLM-produced discrete risk rating and defers the most uncertain cases under the same budget constraint (Strong et al., 2025; Machcha et al., 2025; Mao et al., 2025). This category isolates the limitations of one-shot risk scoring compared to our stepwise trace-aware routing.

(v) Proposed method. Finally, we evaluate two variants of our proposed method using `GPT-4o-mini` and `Qwen2-VL-7B-Instruct` (open-source), allowing us to assess robustness of the routing design across different underlying judge models while holding the routing logic and budget constraints fixed.

Table 1. Comparison of data splits under different distributional shift measures. "JS divergence" refers to the Jensen-Shannon divergence. "NN" refers to nearest neighbor".

Metrics	Random split	Distributional shift
Topic overlap	21	0
JS divergence (unigram)	0.265	0.342
Centroid cosine distance	0.012	0.049
Average NN distance	0.256	0.343
$\mathbb{P}(Y_H = 1)$ (train)	0.349	0.363
$\mathbb{P}(Y_H = 1)$ (test)	0.357	0.302

Construction of training and testing datasets with distributional shift. To construct training and testing sets with distributional shift, we first embed every caption W into a semantic vector space using a fixed text encoder, and then cluster these caption embeddings into K caption topic groups, treating each cluster as a latent topic. We create shifted splits as follows: We assign all examples whose captions fall into a selected subset of clusters to the test set and all remaining clusters to the training set, ensuring the train and test caption topics are disjoint.To control shift severity, we rank clusters by their distance from the global caption-embedding centroid and designate the farthest clusters as held-out test topics; the remaining clusters form the training set. Finally, we quantify the resulting distributional shift using complementary measures, including (i) the AUC of a classifier trained to distinguish train vs. test captions from their embeddings, where a higher AUC indicates a stronger shift, (ii) Jensen-Shannon (JS) divergence between train and test unigram distributions, and (iii) embedding-space distances such as centroid cosine distance or average nearest-neighbor (NN) distance from test to train.

To make the distributional-shift setting concrete, we summarize the resulting train/test splits in Table 1. Compared to a

standard random split, the proposed topic-holdout construction yields zero topic overlap between training and testing, confirming that test captions come from entirely disjoint latent caption clusters. Consistent with this, all shift quantification measures indicate a substantially stronger covariate shift: unigram JS divergence increases and embedding-space distances also grow, collectively showing both lexical and semantic separation between the two sets. Importantly, the human label rate remains of similar scale across splits, although the shift split exhibits a noticeable label-rate change between train and test, suggesting that topic holdout can also induce a mild label distribution shift in addition to covariate shift. Appendix Figure 9 provides a qualitative sanity check by visualizing representative topics from each split, illustrating that the training and testing captions concentrate on different semantic themes, as intended.

Evaluation metrics. Let $E_i \in \{0, 1\}$ indicate whether instance i is actually escalated to a human, and let $\widetilde{Y}_i$ denote the final post-routing label. We report: (1) Classification accuracy $\Pr(\widetilde{Y} = Y_H)$, measuring agreement between the final routed label and the human label. (2) Escalation rate $\Pr(E = 1)$. (3) Correct escalation rate / error recall $\Pr(E = 1 \mid Y_{AI} \neq Y_H)$, the fraction of AI mistakes caught by escalation. (4) False escalation rate $\Pr(E = 1 \mid Y_{AI} = Y_H)$, the fraction of already-correct AI labels unnecessarily sent to humans. (5) Escalation precision $\Pr(Y_{AI} \neq Y_H \mid E = 1)$. For metrics involving Y_{AI}, Y_{AI} denotes the completed AI-judge label from the logged evaluation run; it is used to determine whether an escalation caught a counterfactual AI mistake. In addition, we report total tokens and LLM runtime to characterize AI-side computation.

Ablation study. We run two ablations to isolate the main sources of gain in the proposed router. First, we compare the online stepwise router with a *full-trace one-shot* router. This baseline keeps the same disagreement scorer f_θ, disagreement ordering $\widehat{p}_i$, demonstration pool $\mathcal{D}_{\text{demo}}$, router model, prompt template, and budget constraint, but waits until the AI judge has finished and queries the router once on $H_{i,T_i} = (W_i, X_i, r_{i,1:T_i})$. Thus, it uses the same trace evidence as the proposed method but cannot early-stop judge generation. Second, we vary the polling interval $\Delta \in \{1, 2, 4\}$, where the router is queried after every Δ completed reasoning steps, with a final query allowed at judge completion. For each Δ, we re-select the router design $\widehat{\psi}_\Delta$ on $\mathcal{D}_{\text{dev}}$ using the same budget-constrained criterion with $\rho = 0.30$. These ablations test whether the gains come from online intervention rather than trace access alone, and how router-query frequency trades off compute and accuracy.

4.2. Experiment results

We summarize the accuracy–escalation tradeoff in Figure 3. On the image–caption agreement task, the judge-only prompting baselines that do not involve human routing achieve relatively low agreement with human labels (zero-shot and few-shot, both at 0% escalation). We further include two stronger *AI-labeler-only* baselines—`AI-only (OpenAI)` and `AI-only (Qwen)`—which mirror our pipeline in that they run a capable VLM/LLM judge to produce a final yes/no label (and a full reasoning trace) but never escalate to humans; these methods achieve substantially higher accuracy at 0% escalation, establishing a meaningful "strong AI-only" reference point. However, routing remains valuable because it selectively invokes humans under a fixed budget and, crucially, enables early stopping of the judge's reasoning on cases that will be escalated, reducing judge-side compute (Table 2). Introducing human routing substantially improves post-routing accuracy because escalated cases are corrected by humans. Among trace-based routers, `Reasoning trace + LLM retriever` achieves the highest accuracy but at a high escalation rate, making it expensive in human effort, while `Reasoning trace only` attains comparable accuracy at a higher escalation. Under a more practical fixed budget around 30% escalation, our proposed trace-aware router provides a stronger operating point: `Proposed (OpenAI)` achieves high accuracy at 0.300 escalation—close to the best trace+retriever accuracy but with roughly half the human workload—and `Proposed (Qwen)` achieves comparable accuracy at the same escalation level, indicating the routing logic remains effective when replacing the underlying judge with an open-source model. Importantly, the no-trace ablation highlights the *value of reasoning traces*. When we keep the same budgeted routing procedure and the same MLP-based difficulty ranking but remove access to the judge's intermediate reasoning, performance drops to lower accuracy at 30% escalation. This gap shows that gains are not explained by budget control or prioritization alone: conditioning escalation decisions on the evolving reasoning trace materially improves which instances are escalated and allows the router to intervene at more informative points in the decision process. Finally, at matched escalation rates, the learning-to-defer (L2D) baselines are consistently lower, suggesting that one-shot risk scoring from CLIP-style features is less effective than stepwise, trace-conditioned escalation.

Complementing these accuracy results, Table 2 reports AI-side computational cost: compared to AI-only labeling (which always generates a full trace), our budgeted routing reduces total judge output tokens and wall-clock runtime by early-stopping the judge on escalated cases, showing that the proposed router improves labeling accuracy *and* reduces judge inference cost while respecting a fixed human-budget

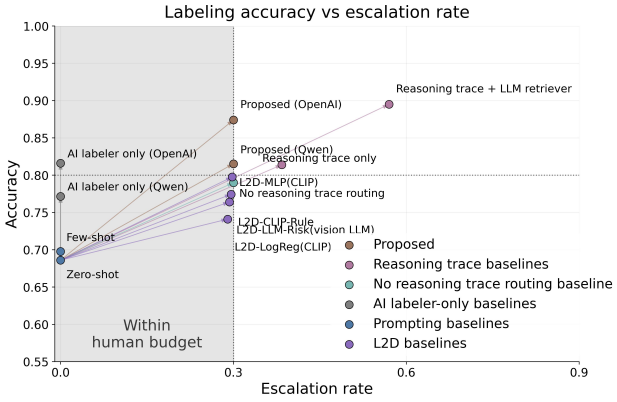

Figure 3. Labeling accuracy and escalation rate trade-off among the benchmark methods in comparison. The grey shaded area represents the area "within human labeling budget" set at 30%.

Table 2. Compute comparison between AI-only labeling and budgeted trace-aware routing. We report (i) generated judge tokens, which is a proxy for model inference cost and (ii) end-to-end runtime.

Method	Total tokens	Runtime (min)
AI-only (GPT)	40,000	17.0
AI-only (Qwen)	44,000	72.9
Proposed (GPT, ρ=0.30)	32,800	13.94
Proposed (Qwen, ρ=0.30)	36,800	60.75

constraint.

Figure 4 reports escalation metrics in detail. The proposed routers achieve the strongest combination of error recall and escalation precision, especially Proposed (OpenAI), indicating that they catch more true AI mistakes while spending fewer human labels on already-correct cases. In contrast, the trace-only baselines and most L2D methods have lower recall or lower precision. False escalation rates are broadly comparable across methods, suggesting that the recall gains of the proposed routers do not come from substantially more unnecessary escalations.

We summarize the two ablations in Figure 5 and Appendix

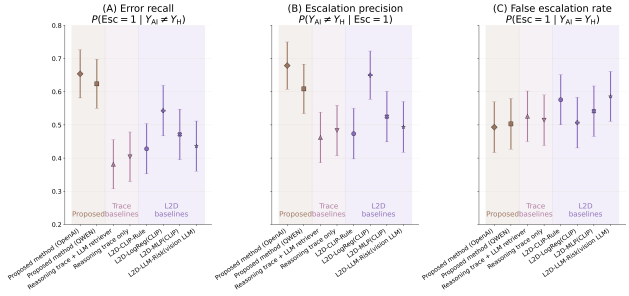

Figure 4. Comparison of (A) error recall rate, (B) escalation precision, and (C) false escalation rate.

Figure 5. Ablation study: online stepwise monitoring versus full-trace one-shot routing. (A) Final accuracy, (B) error recall, (C) escalation precision, (D) total tokens, and (E) end-to-end runtime.

Figure 10. The full-trace one-shot router uses the same trace evidence as the proposed router, but only after judge completion. It therefore achieves slightly stronger quality metrics, but cannot reduce judge-generation cost. In contrast, the proposed online router trades a small amount of final-label quality for lower token usage and runtime by intervening before the trace is complete. The polling-interval ablation further shows that less frequent router queries reduce router overhead, while delaying possible intervention and increasing judge-side computation. Overall, these results show that the main benefit comes not merely from using reasoning traces, but from using them sequentially during generation, with moderate polling providing a practical compute-accuracy tradeoff.

Due to space constraints, we defer more experiment results to Appendix Section A.

5. Discussion

We propose a trace-aware AI router that monitors the AI labeler's evolving evidence and chooses the earliest point to intervene, which yields practical benefits in improving labeling accuracy under a fixed budget constraint. We acknowledge that the current method has several limitations that warrant further exploration. Our deployment procedure enforces the escalation budget by prioritizing instances using predicted disagreement probability and stopping once the budget is exhausted. In settings where fairness or coverage constraints matter, additional mechanisms may be necessary.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Agrawal, A., Kedia, N., Panwar, A., Mohan, J., Kwatra, N., Gulavani, B. S., Tumanov, A., and Ramjee, R. Taming throughput-latency tradeoff in LLM inference with sarathi-serve. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, 2024.

Han, T., Wang, Z., Fang, C., Zhao, S., Ma, S., and Chen, Z. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., and Choi, Y. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Levinboim, T., Thapliyal, A. V., Sharma, P., and Soricut, R. Quality estimation for image captions based on large-scale human evaluations. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 3157–3166, 2021.

Machcha, S., Yerra, S., Sultana, S., Yu, H., and Yao, Z. Do large language models know when not to answer in medical qa? In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025)*, pp. 27–35, 2025.

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

Mao, Y., Durand, T., Mehrasa, N., He, J., and Ester, M. Calibrating llms for selective prediction: Balancing coverage and risk. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*, 2025.

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

Strong, J., Men, Q., and Noble, J. A. Trustworthy and practical ai for healthcare: A guided deferral system with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28413–28421, 2025.

Yao, Y., Jin, H., Shah, A. D., Han, S., Hu, Z., Stripelis, D., Ran, Y., Xu, Z., Avestimehr, S., and He, C. Scalellm: A resource-frugal LLM serving framework by optimizing end-to-end efficiency. In *Proceedings of EMNLP 2024 Industry Track*, 2024.

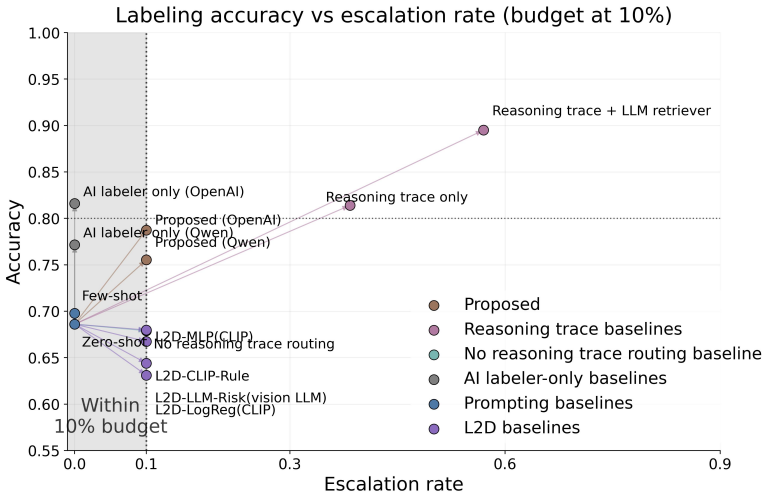

Figure 6. Labeling accuracy and escalation rate trade-off among the benchmark methods in comparison. The grey shaded area represents the area "within human labeling budget" set at 10%.

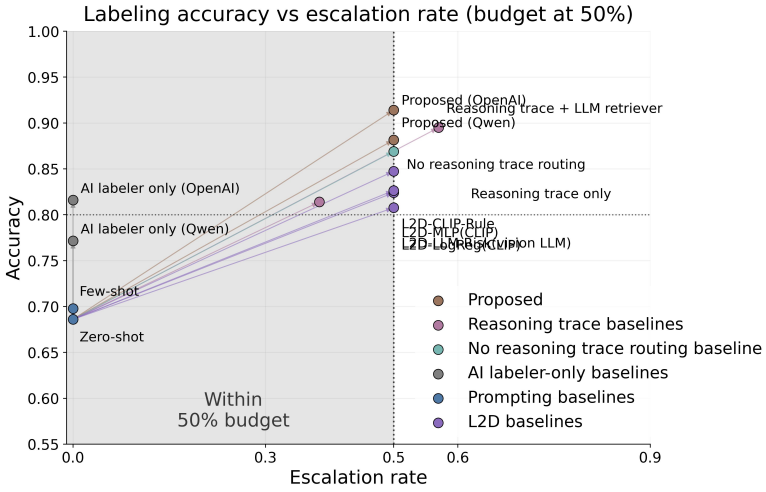

Figure 7. Labeling accuracy and escalation rate trade-off among the benchmark methods in comparison. The grey shaded area represents the area "within human labeling budget" set at 50%.

A. Additional experiment results

In this section, we provide additional experiment results to show the case when human–AI escalation behavior under two other fixed budget settings (10%, and 50% escalation rate) in Figure 6 and Figure 7. These two figures demonstrate the gain in accuracy of our proposed method by spending additional human effort. In the second set of experiment results, we consider a setting where there's no distribution shift in the train and test datasets, which corresponds to "random split" in Table 1. We summarize the results in Figure 8. Figure 8 shows an overall improvement in accuracy under no distributional shift, which further validates the robustness of our proposed method when there is distributional shift (Figure 3 in the main manuscript).

Additionally, we provide examples of the most common training versus testing set topics in Figure 9. We also provide additional ablation study results in Figure 10. Figure 10 suggests that when the polling interval is increased from $\Delta = 1$ to $\Delta \in \{2, 4\}$, the router is invoked less often, which mechanically reduces router-side token consumption. Empirically, router-side token usage decreases approximately in proportion to $1/\Delta$, consistent with the fact that the number of router calls per example is reduced by the same factor. At the same time, less frequent polling delays the earliest possible intervention point, so the judge is allowed to continue generating for longer before escalation can occur. As a result, judge-side token

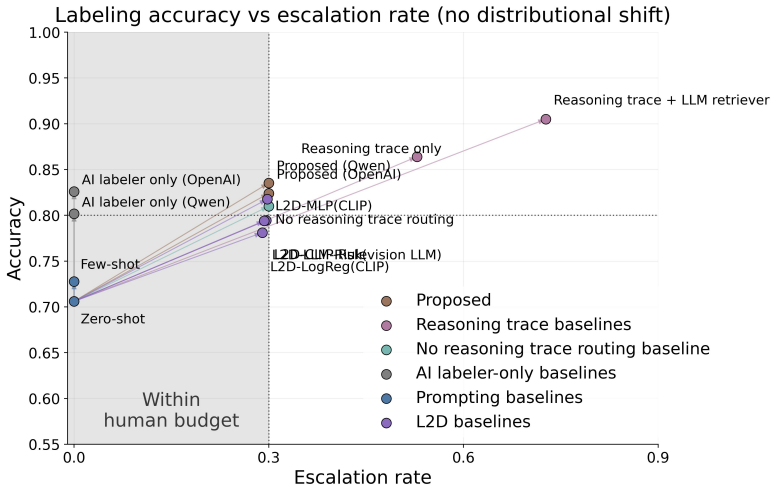

Figure 8. Labeling accuracy and escalation rate trade-off among the benchmark methods in comparison when there is no distributional shift between train and test datasets. The grey shaded area represents the area "within human labeling budget" set at 30%.

Figure 9. Most common training versus testing set topics

usage rises with Δ and progressively approaches the AI-only baseline. This pattern highlights the central tension in the online design: more frequent monitoring increases router overhead, but also enables earlier stopping of judge generation when the trace already contains sufficient evidence of likely error. These findings suggest that moderate polling can preserve most of the efficiency gains of online routing while substantially reducing router-side overhead.

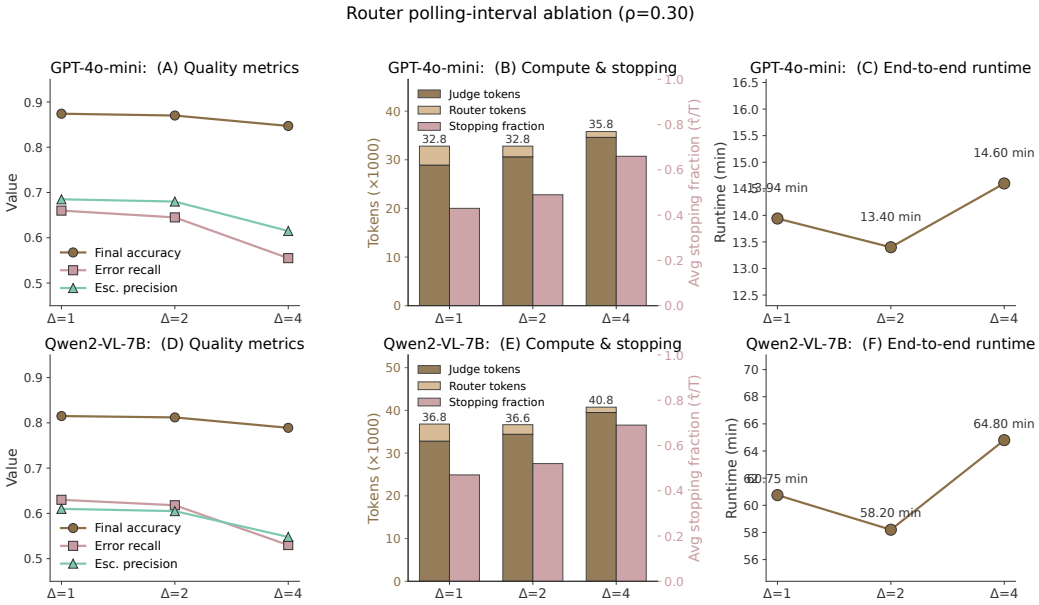

Figure 10. Ablation study: Router polling-interval ablation for GPT model (A) – (C) and Qwen (D)–(F) under polling intervals $\Delta \in \{1, 2, 4\}$. (A) and (D): Final accuracy, error recall, and escalation precision. (B) and (E): Stacked judge/router token usage and average stopping fraction. (C) and (F): runtime in minutes.