Email Classification

System Design

Krishna Kumar
Data Scientist
kkcareer@hotmail.com

I. Problem Statement

A. Introduction

Email classification is a core capability for operational workflows within the QMA workflow. The predicted intent directly drives downstream actions such as email routing, enrichment, exception handling, and potential automated tool invocation. As a result, the classifier must be both *accurate* and *operationally reliable*, exhibiting low latency, high throughput, and predictable behavior under distribution shift.

Today, two unsatisfactory extremes are observed: (i) an LLM-based classifier (e.g., LLaMA-70B) that provides strong accuracy but suffers from high latency, limited throughput, and operational rigidity (e.g., tightly coupled hierarchical constraints), and (ii) a legacy traditional model (e.g., Random Forest) that is operationally efficient but insufficiently accurate for action-triggering decisions.

B. Design Goal

We aim to design a **curated, production-grade email classification system** that achieves:

- **Low latency and high throughput:** suitable for real-time or near-real-time email processing.
- **High accuracy with calibrated confidence:** robust enough to safely drive automated actions while supporting abstention when uncertain.
- **Customization and maintainability:** support for evolving taxonomies, business-line-specific classes, and rapid iteration.
- **Governance readiness:** traceability, monitoring, and audit-friendly artifacts aligned with model risk and operational controls.

C. Problem Statement

The system is required to:

- Ingest an email, including subject, body, inline images, attachments, and relevant metadata.
- Predict the most appropriate intent label from a configurable taxonomy.
- Return a calibrated confidence score with explicit support for abstention. (The classifier must not only predict what the email intent is, but also how confident it is in that prediction — and that confidence must be meaningful and trustworthy, not just a raw model score. If the model is not confident enough, it must be allowed to say: "I don't know" instead of forcing a potentially dangerous wrong classification. Either Return one label or abstain with reason)
- Emit a standardized decision payload (say Json or Markup)that drives downstream actions and tool-calling.

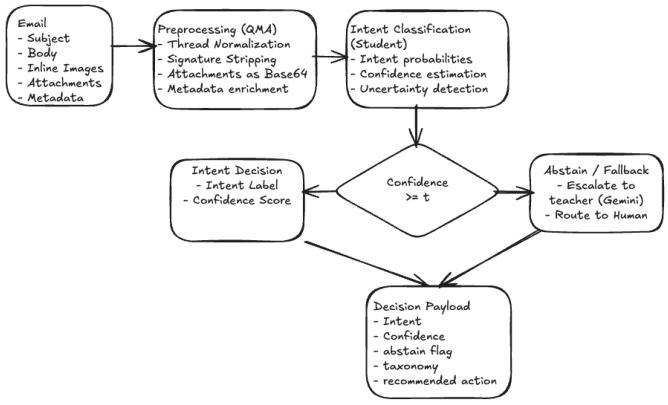

Fig. 1. Conceptual email intent classification pipeline. An input email is processed through signal extraction and intent classification, followed by confidence-based decision gating. The system either emits a finalized intent or abstains and triggers a fallback, producing a standardized decision payload for downstream automation.

D. Machine Learning Formulation

Let x denote an input email and $\mathcal{Y}$ denote the intent taxonomy. The learning objective is to:

- Predict an intent label $y \in \mathcal{Y}$.
- Estimate a calibrated confidence score $p(y \mid x)$.
- Abstain or route to a higher-accuracy fallback when $p(y \mid x) < \tau$.

In addition, the system adopts a **teacher–student distillation** paradigm, in which a high-capacity teacher model (Gemini) produces labels and structured step-by-step rationales, and a lower-latency student model (Gemma) is trained to replicate this behavior while meeting strict inference-time performance constraints.

II. Metrics

This section describes the metrics used to evaluate the performance of the email intent classification system, both as a standalone machine learning component and as a critical dependency within larger operational workflows. Similar to established evaluation practices, we distinguish between *offline*

metrics, which assess model quality in isolation, and *online metrics*, which measure the impact of the classifier on end-to-end system performance

A. Offline vs. Online Metrics

Offline metrics are used to:

- Compare different classification models and training strategies.
- Measure intrinsic classification quality independent of deployment.

Online metrics are used to:

- Measure the performance of downstream workflows when a given classifier is deployed.
- Quantify operational impact, including automation quality, cost, and latency.

Both classes of metrics are required, as strong offline performance does not necessarily translate into improved system-level outcomes.

B. Offline Metrics

Offline evaluation focuses on the intent classification component itself. Given a labeled dataset of emails, the model predicts an intent label or abstains.

1) Classification Accuracy Metrics: Let $\mathcal{Y}$ denote the intent taxonomy. Standard multi-class classification metrics are used:

- **Precision:** the fraction of predicted intents that are correct.
- **Recall:** the fraction of true intents that are correctly predicted.
- **F1-score:** the harmonic mean of precision and recall.

These metrics capture the trade-off between false positives (incorrectly triggering actions) and false negatives (failing to trigger required actions).

2) Abstention-Aware Metrics: Because abstention is a first-class outcome, traditional accuracy metrics are extended as follows:

- **Coverage:** the fraction of emails for which the model emits a non-abstained prediction.
- **Selective accuracy:** accuracy computed only over non-abstained predictions.

By varying the confidence threshold τ, coverage–accuracy curves can be generated to identify safe operating points.

3) Calibration Metrics: Since confidence scores drive abstention and cascading decisions, probability calibration is critical. We evaluate:

- **Expected Calibration Error (ECE):** the discrepancy between predicted confidence and empirical accuracy. It is a summary measure of the gap between predicted confidence and observed accuracy.
- **Reliability diagrams:** visual inspection of confidence calibration. It is a visual tools that compare predicted confidence against actual accuracy.

Temperature scaling is applied to calibrate the student model's confidence estimates without altering its predicted labels. A scalar temperature parameter T rescales the model's output logits prior to probability computation, with $T > 1$ reducing overconfidence. The temperature is learned on a held-out validation set to minimize calibration error, ensuring that confidence thresholds behave predictably in production.

Well-calibrated confidence scores ensure that threshold-based decision policies behave predictably in production.

4) Coverage vs. Confidence: Coverage measures the fraction of emails for which the system emits a confident intent prediction rather than abstaining. By varying the confidence threshold, coverage–confidence curves illustrate the trade-off between automation and certainty: higher thresholds reduce coverage but improve reliability, while lower thresholds increase coverage at the cost of greater risk. These curves are used to select operating points that balance safety, throughput, and fallback cost.

5) Micro vs. Macro-Averaged Metrics: As in entity linking evaluation, both micro- and macro-averaged metrics are reported :

- **Micro-averaged metrics** aggregate predictions across all emails and are appropriate when overall volume-weighted performance is the primary concern.
- **Macro-averaged metrics** compute metrics per intent class and then average, highlighting performance on rare or high-risk intents.

Macro-averaged metrics are particularly important in this domain, as infrequent intents (e.g., margin calls or settlement exceptions) often carry disproportionate operational risk.

6) Cost-Sensitive Error Analysis: In addition to aggregate metrics, errors are categorized by operational severity:

- **High-cost false positives:** incorrect intent predictions that trigger irreversible or expensive actions.
- **High-cost false negatives:** failures to detect intents that require timely intervention.

Error rates are reported separately for these categories to reflect real-world risk.

C. Online Metrics

Online metrics evaluate the classifier as a component within live workflows, typically through controlled A/B experiments .

Key online metrics include:

- **Automation success rate:** fraction of automated actions completed without manual intervention.
- **Fallback rate:** fraction of emails routed to human review or teacher models.
- **End-to-end latency:** impact on workflow completion time.
- **Operational error rate:** rate of downstream failures attributable to misclassification.

These metrics directly measure whether improvements in offline classification performance translate into meaningful gains for the overall system.

D. Interpretation of Confidence Scores

In this system, confidence scores play a central role in both training and inference. However, the notion of confidence differs between the teacher and student models due to their distinct modeling roles and output structures. It is therefore important to clarify how confidence is interpreted and used for each model.

1) Teacher Model Confidence: The teacher model is a high-capacity generative language model that does not natively output calibrated class probabilities. As a result, teacher confidence is not treated as a precise probability estimate. Instead, it is derived from behavioral signals such as prediction consistency across multiple runs, coherence and specificity of generated rationales, and internal likelihood indicators.

Teacher confidence is primarily used:

- During offline training, to filter or down-weight low-quality or ambiguous supervision examples.
- During online fallback inference, to assess the reliability of teacher-generated outputs before accepting them as final decisions.

Teacher confidence is therefore interpreted as an approximate measure of reliability rather than a calibrated probability.

2) Student Model Confidence: In contrast, the student model is trained as a discriminative classifier that explicitly produces class probability estimates. These probabilities are calibrated using held-out validation data to ensure that confidence values correspond to empirical correctness.

Student confidence is used:

- To determine whether an intent prediction should be accepted or abstained.
- To trigger cascaded fallback to the teacher model when confidence falls below a predefined threshold.
- To support consistent and auditable decision policies in production.

Unlike teacher confidence, student confidence is designed to be directly interpretable as the likelihood that a predicted intent is correct.

TABLE I. Interpretation of confidence scores for teacher and student models.

	Teacher Model	**Student Model**
Confidence type	Derived / heuristic	Explicit / calibrated
Primary use	Training supervision	Production gating
Used for abstention	No	Yes
Probability interpretation	Approximate	Direct

3) Summary of Confidence Usage: Only the student's confidence is trusted for automated decisions; the teacher's confidence is used to guide supervision and fallback reliability.

Overall, teacher and student confidence serve complementary purposes. Teacher confidence provides a heuristic measure of supervision quality and fallback reliability, while student confidence provides a calibrated signal for automated decision-making. This distinction enables robust cascading behavior without over-reliance on heuristic confidence estimates in production.

E. Model Comparison and Selection

Final model selection is based on a combination of:

- Offline classification and calibration metrics.
- Coverage–accuracy trade-offs under different abstention thresholds.
- Online system-level performance in A/B testing.

A model is considered superior only if it demonstrates consistent improvements across both offline and online metrics, aligning model quality with operational reliability.

III. System Architecture

This section describes the architectural components of the email intent classification system. The architecture is organized into two primary paths:

1) Model generation path (training flow)
2) Model execution (Inference) path (prediction flow)

This separation enables independent evolution of training pipelines and production inference services, while ensuring consistency through shared data contracts and evaluation metrics.

A. Model Generation Path (Training Flow)

The model generation path is responsible for producing and continuously improving the intent classification models. It consists of the following components.

1) Training Data Generation: Training data is derived from a combination of:

- Historical emails with human-validated intent labels.
- Manually curated edge cases for high-risk intents.
- Synthetic supervision generated by a high-capacity teacher model.

Emails are normalized and enriched with metadata prior to labeling to ensure consistency between training and inference.

2) Teacher Model Inference: A high-capacity foundation model (Gemini) acts as the teacher in the distillation process. For each training example, the teacher produces:

- An intent label aligned with the current taxonomy.
- A structured step-by-step rationale explaining the classification decision.
- A confidence estimate reflecting prediction uncertainty.

These outputs serve as high-quality supervision signals for training the student model.

In our design, the teacher model (Gemini) serves two distinct roles. During offline training, the teacher is used to generate labeled training triples consisting of the input email, a high-quality intent label, and a structured rationale explaining that prediction. These triples form the supervision signal used to train the student model (Gemma) under a step-by-step distillation objective. Once the student is trained, it is deployed for online inference due to its low latency and high throughput. Optionally, in a cascaded inference strategy, the teacher model may be invoked at runtime for cases where the student's calibrated confidence falls below a defined threshold, providing a high-accuracy fallback mechanism while controlling operational cost and latency.

3) Student Model Training: The student model (Gemma) is trained using a distillation objective that incorporates:

- Label prediction loss. The Label loss teaches the student what answer to give.
- Rationale consistency loss. The Rationale loss teaches the student how to reason.
- Optional calibration-aware objectives. Calibration loss teaches the student when to be unsure.

This approach enables the student model to approximate the teacher's reasoning while meeting strict latency and throughput requirements.

The student model (Gemma) is trained using a distillation-based objective that combines multiple learning signals. In addition to learning to predict the correct intent label, the student is encouraged to align its reasoning with the teacher model through structured rationale supervision. Optional calibration-aware objectives are used to ensure that predicted confidence scores accurately reflect uncertainty.

Together, these objectives allow the student model to approximate the teacher's decision-making behavior while remaining efficient enough to satisfy strict latency and throughput requirements in production.

In Nutshell, The student model is trained to match not only the teacher's predictions, but also its reasoning and confidence behavior. This enables high-quality decisions at significantly lower inference cost.

4) Evaluation and Metrics: Offline metrics are computed during training to assess:

- Classification quality (precision, recall, F1-score).
- Calibration quality (ECE, reliability curves).
- Coverage–accuracy trade-offs under abstention.

These metrics guide model selection and determine readiness for deployment.

B. Model Execution Path (Prediction Flow)

The model execution path governs how the system processes live emails in production.

1) Input Processing: Each incoming email, including subject, body, attachments, and metadata, is passed through a preprocessing layer that performs normalization, feature extraction, and schema validation.

2) Student Model Inference: The preprocessed email is evaluated by the deployed student model, which outputs:

- A predicted intent label.
- A calibrated confidence score.
- An abstention signal if uncertainty exceeds predefined thresholds.

3) Confidence-Based Decision Gating: A decision gate evaluates the model output:

- If confidence exceeds the acceptance threshold, the intent prediction is finalized.
- If confidence falls below the threshold, the system abstains and triggers a fallback path.

Fallback actions may include escalation to the teacher model or routing to human review, depending on operational policies.

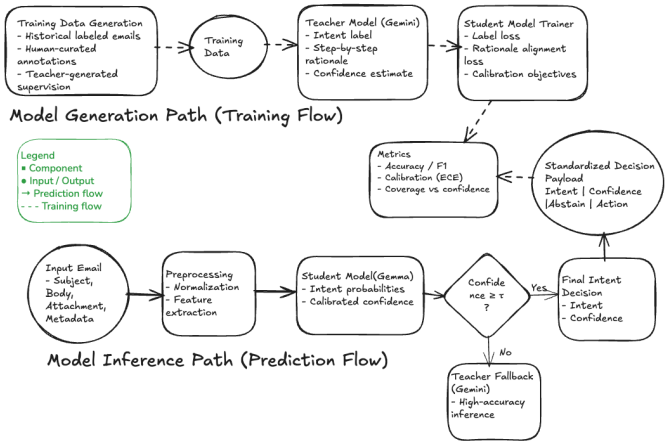

Fig. 2. System architecture for email intent classification with teacher–student distillation. The architecture is organized into a model generation path (training flow) and a model execution path (inference flow). A high-capacity teacher model provides labels and structured rationales during training, while a low-latency student model serves as the primary inference engine. Confidence-based decision gating enables selective fallback to the teacher model in low-confidence cases, balancing accuracy, latency, and operational risk.

4) Decision Payload Emission: The final output of the execution path is a standardized decision payload containing:

- Intent label (if available).
- Confidence score.
- Abstention flag.
- Taxonomy and model version metadata.
- Recommended downstream action.

This payload is consumed deterministically by downstream automation and tool-calling systems.

Figure 2 illustrates the end-to-end system architecture, highlighting the separation between training and inference, the role of teacher–student distillation, and the confidence-based cascading strategy used to control risk and cost.

C. Architectural Benefits

This two-path architecture provides:

- Clear separation between training and inference concerns.
- Scalable, low-latency production inference.
- Controlled use of high-cost teacher models.
- Strong governance through explicit metrics and versioning.

IV. Cascaded Inference Strategy

This section describes the cascaded inference strategy used to balance accuracy, latency, throughput, and operational risk in the email intent classification system. The strategy combines a low-latency student model with a high-capacity teacher model, invoking the latter only when necessary.

A. Motivation

A single-model approach presents an inherent trade-off. High-capacity foundation models provide strong classification accuracy but incur high latency and cost, making them unsuitable for large-scale, real-time deployment. Conversely, lightweight models offer efficient inference but may struggle with ambiguous or rare cases.

The cascaded inference strategy addresses this trade-off by:

- Using a fast student model for the majority of predictions.
- Reserving the teacher model for cases where confidence is insufficient.
- Ensuring safe automation without sacrificing throughput.

B. Cascaded Decision Flow

At inference time, each email follows a structured decision flow:

1) The incoming email is processed by the student model, which produces an intent prediction and a calibrated confidence score.
   The confidence score is derived from the student model's predicted class probabilities and represents an estimate of the likelihood that the predicted intent is correct. Raw model probabilities are calibrated using a held-out validation set to ensure that confidence values correspond to empirical accuracy. The calibrated confidence score is then used to drive abstention and cascading decisions in production.
2) A confidence-based decision gate evaluates the prediction.
3) If the confidence exceeds a predefined threshold, the prediction is accepted and emitted.
4) If the confidence falls below the threshold, the system abstains and triggers a fallback path.

This gating mechanism ensures that low-confidence predictions do not directly drive automated actions.

C. Fallback Mechanisms

When the student model abstains, one or more fallback actions may be invoked depending on operational requirements:

- Escalation to the teacher model for higher-accuracy inference.
- Routing to human review for critical or high-risk intents.
- Invocation of conservative default workflows.

Fallback decisions are policy-driven and can be adjusted independently of model retraining.

D. Threshold Selection and Trade-offs

The confidence threshold τ governs the balance between automation and safety:

- Lower thresholds increase automation coverage but risk higher error rates.
- Higher thresholds reduce errors but increase fallback frequency and cost.

Thresholds are selected based on offline calibration metrics and validated through online experiments to achieve acceptable trade-offs between accuracy, latency, and operational cost.

E. Benefits of Cascaded Inference

The cascaded strategy provides several key benefits:

- **Performance efficiency:** most emails are handled by the low-latency student model.
- **Accuracy preservation:** ambiguous cases benefit from the teacher model's superior reasoning.
- **Cost control:** expensive inference is applied selectively rather than universally.
- **Operational safety:** explicit abstention prevents unsafe automated actions.

Overall, cascaded inference enables the system to deliver high-quality intent classification while meeting stringent production constraints.

V. Training Data Generation

High-quality training data is a critical dependency for the performance, robustness, and governance of the email intent classification system. Unlike generic text classification tasks, this system operates in a domain where classification outcomes directly drive downstream actions. As a result, training data must be both representative of real operational traffic and aligned with evolving business taxonomies.

Following established practices in complex NLP systems, training data generation is approached through a combination of curated human-labeled data and model-generated supervision.

A. Sources of Training Data

Training data is derived from the following complementary sources:

1) Historical Labeled Emails: The primary source of training data consists of historical emails that have been manually labeled or implicitly resolved through downstream workflows. These examples reflect real operational distributions, including class imbalance, ambiguous cases, and domain-specific language.

Where necessary, labels are normalized to the current intent taxonomy to ensure consistency across training iterations.

2) Human-Curated Annotations: For high-risk or business-critical intents, additional data is generated through targeted human annotation. Subject-matter experts review selected emails and assign intent labels according to clearly defined guidelines.

This process is particularly important for:

- Rare but high-impact intents.
- Newly introduced intent categories.
- Edge cases that are poorly represented in historical data.

Human-curated examples serve as high-quality anchors during model training and evaluation.

3) Teacher-Generated Supervision: To improve data efficiency and generalization, a high-capacity teacher model is used to generate additional supervision signals. For each selected email, the teacher produces:

- An intent label aligned with the target taxonomy.

- A structured rationale explaining the classification decision.
- Optional confidence or uncertainty indicators.

These teacher-generated outputs form training triples of the form (x, y, r), where x is the input email, y is the intent label, and r is the corresponding rationale. This enriched supervision is used to train the student model via step-by-step distillation.

B. Data Curation and Quality Control

All training data sources are subject to validation and quality checks prior to model training. These include:

- Consistency checks against the active intent taxonomy.
- Deduplication and thread-level normalization.
- Review of label distributions to identify imbalance or drift.

Inconsistent or low-confidence labels are either corrected or excluded to prevent the propagation of noise into the student model.

C. Dataset Composition

The final training dataset is composed by combining:

- Human-labeled examples for accuracy and governance.
- Teacher-generated examples for coverage and scale.

Sampling strategies are applied to ensure adequate representation of rare intents while preserving realistic traffic distributions.

D. Alignment with Inference-Time Behavior

Training data generation is designed to closely mirror inference-time conditions. The same preprocessing steps, input representations, and taxonomy versions used in production are applied during training. This alignment minimizes training–serving skew and improves the reliability of confidence-based decision policies.

Overall, this data generation strategy enables the student model to learn both accurate intent predictions and robust reasoning patterns, while remaining adaptable to evolving business requirements.

VI. Modeling

This section briefly describes the modeling choices underlying the email intent classification system. The goal of the modeling layer is to produce contextual representations of email content that support accurate intent prediction, calibrated confidence estimation, and efficient inference at scale.

A. Contextual Representation of Email Content

Email intent classification requires understanding the meaning of words and phrases in context, as the same terms may imply different intents depending on surrounding language, metadata, or attachments. As a result, the modeling approach relies on contextualized text representations rather than static word embeddings.

Modern transformer-based language models provide bidirectional contextual representations that jointly encode information from the entire input sequence. This allows the model to capture semantic cues that may appear before or after key phrases, which is essential for accurately interpreting operational emails.

B. Teacher and Student Model Roles

The system employs two classes of models with distinct responsibilities:

- **Teacher model:** A high-capacity foundation model is used during offline training to generate high-quality intent labels and structured rationales. Its role is to provide rich supervision rather than serve as the primary inference engine.
- **Student model:** A smaller, optimized language model is trained to approximate the teacher's behavior while meeting strict latency and throughput constraints. This model is deployed for online inference.

Both models operate on the same input representation to minimize training–serving skew.

C. Intent Classification Head

On top of the contextual representation produced by the language model, a lightweight classification head is used to predict intent probabilities over the configured taxonomy. The output of this head serves two purposes:

- Selecting the most likely intent label.
- Producing class probabilities that are subsequently calibrated and used for abstention and cascading decisions.

This separation allows the underlying language model to focus on representation learning while keeping the decision logic explicit and auditable.

D. Distillation-Oriented Training

Rather than training the student model solely on intent labels, the modeling approach incorporates step-by-step distillation. The student is trained to align with the teacher model's predictions and reasoning patterns, enabling improved generalization and robustness compared to label-only supervision.

This modeling choice is particularly effective for rare or ambiguous intents, where purely supervised learning may be data-limited.

E. Model Selection Considerations

Model architectures are selected based on the following criteria:

- Ability to capture long-range context in email content.
- Compatibility with distillation-based training.
- Inference efficiency under production traffic.
- Support for calibrated probability outputs.

The resulting modeling stack balances expressiveness, interpretability, and operational efficiency.

Appendix

This annexure describes how the student model is trained using supervision generated by a higher-capacity teacher model. The objective of this process is to transfer the teacher's decision-making behavior to a smaller, more efficient model suitable for production deployment.

A. Teacher-Generated Training Signals

For each training example, the teacher model processes the input email and produces a structured supervision triple consisting of:

- **Input (x):** the original email content, including subject, body, and relevant metadata.
- **Label (y):** the predicted intent label aligned with the target taxonomy.
- **Rationale (r):** a structured explanation describing the reasoning behind the predicted intent.

These teacher-generated triples (x, y, r) provide richer supervision than label-only datasets by explicitly capturing both the outcome and the reasoning process.

B. Student Training Objectives

The student model is trained to learn from the teacher using multiple complementary objectives.

1) Label Prediction Objective: The primary objective of the student model is to correctly predict the intent label y given the input email x. This objective ensures that the student learns to replicate the teacher's final decision.

2) Rationale Alignment Objective: In addition to label prediction, the student is trained to align with the teacher's rationale. This encourages the student to internalize the reasoning patterns used by the teacher, rather than relying on superficial correlations in the data.

Rationale alignment improves generalization, particularly for rare or ambiguous intents where labeled data may be limited.

3) Confidence and Uncertainty Learning: Optionally, the student model is trained to align its confidence estimates with the teacher's behavior. This helps ensure that high confidence corresponds to high empirical accuracy, which is critical for downstream abstention and cascading decisions.

C. Combined Learning Objective

During training, the student optimizes a combined objective that incorporates:

- Correct intent classification.
- Consistency with the teacher's reasoning.
- Reliable confidence estimation.

By balancing these objectives, the student learns not only what decision to make, but also when to be confident and when to defer.

D. Benefits of Teacher–Student Learning

This training approach provides several advantages:

- Improved accuracy compared to label-only supervision.
- Better performance on rare and edge-case intents.
- More reliable confidence scores for safe automation.
- Significantly lower inference latency and cost compared to deploying the teacher model directly.

As a result, the student model approximates the teacher's behavior while remaining efficient enough to satisfy production requirements.

E. Role of the Teacher at Inference Time

Once training is complete, the student model is deployed as the primary inference engine. The teacher model is not required for routine inference and is invoked only as an optional fallback in low-confidence or high-risk scenarios, as described in the cascaded inference strategy.

This separation allows the system to achieve both high performance and operational efficiency.

References

[1] W. Wang, A. Srivastava, J. Wei, A. Rajkumar, N. Arora, and C. Xiong, "Distilling Step-by-Step: Instruction-Following Language Models Can Be Outperformed by Smaller Models via Step-by-Step Rationales," *arXiv preprint arXiv:2305.02301*, 2023. Available: https://arxiv.org/abs/2305.02301

[2] Hugging Face, "Distilling Step-by-Step – Paper with Code," 2023. Available: https://huggingface.co/papers/2305.02301

[3] Data Science Collective, "How to Distill a LLM Step-by-Step," *Medium*, 2023. Available: https://medium.com/data-science-collective/how-to-distill-a-llm-step-by-step-58f06fcf4bfa