CalCo: A Hierarchical Bayesian Framework for Scalable Human-LLM Hybrid Labeling

Viet-An Nguyen
Meta
Menlo Park, CA, USA

Xu Chen
Meta
Menlo Park, CA, USA

Udi Weinsberg
Meta
Menlo Park, CA, USA

Abstract

With the emergence of Large Language Models (LLMs), data annotation workflows are increasingly shifting toward hybrid systems where LLM-generated labels complement or replace human effort. However, effectively incorporating these non-deterministic signals into existing human labeling processes remains a challenge, particularly in complex industrial settings involving multiple correlated tasks and diverse model architectures. In this work, we introduce CalCo, a comprehensive system deployed at scale, designed to integrate LLM and human labels in a principled manner. At the core of CalCo is a novel hierarchical Bayesian model that explicitly captures the complex dependencies inherent in the multi-task, multi-LLM annotation process. Unlike traditional approaches that treat annotation inputs in isolation, CalCo jointly models the inter-dependencies across related tasks and the correlations among diverse LLMs. This holistic approach allows the system to share statistical strength across questions and capture shared error profiles among models. We demonstrate the efficacy of our approach through extensive empirical results on both simulated and real-world datasets, showing that CalCo significantly outperforms baseline methods in a wide range of applications.

CCS Concepts

• **Information systems → Crowdsourcing**; • **Computing methodologies → Latent variable models**.

Keywords

Human-AI Collaboration; Data Annotation; Crowdsourcing; Hierarchical Bayesian Modeling

ACM Reference Format:

Viet-An Nguyen, Xu Chen, and Udi Weinsberg. 2026. CalCo: A Hierarchical Bayesian Framework for Scalable Human-LLM Hybrid Labeling. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3818140

1 Introduction

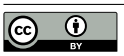
With the rapid emergence and increasing capabilities of Large Language Models (LLMs), the landscape of data annotation is undergoing a fundamental shift. It is becoming increasingly popular to utilize labels generated by LLMs, either as a replacement for or in conjunction with human annotations, giving rise to "LLMs-as-Judges" and "LLMs-as-Labelers" paradigms [3, 4, 34]. This shift promises to scale data production and evaluation significantly; however, it also introduces the complexity of managing non-deterministic, machine-generated outputs alongside traditional crowd-sourced data.

A primary challenge in this evolving landscape is determining how to incorporate LLM-generated labels into existing human labeling processes in a principled, statistically-sound way [39]. While LLMs can be cost-effective, they are not infallible, often exhibiting specific biases or hallucinations [16]. Basic methods such as averaging and thresholding probabilities or treating LLMs as independent "experts" can lead to suboptimal results. To fully leverage the complementary strengths of humans and AI, we require robust probabilistic frameworks capable of calibrating these diverse signals and combining them to infer ground truth with high confidence [33].

Figure 1 illustrates the multi-task hybrid labeling workflow considered in this work. For a given input (e.g., a social media post), the system collects signals from a diverse set of agents, comprising continuous probability scores from multiple LLMs (e.g., GPT-4, Llama) and discrete labels from human annotators. These agents annotate multiple semantically related tasks simultaneously—such as Toxicity, Sentiment, and Topic—where the label prevalence in one task (e.g., "Toxic") inherently carries statistical information about another (e.g., "Negative" Sentiment). This setup produces a heterogeneous set of noisy observations, ranging from granular confidence distributions provided by AI models to sparse, categorical votes provided by humans.

Further complications arise in multi-task annotations, in which annotators (AI and humans) are asked multiple related tasks (or questions) regarding a given "job", such as assessing a sample of posts in a social network for their topics, sentiments, and whether they are potentially offensive. Such tasks often have dependencies; the difficulty or prevalence of one task often informs another. As such, there are opportunities to save resources by reducing the number of annotators deployed for each task. By modeling these inter-dependencies, a system can leverage information from high-confidence annotations in one task to improve inference in related, sparse, or noisy tasks.

Finally, industrial deployments often have access to multiple models, ranging from traditional machine learning classifiers to various LLM families (e.g., Llama [37], GPT [1], Gemini [35]). These models can exhibit varying levels of performance and operational costs, and can be correlated. For example, models fine-tuned from similar base architectures (e.g., GPT-5.2, GPT-5 mini, and GPT-5 nano) likely share failure modes. Ignoring these correlations can lead to overconfidence in consensus predictions when models make



KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3818140

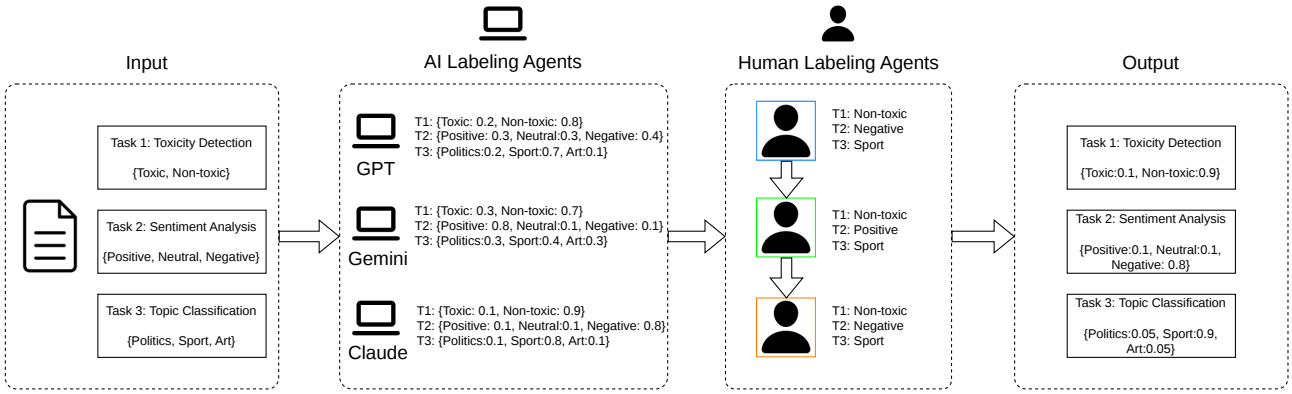


Figure 1: Illustration of the multi-task labeling process we focus on in this paper

identical errors. Therefore, capturing the correlations among multiple models–specifically their shared latent archetypes or families–is important for an accurate and efficient labeling system.

In this paper, we address these challenges by proposing CalCo, a comprehensive framework for multi-task, multi-LLM, and human-in-the-loop annotation. Our main contributions are as follows:

- We introduce a hierarchical Bayesian modeling approach for multi-task labeling processes. Unlike previous works that often treat tasks or annotators independently, our model explicitly captures the correlations across tasks and across models, which allows for more robust inference of ground truth labels and better estimation of annotator/model reliability.
- We report extensive empirical results on both simulated data and real-world applications. We demonstrate that CalCo significantly outperforms baseline aggregation methods in terms of accuracy and efficiency, particularly in scenarios involving correlated tasks and diverse model ensembles.
- We describe a large-scale deployed system that integrates LLM-generated labels into existing human labeling processes in a principled, statistically grounded manner. This system manages the trade-off between efficiency and quality, dynamically allocating human effort where it is most needed.

2 Related Work

The emergence of LLMs has shifted data annotation paradigms toward "LLMs-as-Labelers" and "LLMs-as-Judges," where models generate or evaluate data at scale [3, 4, 10, 34]. While promising, replacing human judgment entirely is complicated by issues such as bias and the "Parrot Dilemma" [22]. Consequently, the field is moving toward hybrid architectures that integrate LLMs with human-in-the-loop processes. Systems such as MEGAnno+ [14] and CoAnnotating [18] facilitate collaborative annotation, leveraging the complementary strengths of human reviewers and AI [17, 33, 41]. Research has also focused on allocation strategies, determining when to rely on models versus humans based on correctness likelihood and trust [20, 23].

Incorporating LLM signals effectively requires addressing their non-deterministic nature. Prior work emphasizes the importance of confidence elicitation and calibration to ensure model outputs are reliable for decision-making [8, 42]. Techniques such as self-training [19] and verbalized confidence [36] have been explored to improve uncertainty expression. However, relying on uncalibrated annotations can lead to confident yet erroneous conclusions [9], necessitating robust frameworks to manage these risks [40].

Researchers have extensively utilized Bayesian frameworks to merge diverse signals from humans and models. Foundational approaches like the Dawid-Skene model [6] estimate observer error rates but often treat annotators as independent. Recent methods combine human predictions with model probabilities via confusion matrices and calibration [13].

CalCo extends the literature on correlated consensus prediction. Previous studies have explored Bayesian combination of probabilistic classifiers using multivariate normal mixtures [27] and explicit models of correlation for classifier fusion [38]. Recent work on Bayesian inference for correlated experts [12] models expert correlation via joint latent representations to minimize human queries. Similarly, online learning frameworks have been developed for consensus prediction to handle streaming data [32]. CalCo builds on these by employing Logistic-Normal distributions [2, 15] to capture task prevalence dependencies, and introduces hierarchical structures to account for shared error profiles in LLM families.

3 Background and Motivation

We formally motivate our design by analyzing two statistical phenomena inherent to hybrid labeling: (1) the limitations of aggregating correlated agents (specifically LLMs), which leads to underestimated variance if unaddressed; and (2) the opportunity to reduce sample complexity by leveraging correlations between tasks.

3.1 Preliminaries

We consider a multi-task classification setting aiming to infer ground truth labels for a dataset of N items, denoted by $\mathcal{I} = \{1, ..., N\}$. Each item i is associated with T distinct tasks. For each task $t \in \{1, ..., T\}$, the output space is a categorical set of K_t unique classes. The true latent label for item i on task t is denoted by $y_{i,t} \in \{1, ..., K_t\}$.

Since $y_{i,t}$ is unobserved, we rely on noisy annotations from two distinct sets of labeling agents:

(1) **Human Agents** (or reviewers/annotators): Let $\mathcal{A} = \{1, ..., A\}$ denote the set of human annotators who provide discrete labels $\{r_{i,t,j}\}$.
(2) **AI Agents** (or LLMs/models/classifiers): Let $\mathcal{M} = \{1, ..., M\}$ denote the set of AI models which provide continuous score vector $\{s_{i,t,m}\}$.

Our objective is to infer the posterior distribution of $y_{i,t}$ and the reliability parameters of all agents given the observed human labels $\mathcal{R}$ and AI scores $\mathcal{S}$. A summary of the key notations used throughout this paper is provided in Table 1.

Notation	Description
N, T	Number of items and tasks
K_t	Number of classes for task t
$\mathcal{A}, A$	Set and count of Human agents
$\mathcal{M}, M$	Set and count of AI agents
J_i	Number of human labels item i receives
F	Number of latent AI agent archetypes
$y_{i,t} \in \{1, \ldots, K_t\}$	Latent true label for item i, task t
$r_{i,t,j} \in \{1, \ldots, K_t\}$	j-th discrete label for task t of item i
$a_{i,j} \in \{1, \ldots, A\}$	The ID of the Human agent providing the j-th annotation for item i
$\boldsymbol{s}_{i,t,m} \in \Delta^{K_t-1}$	Continuous score vector from AI agent m
$\theta_{i,t}$	Item-specific class prevalence vector
$\phi_{a,t}$	Confusion matrix for Human agent a
$\psi_{m,t}$	Confusion matrix for AI agent m

Table 1: Key notations used in the CalCo framework.

3.2 The Opportunity of Task Dependency

As discussed in Section 1, multi-task annotation scenarios—where agents (AI or human) are asked to assess multiple related aspects of a given "job"—are common in real-world applications. For example, evaluating social media posts for topic, sentiment, and offensiveness. These tasks are often dependent: the prevalence or difficulty of one task can inform another. Modeling task dependencies allows us to use annotations from one task to improve inference and reduce effort for related, less certain tasks. Task correlation thus enables more efficient annotation by sharing information across tasks.

To formalize this intuition, we consider a scenario involving two dependent tasks A and B, without loss of generality to multi-task settings. Following the notations in Table 1, the distributions of the unobserved true labels y_A and y_B are governed by the prevalence vector θ_A and θ_B, respectively, with $y_A \sim \text{Dirichlet}(\theta_A)$, $y_B \sim \text{Dirichlet}(\theta_B)$. The task dependency forms the basis for the following result on variance reduction.

Proposition 1 (Variance Reduction via Task Correlation). *Acquiring information about Task A reduces the uncertainty in inferring the other Task B, that is*

$$\mathbb{E}\left[Var\left(\theta_B \mid \theta_A\right)\right] \leqslant Var\left(\theta_B\right)$$

and the reduction in variance is $Var\left(\mathbb{E}\left[\theta_B \mid \theta_A\right]\right)$, the part of variance explained by B.

Remark 1. *The proof follows directly from the law of total variance. In our proposal, as detailed in Section 4.1, we model the task dependency by introducing a latent vector q and constructing a multivariate Gaussian to capture inter-task dependencies. This approach circumvents the simplex constraint on θ. For binary tasks, we have*

$$\begin{pmatrix} q_A \\ q_B \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}, \begin{pmatrix} \sigma_A^2 & \rho\sigma_A\sigma_B \\ \rho\sigma_A\sigma_B & \sigma_B^2 \end{pmatrix}\right)$$

and the conditional variance reduction simplifies to the classical result

$$Var(q_B \mid q_A) = \sigma_B^2(1-\rho^2) \leqslant \sigma_B^2 = Var(q_B)$$

indicating that the uncertainty of Task B is strictly reduced by observing Task A, provided that the two tasks are dependent (i.e., $\rho \neq 0$). The reduction factor ρ^2 represents explained variance by shared information from Task A.

While the theoretical result is not surprising, the intuition on variance reduction under dependency underscores both the benefit and necessity of explicitly modeling the dependency structure; this insight directly motivates our modeling approach. By capturing these dependencies, we can leverage information from related tasks to achieve a target confidence threshold with fewer direct samples than would be required under an independence assumption.

3.3 The Challenge of AI Agent Correlation

In contrast to the beneficial role of task dependency, correlation among AI agents introduces a distinct challenge: if ignored or left unmodeled, it drives uncertainty misleadingly low, producing overconfident and biased estimates. With the score vectors from multiple agents, a common approach in ensemble methods and meta-analysis is to aggregate scores via a weighted average, where the weights are derived from the estimated variance of each source—typically by taking the inverse variance and normalizing [11]. While this strategy appropriately accounts for heterogeneous reliability across sources, it overlooks the dependency structure among agents. When the assumption is violated, this estimation procedure can yield arbitrarily poor uncertainty quantification, even asymptotically. We formalize this phenomenon in the following theorem, which establishes that the variance underestimation induced by ignoring cross-agent correlation does not vanish as the number of agents grows.

Consider M AI agents producing score vectors $\{s_m\}_{m=1}^M \in \Delta^{K-1}$. Let $\boldsymbol{V}_m = \text{Cov}(s_m)$ and $\boldsymbol{V}_{mn} = \text{Cov}(s_m, s_n)$. Denote the weighted average ensemble as $\bar{s}_w = \sum_{m=1}^M w_m s_m$, where the weights $\{w_m\}_{m=1}^M$ satisfy

Assumption 1 (Regularity Conditions). *(R1) Normalization: $\sum_{m=1}^{M} w_m = 1$; (R2) Non-negativity: $w_m \geq 0$ for all m; (R3) Bounded influence: $w_m \leq C/M$ for some constant $C > 0$*

Furthermore, assume persistent positive cross-correlation among agents:

Assumption 2 (Persistent Covariance). *(C1) There exist $\tau_{\min} > 0$ and a positive definite matrix $\boldsymbol{V}_{\min} \succ \boldsymbol{0}$ such that*

$$\frac{1}{2}\left(\boldsymbol{V}_{mn} + \boldsymbol{V}_{mn}^{\top}\right) \succeq \tau_{\min} \cdot \boldsymbol{V}_{\min} \quad \text{for all } m \neq n. \tag{1}$$

Under these assumptions, we have:

Theorem 1 (Persistent Variance Underestimation under Agent Correlation). *Define the true and naive covariance matrices of the weighted average as:*

$$\boldsymbol{V}_{\text{true}} = \sum_{m=1}^{M}\sum_{n=1}^{M} w_m w_n \boldsymbol{V}_{mn}, \qquad \boldsymbol{V}_{\text{naive}} = \sum_{m=1}^{M} w_m^2 \boldsymbol{V}_m. \tag{2}$$

Then the bias matrix $\boldsymbol{\Delta}_M := \boldsymbol{V}_{\text{true}} - \boldsymbol{V}_{\text{naive}}$ *satisfies:*

$$\lim_{M\to\infty} \boldsymbol{\Delta}_M \succeq \tau_{\min} \cdot \boldsymbol{V}_{\min} \succ \mathbf{0} \tag{3}$$

In particular:

(1) (Non-vanishing total bias) $\lim_{M\to\infty} \operatorname{tr}(\boldsymbol{\Delta}_M) \geq \tau_{\min} \cdot \operatorname{tr}(\boldsymbol{V}_{\min}) > 0$*;*

(2) (Divergent relative error) $\lim_{M\to\infty} \frac{\|\boldsymbol{\Delta}_M\|_F}{\|\boldsymbol{V}_{\text{naive}}\|_F} = +\infty.$

A proof is in Appendix 9.2.

Remark 2 (Inverse-variance weighting is insufficient). *The condition (R3) is satisfied by inverse-variance weights whenever the individual variances are lower bounded–no one AI agent gives perfect label. Thus, the practice of variance-based weighting offers no protection against the bias induced by unmodeled correlation.*

Remark 3 (The relative error diverges). *While the naive variance* $\boldsymbol{V}_{\text{naive}} = O(1/n) \to \mathbf{0}$*, the bias* $\boldsymbol{\Delta}_M \to \boldsymbol{V}^* \succ \mathbf{0}$ *remains strictly positive. Consequently, the relative underestimation*

$$\frac{\|\boldsymbol{V}_{\text{true}} - \boldsymbol{V}_{\text{naive}}\|_F}{\|\boldsymbol{V}_{\text{naive}}\|_F} \to +\infty \tag{4}$$

grows without bound—a concerning pathology for large-scale agent ensembles.

4 Modeling Approach

We address the challenges and opportunities identified in Section 3 via CalCo, a hierarchical Bayesian model. Our design directly maps to the motivations above:

(1) **Bias Correction:** By inferring latent confusion matrices $\psi_{m,t}$ and $\phi_{a,t}$, CalCo mathematically inverts the systematic bias identified in Section 3.3, aligning with the "misclassification-adjusted estimator" recommended by Lee et al. [16].
(2) **Variance Management:** By grouping models into archetypes with precision γ, CalCo constrains the effective sample size, respecting the variance floor caused by agent correlations (Theorem 1).
(3) **Efficiency:** By learning the covariance matrices between tasks Σ_θ, CalCo enables the borrowing of statistical strength (Proposition 1), allowing for accurate inference on sparse tasks by leveraging data from dense, correlated tasks.

4.1 Modeling Item-Specific Task Prevalence

Standard aggregation methods (e.g., Dawid-Skene [6] and its extensions [25, 26]) often assume a single global class prevalence vector for each task. However, in multi-task settings, the prevalence of labels often varies significantly from item to item based on the item's latent context. For example, an item about "Politics" (Task A) is far more likely to be "Toxic" (Task B) than an item about "Sports".

To capture these item-specific dependencies, we model a local prevalence vector $\theta_{i,t} \in \Delta^{K_t-1}$ for each item i and task t. To enable information transfer across tasks, we model the correlations between these vectors in a transformed continuous space, similar to prior work [12, 15, 27] leveraging the Logistic Normal distribution [2]. We map each probability vector $\theta_{i,t}$ to the unconstrained real space using the Additive Logistic Transformation (ALT). Let $\mathbf{q}_{i,t} \in \mathbb{R}^{K_t-1}$ be the transformed vector for item i on task t:

$$q_{i,t,k} = \log\left(\frac{\theta_{i,t,k}}{\theta_{i,t,K_t}}\right) \tag{5}$$

We concatenate the vectors for all T tasks into a single latent vector $\mathbf{q}_i = [\mathbf{q}_{i,1}, \ldots, \mathbf{q}_{i,T}]^\top \in \mathbb{R}^{K-T}$. We assume that for every item i, this vector is drawn from a global multivariate Normal distribution that captures the task correlations:

$$\mathbf{q}_i \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \Sigma_\theta) \tag{6}$$

Here, $\boldsymbol{\mu}_\theta$ represents the global average prevalence in ALT space, and Σ_θ captures the covariance structure (e.g., the correlation between "Politics" in Task 1 and "Toxic" in Task 2). We model Σ_θ using an LKJ-Cholesky prior to efficiently learn these correlations:

$$\mathbf{L}_\theta \sim \text{LKJCholesky}(\eta_\theta) \tag{7}$$

$$\Sigma_\theta = \operatorname{diag}(\boldsymbol{\sigma}_\theta)\mathbf{L}_\theta\mathbf{L}_\theta^\top\operatorname{diag}(\boldsymbol{\sigma}_\theta) \tag{8}$$

Finally, the item-specific prevalence $\theta_{i,t}$ is recovered via the inverse ALT (Softmax):

$$\theta_{i,t} = \text{Softmax}([\mathbf{q}_{i,t}, 0]^\top) \tag{9}$$

This formulation allows statistical strength to be shared: observing a hard label for Task A updates the posterior for $\mathbf{q}_i$, which, via Σ_θ, shifts the prior $\theta_{i,t}$ for Task B, enabling robust transfer learning.

4.2 Modeling Human Agent Performance

Following standard practice in the crowdsourcing literature [6, 25, 26], we employ a *confusion matrix* to model the performance of each human agent. Specifically, for each reviewer $a \in \{1, \cdots, A\}$ and task t, the confusion matrix $\phi_{a,t} \in \mathbb{R}^{K_t \times K_t}$ characterizes classification behavior. Each row $k \in \{1, \cdots, K_t\}$ of this matrix represents a probability distribution over the K_t possible labels given that the true label is k. We assume these rows are independently drawn from a Dirichlet distribution:

$$\phi_{a,t,k} \sim \text{Dirichlet}(\beta_\phi) \tag{10}$$

where β_ϕ is a hyperparameter.

4.3 Modeling AI Agent Performance and Their Correlations

For an item i and task t, each AI agent $m \in \{1, \cdots, M\}$ outputs a score vector $s_{i,t,m} \in \mathbb{R}^{K_t}$. Similar to our treatment of human agents, we characterize the performance of AI agent m for task t via a *confusion matrix* $\psi_{m,t} \in \mathbb{R}^{K_t \times K_t}$. To account for the fact that LLMs and ML models are often derived from similar architectures or trained on overlapping datasets, we consider approaches to capture the *inter-model correlations*. Specifically, for each task $t \in [1, T]$ and each true label $k \in [1, K_t]$, we capture the dependencies among the probability vectors $\{\psi_{m,t,k}\}_{m=1}^{M}$, which allows CalCo to learn, e.g., if two specific LLMs are likely to make the same type of classification error.

We consider a hierarchical approach by assuming the M AI agents belong to F *archetypes* (or families), where $F \ll M$. Each

archetype $f \in \{1, \ldots, F\}$ is characterized by a *base confusion matrix* whose k-th row is drawn from a global prior:

$$\bar{\psi}_{f,t,k} \sim \text{Dirichlet}(\beta_\psi) \tag{11}$$

where β_ψ is a hyperparameter similar to β_ϕ above.

The specific confusion matrix for AI agent m, belonging to archetype z_m, is then generated as a concentrated perturbation of its archetype's base matrix. The degree of similarity is controlled by a precision parameter $\gamma > 0$:

$$\psi_{m,t,k} \sim \text{Dirichlet}(\gamma \cdot \bar{\psi}_{z_m,t,k}) \tag{12}$$

In this formulation, a large γ implies that models within the same family have highly similar performance characteristics, whereas a small γ allows for greater intra-archetype variation.

In scenarios where the specific mapping of models to archetypes is unknown, we treat the archetype assignment $z_m \in \{1, \cdots, F\}$ as a latent variable to be inferred. We assume a global mixing distribution $\pi \in \Delta^{F-1}$ drawn from a symmetric Dirichlet, from which the archetype assignment for AI agent m is sampled as:

$$\pi \sim \text{Dirichlet}(\alpha) \qquad z_m \sim \text{Categorical}(\pi) \tag{13}$$

where α is a hyperparameter.

4.4 Generating the Observed Data

For each item i and task t, we assume a latent true discrete label $y_{i,t}$ drawn from the prevalence $\theta_{i,t}$: $y_{i,t} \sim \text{Categorical}(\theta_{i,t})$.

Let $a_{i,j}$ denote the human reviewer providing the j^{th} annotation for item i. The observed label for task t is generated according to the reviewer's the confusion matrix $\phi_{a_{i,j},t}$ conditioned on the item's latent true label $y_{i,t}$, as follows:

$$r_{i,t,j} \sim \text{Categorical}\left(\phi_{a_{i,j},t,y_{i,t}}\right) \tag{14}$$

Similarly, the performance of AI agent m on task t is governed by its confusion matrix $\psi_{m,t}$. Since the AI agent outputs a continuous score vector $s_{i,t,m} \in \mathbb{R}^{K_t}$ (where $\sum_k s_{i,t,m,k} = 1$) rather than a discrete label, we model $s_{i,t,m}$ using a Dirichlet distribution centered on the corresponding row of the confusion matrix:

$$s_{i,t,m} \sim \text{Dirichlet}\left(\lambda \cdot \psi_{m,t,y_{i,t}}\right) \tag{15}$$

5 Posterior Inference

Given the observed human labels $\mathbf{R} = \{r_{i,t,j}\}$ and AI scores $\mathbf{S} = \{s_{i,t,m}\}$, we seek to infer the posterior distribution of the latent variables $\mathbf{Z} = \{\Sigma_\theta, \theta, \phi, \psi\, \pi\}$. Exact inference is intractable due to the non-conjugacy of the Logistic-Normal prior and the hierarchical dependencies. We employ Stochastic Variational Inference (SVI) to approximate the posterior $p(\mathbf{Z} \mid \mathbf{R}, \mathbf{S})$ with a variational family $Q_\xi(\mathbf{Z})$ parameterized by ξ. To ensure scalability for large-scale production data, we use two key techniques: (1) *Amortized Variational Inference* [7, 21] using a neural inference network for item-specific prevalence $\{\theta_{i,t}\}$, and (2) *Collapsed Inference* to marginalize out the discrete latent variables $\{y_{i,t}\}$ and $\{z_m\}$.

5.1 Variational Approximation

We assume a mean-field factorization for the global and agent-specific parameters, while retaining the covariance structure for item-specific parameters. The variational posterior is factorized as:

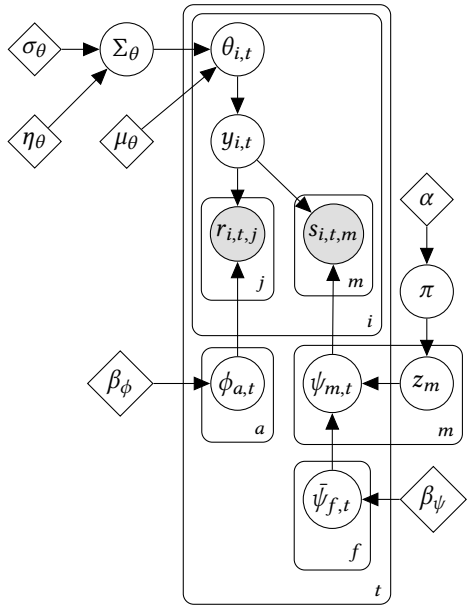


Figure 2: Graphical representation of the hierarchical Bayesian model used in CalCo. Circle nodes represent random variables, shaded nodes are observed, diamond nodes represent hyperparameters, edges are probabilistic dependencies, and plates represent repetition.

$$Q(\mathbf{Z}) = \underbrace{Q(\Sigma_\theta)}_{\text{Correlation}} \cdot \underbrace{\left[\prod_{i=1}^{N} Q_\xi(\theta_i \mid \mathbf{s}_i)\right]}_{\text{Amortized Prevalence}} \cdot \underbrace{\left[\prod_{a=1}^{A} Q(\phi_a)\right]}_{\text{Human Agents}}$$
$$\cdot \underbrace{Q(\boldsymbol{\pi}) \cdot \left[\prod_{f=1}^{F} Q(\bar{\psi}_f)\right]}_{\text{Archetypes}} \cdot \underbrace{\left[\prod_{m=1}^{M} Q(\psi_m)\right]}_{\text{AI Agents}}$$

The specific variational distributions are defined as follows:

Global Correlations ***($Q(\Sigma_\theta)$).*** We approximate the posterior of the task correlation matrix using a point estimate (Delta distribution) on the Cholesky factor $\mathbf{L}_\theta$, constrained such that $\Sigma_\theta = \text{diag}(\sigma)\mathbf{L}_\theta\mathbf{L}_\theta^\top\text{diag}(\sigma)$.

Agent Parameters ***($Q(\phi), Q(\psi), Q(\bar{\psi}), Q(\boldsymbol{\pi})$).*** For every human agent a and AI agent m, the variational posteriors for their confusion matrix rows are modeled as independent Dirichlet distributions with learnable concentration parameters. Similarly, the archetype mixing weights $\boldsymbol{\pi}$ and base confusion matrices $\bar{\psi}$ are governed by variational Dirichlet distributions.

Amortized Item Prevalence ***($Q_\nu(\theta_i|\mathbf{s}_i)$).*** Instead of learning free variational parameters for every item i (which scales linearly with N), we employ an *Inference Network* (Encoder) parameterized by ν. The encoder takes the aggregated AI agent scores $\mathbf{s}_i$ as input and outputs the location parameter $\boldsymbol{\mu}_i$ for the latent prevalence in ALT-space. To preserve the learned task correlations, the variational

distribution for item i shares the global covariance structure:

$$Q(\mathbf{q}_i \mid \mathbf{s}_i) = \mathcal{N}(\text{Encoder}_\nu(\mathbf{s}_i), \Sigma_\theta) \tag{16}$$

where $\mathbf{q}_i$ is the ALT-transformed prevalence vector. This amortization reduces the complexity of the variational parameters from $O(N \times (K-T))$ to $O(|\nu|)$, the constant size of the encoder network, thereby decoupling CalCo's complexity from the dataset size.

Additionally, the Encoder enables our framework to condition the latent prevalence on any item-specific feature vector $\boldsymbol{x}_i$. This formulation generalizes a rich line of prior work that models the dependency between instance features and ground truth labels using fixed architectures, such as logistic regression [28, 29] and Gaussian Processes [24, 30, 31]. In the specific case where $\boldsymbol{x}_i$ consists of the aggregated AI agent scores (as implemented in our experiments in Section 7.3), the Encoder functions as a non-linear calibration mechanism. This extends the approach of Kerrigan et al. [13], which aligns raw model scores with human labels using a restricted, single-parameter Temperature Scaling function.

5.2 Marginalization of Discrete Latent Variables

We utilize Stochastic Variational Inference (SVI) to optimize the ELBO. Standard gradient estimators for SVI (such as the reparameterization trick) require latent variables to be continuous and differentiable. Our model, however, contains two sets of discrete latent variables: the ground truth labels $y_{i,t}$ and the AI agent archetype assignments z_m. To enable fully differentiable training, we analytically marginalize these discrete variables from the objective.

Marginalizing Ground Truth *($y_{i,t}$).* For every item i and task t, the true label $y_{i,t}$ is a categorical variable. We sum over the K_t possible classes to compute the marginal likelihood of the observed human labels $\mathbf{r}_{i,t}$ and AI scores $\mathbf{s}_{i,t}$. This is efficiently implemented using the `LogSumExp` operation: $\log p(\boldsymbol{r}_{i,t}, \mathbf{s}_{i,t} \mid \theta_{i,t}, \phi, \psi) =$

$$\log \sum_{k=1}^{K_t} \left[p(y_{i,t}=k \mid \theta_{i,t}) \prod_{j=1}^{J_i} p(r_{i,t,j} \mid y_{i,t}=k, \phi) \prod_{m=1}^{M} p(\mathbf{s}_{i,t,m} \mid y_{i,t}=k, \psi) \right]$$

Marginalizing Archetype Assignments *(z_m).* When the mapping between AI agents and archetypes is unknown, $z_m \in \{1, \ldots, F\}$ is a discrete latent variable. Instead of sampling z_m, we model the generation of the agent's confusion matrix ψ_m using a Mixture Model. We marginalize out the assignment by summing over all F possible archetypes, weighted by the mixing proportions $\boldsymbol{\pi}$:

$$p(\psi_m \mid \boldsymbol{\pi}, \bar{\Psi}) = \sum_{f=1}^{F} \pi_f \cdot \text{Dirichlet}(\psi_m \mid \gamma \cdot \bar{\psi}_f) \tag{17}$$

This exact marginalization allows gradients to flow directly to the continuous distributional parameters $(\theta, \phi, \bar{\Psi}, \boldsymbol{\pi})$ without requiring high-variance score function estimators for the discrete variables.

5.3 Optimization

We maximize the Evidence Lower Bound (ELBO) using the Adam optimizer. The training procedure utilizes *mini-batch subsampling* of items, enabling the model to train on large-scale datasets that do not fit in memory.

To improve convergence, we employ an *empirical Bayes* initialization strategy. Before SVI begins, we compute an empirical correlation matrix from the raw model scores and human votes to initialize the prior Σ_θ. This provides a warm-start for the covariance estimation, stabilizing the optimization of the complex objective. The full training procedure is summarized in Algorithm 1.

6 Evaluations on Simulated Data

6.1 Empirical Verification of Theorem 1

To complement the theoretical analysis, we empirically verify the three predictions of Theorem 1 via a controlled simulation. We consider M AI agents producing K-dimensional score vectors with equal weights $w_m = 1/M$, individual covariance $\boldsymbol{V}_m = \sigma^2 \boldsymbol{I}_K$, and uniform pairwise cross-covariance $\boldsymbol{V}_{mn} = \rho\sigma^2 \boldsymbol{I}_K$ for all $m \neq n$. We set $\sigma^2 = 0.1$, $K = 3$, vary $M \in \{2, 3, 5, 10, 20, 50, 100, 200, 500\}$, and sweep $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

Table 2 reports the numerical results for $\rho = 0.5$, and Figure 3 visualizes all three predictions across all correlation levels:

(1) **Divergent relative error** (Figure 3a): The ratio $\|\boldsymbol{\Delta}_M\|_F / \|\boldsymbol{V}_{\text{naive}}\|_F$ grows as $\rho(M-1)$, reaching 249.5 at $M = 500$ for $\rho = 0.5$. This confirms that the naive variance estimate becomes arbitrarily wrong as the number of agents increases.
(2) **Non-vanishing bias** (Figure 3b): The trace $\text{tr}(\boldsymbol{\Delta}_M)$ saturates at $\rho\sigma^2 K = 0.15$, a strictly positive constant independent of M. Even with $M = 500$ agents, the bias remains at 0.150—it does not diminish with more data sources.
(3) **Vanishing naive estimate** (Figure 3c): While $\text{tr}(\boldsymbol{V}_{\text{naive}}) \to 0$ as σ^2/M, the true variance $\text{tr}(\boldsymbol{V}_{\text{true}})$ remains bounded above $\rho\sigma^2 K$. At $M = 500$, the naive estimate (0.0006) understates the true variance (0.1503) by a factor of 250×.

These results empirically validate the necessity of modeling agent correlations: simply adding more LLMs without accounting for their shared error profiles leads to arbitrarily overconfident estimates. This motivates CalCo's archetype-based approach (Section 4.3), which explicitly captures inter-model dependencies to recover calibrated uncertainty.

M	$\|\boldsymbol{\Delta}_M\|_F / \|\boldsymbol{V}_{\text{naive}}\|_F$	$\text{tr}(\boldsymbol{\Delta}_M)$	$\text{tr}(\boldsymbol{V}_{\text{naive}})$
2	0.50	0.075	0.150
5	2.00	0.120	0.060
10	4.50	0.135	0.030
50	24.50	0.147	0.006
100	49.50	0.149	0.003
500	249.50	0.150	0.001

Table 2: Empirical verification of Theorem 1 ($\rho = 0.5$, $\sigma^2 = 0.1$, $K = 3$). The relative error grows as $\rho(M-1)$, the bias $\text{tr}(\boldsymbol{\Delta}_M)$ converges to $\rho\sigma^2 K = 0.15$, and $\text{tr}(\boldsymbol{V}_{\text{naive}}) \to 0$ as σ^2/M.

6.2 Synthetic Data Simulation Process

To evaluate CalCo against baselines, we simulate synthetic data using a generative process similar to Section 4, but with fixed ground-truth correlations. Crucially, we generate inter-task and inter-model dependencies using Multivariate Normal distributions given input correlation matrices $(\Omega_\theta, \Omega_\psi)$. These parameters were selected to match the empirical statistics observed in our large-scale production data, ensuring the evaluation reflects realistic scenarios.

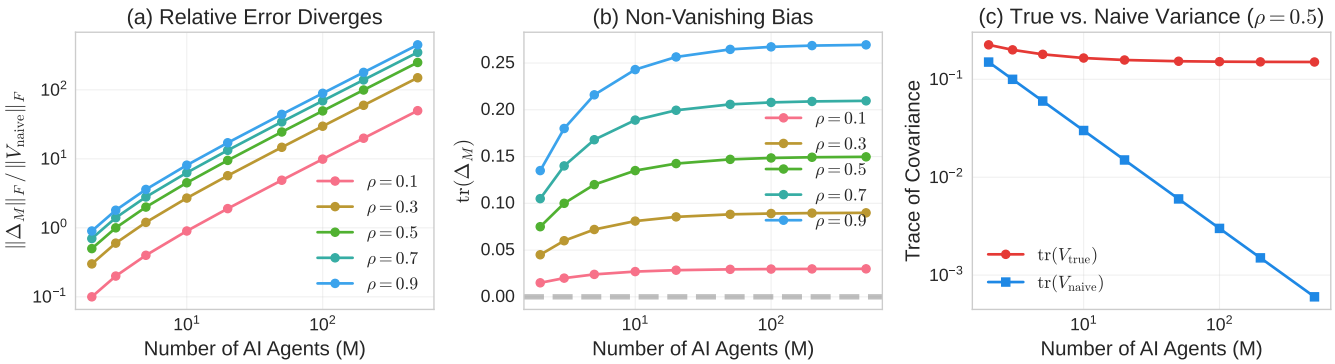


Figure 3: Empirical verification of Theorem 1. (a) The relative error $\|\Delta_M\|_F / \|V_{\text{naive}}\|_F$ diverges as M grows, for all correlation levels ρ. (b) The bias $\text{tr}(\Delta_M)$ saturates at $\rho\sigma^2 K$ (a positive constant), confirming non-vanishing total bias. (c) For $\rho = 0.5$: the naive variance $\text{tr}(V_{\text{naive}}) \to 0$ while the true variance $\text{tr}(V_{\text{true}})$ remains bounded—producing unbounded relative underestimation.

Baseline	Description
MV	**Majority Vote**: Computes the mode of the observed labels from all agents (humans and models treated equally).
Disc-DS	**Discrete Dawid-Skene**: Converts continuous model scores into discrete labels and learns a confusion matrix for each AI and human agent using Expectation-Maximization [6].
Cal-DS	**Calibrated Dawid-Skene**: Based on Kerrigan et al. [13], this method combines calibrated model probabilities with human confusion matrices, weighting the model as a prior relative to human annotations.
HAIC	**Human-AI Complementarity**: Based on Steyvers et al. [33], this model captures the joint distribution of human and AI confidence signals to identify complementarity.

Table 3: Baseline approaches compared in our evaluation.

Specifically, we generate correlated ground truth labels by sampling item-specific prevalence vectors in the Additive Logistic Transform (ALT) space from $\mathcal{N}(\boldsymbol{\mu}_\theta, \Sigma_\theta)$ and mapping them to the simplex via softmax. Similarly, to capture shared error profiles among models, we generate correlated accuracies by sampling latent ability vectors from a Multivariate Normal distribution parameterized by Ω_ψ and projecting them to probabilities using the standard normal CDF. Further details are provided in the Appendix.

6.3 Efficiency Optimization via Early Stopping

In industrial hybrid labeling workflows, querying human reviewers is the primary resource constraint. A core value proposition of CalCo is its ability to convert diverse signals—continuous LLM scores $\mathbf{s}_{i,t}$ and sparse human labels $r_{i,t}$—into a calibrated posterior probability $P(y_{i,t}|\mathcal{D})$. We evaluate the Efficiency-Quality trade-off by simulating an iterative labeling process where the system dynamically decides whether to request another human label or terminate the reviewing process based on its current confidence.

6.3.1 Efficiency-Quality Trade-off. To quantify this trade-off, we define a confidence score for each item i on task t after observing j human labels, denoted as $C_{i,t}^{(j)}$. Here, j ranges from 0 (only AI scores) to J_i (all available human labels). The confidence is defined as the maximum posterior probability of the predicted class:

$$C_{i,t}^{(j)} = \max_k P(y_{i,t} = k \mid \mathcal{D}^{(j)}) \tag{18}$$

We generate trade-off curves by systematically varying a global confidence threshold τ. The labeling process for item i is terminated at step j if the confidence for *all* associated tasks exceeds this threshold (i.e., $\min_t C_{i,t}^{(j)} > \tau$). We compute **Efficiency** as the proportion of human labels saved relative to the maximum budget, and **Quality** using (1) *Accuracy* ($\frac{1}{N \cdot T} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbb{I}\left[\hat{y}_{i,t} = y_{i,t}\right]$), and (2) *Log Loss* ($-\frac{1}{N \cdot T} \sum_{i=1}^{N} \sum_{t=1}^{T} \log p(\hat{y}_{i,t} = y_{i,t} \mid \mathcal{D})$)

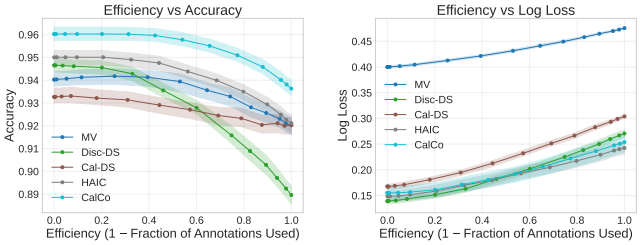


Figure 4: Efficiency-Quality trade-off on simulated data ($N = 10K$ items, $T = 10$ correlated tasks).

Results Analysis. Figure 4 illustrates the trade-off for a dataset with $N = 10K$ items and $T = 10$ tasks.

Accuracy: In the high-efficiency regime (where > 80% of human labels are skipped), CalCo maintains accuracy comparable to the fully-labeled baseline, consistently outperforming HAIC and significantly surpassing standard aggregation methods (Disc-DS,

MV). This confirms that by modeling inter-task correlations, CalCo effectively leverages information across tasks to maintain high classification performance even with sparse human data.

Log Loss: HAIC achieves the best Log Loss across the efficiency spectrum, validating its approach of explicitly modeling the joint correlations between human and AI signals. While CalCo is slightly less calibrated than HAIC, it remains highly competitive and significantly outperforms independent baselines (Disc-DS, MV), reaffirming the importance of capturing inter-agent correlations.

6.3.2 Impact of Task Correlation. A key hypothesis of our framework is that modeling inter-task correlations allows the system to "borrow" statistical strength from easier or more densely labeled tasks to improve inference on harder ones. To verify, we evaluate performance while varying the number of correlated tasks T.

Results Analysis. Figure 5 presents the Area Under the Curve (AUC) for the Efficiency-Accuracy trade-off as a function of T. We observe that CalCo's performance advantage over the baselines widens significantly as the number of tasks increases. This trend confirms that in higher-dimensional settings, the system gains more opportunities to propagate statistical strength from easier or more confidently modeled tasks to harder ones via the learned covariance structure. In contrast, methods that treat tasks independently fail to leverage this information, resulting in performance gains that plateau or degrade as the annotation job becomes more complex.

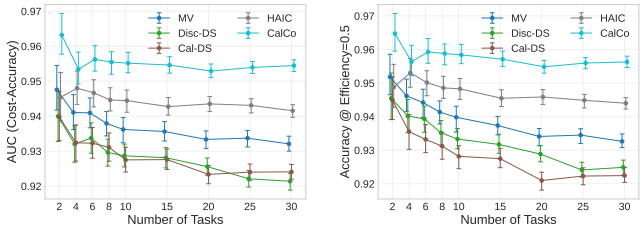


Figure 5: Impact of task dimensionality (T) on labeling efficiency: (Left) Area Under the Curve (AUC) for the Efficiency-Accuracy trade-off curve, and (Right) Accuracy achieved at a fixed 50% efficiency savings.

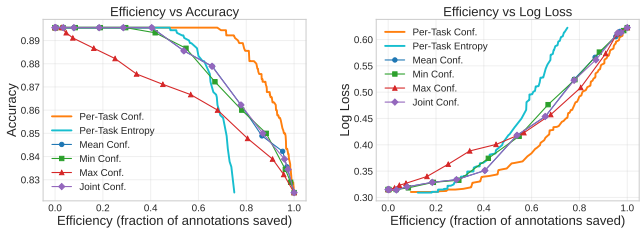


Figure 6: Impact of stopping policies on the Efficiency-Quality trade-off.

6.3.3 Varying the Stopping Policies. The results in Section 6.3.1 use a simple global thresholding policy to demonstrate the baseline efficacy of the posterior. However, CalCo's joint modeling of tasks enables more sophisticated termination criteria that can further optimize the labeling efficiency.

Policy	Condition	Behavior
Minimum	$\min_t C_{i,t} \geq \tau$	High confidence on *all* tasks.
Mean	$\frac{1}{T}\sum_t C_{i,t} \geq \tau$	Average confidence exceeds threshold.
Maximum	$\max_t C_{i,t} \geq \tau$	Stops if *any* task is confident.
Joint	$\prod_t C_{i,t} \geq \tau$	High joint probability of correctness.
Per-Task Confidence	$C_{i,t} \geq \tau_t$	Stops each task independently if it is confident.
Per-Task Entropy	$H(C_{i,t}) \leq \tau_t$	Stops each task independently when the posterior entropy drops below a threshold.

Table 4: Summary of stopping policies. $C_{i,t}$ denotes the model confidence for item i on task t.

Application	#Tasks	MV	Cal-DS	Disc-DS	HAIC	CalCo
Eval	4	0.811	0.832	0.850	0.881	**0.897**
Eval	3	0.825	0.836	0.870	**0.889**	0.882
Eval	5	0.816	0.821	0.884	0.907	**0.912**
IA	1	0.821	0.786	0.797	**0.865**	0.852
IA	6	0.833	0.857	0.906	0.913	**0.925**
CA	8	0.884	0.864	0.980	0.987	**0.990**
CA	8	0.942	0.895	0.957	0.961	**0.983**
CA	7	0.870	0.892	0.947	0.966	**0.976**
CA	7	0.843	0.869	0.854	0.901	**0.925**
CA	8	0.777	0.813	0.887	0.893	**0.917**

Table 5: AUC comparison across 10 real-world use cases. Higher values indicate better efficiency-accuracy tradeoff.

Results Analysis. We evaluated multiple distinct stopping policies (Table 4) to understand how the aggregation of confidence scores affects the trade-off between labeling efficiency and model quality. Figure 6 illustrates the Efficiency-Quality trade-off across various stopping policies. We observe that granular, task-level policies (specifically **Per-Task Confidence**) achieve the most robust Pareto frontier, maintaining near-peak accuracy and the lowest Log Loss even at high efficiency level. This validates our hypothesis regarding task heterogeneity: by decoupling stopping decisions, the system aggressively conserves budget on "easy" sub-tasks while continuing to query experts for uncertain dimensions of the same item. In contrast, coarse job-level policies like **Mean** or **Joint Confidence** force synchronized termination, often wasting budget on resolved tasks or stopping prematurely on difficult ones. Among these, **Minimum Confidence** serves as a reliable conservative baseline, ensuring the least certain task reaches a threshold before stopping the job. Conversely, the **Maximum** policy degrades performance rapidly, confirming that high confidence in one dimension does not imply reliability in another.

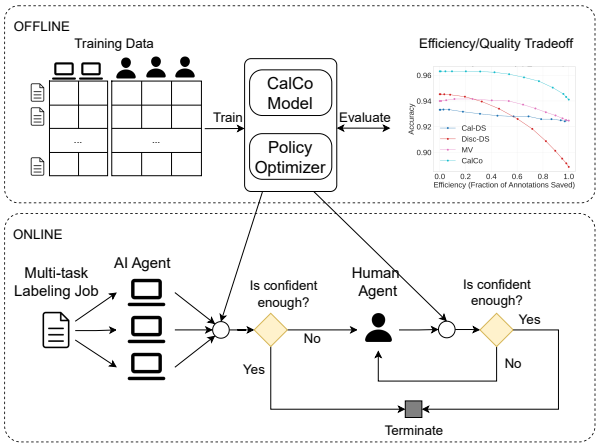


Figure 7: Overview of the CalCo system deployment

7 Evaluation on Real Data

7.1 Deployment

CalCo has been validated on industrial-scale datasets to optimize human labeling efficiency via confidence-based early stopping. Figure 7 provides an overview of the system, which operates in two distinct phases. In the *offline phase*, CalCo fits the hierarchical Bayesian parameters—including agent confusion matrices (ϕ, ψ) and the inter-task correlation matrix (Σ_θ)—using historical annotation data. A *policy optimizer* then simulates annotation sequences on a held-out validation set to identify the optimal stopping policy that satisfies specific business constraints (e.g., minimizing cost subject to a quality floor). In the *online phase*, incoming multi-task jobs are first scored by the ensemble of LLMs. The system then iteratively requests human labels only for specific tasks where the posterior confidence fails to meet the stopping criteria defined by the selected policy, leveraging the learned correlations to infer consensus efficiently.

7.2 Data description

We evaluate CalCo on real datasets spanning 10 annotation use cases, drawn from application areas including content analysis, ML evaluation and intent analysis. The diversity of this dataset—spanning multiple domains, languages–enables evaluation of the proposed approach across varying levels of task difficulty, linguistic complexity, and annotator expertise. Details of the datasets are in Table 6.

Application	#Datasets	$\frac{\sum_i J_i}{N}$	T	N
Content Analysis (CA)	5	2-3	4-8	1k-3k
Evaluation (Eval)	3	2-19	3-5	1k-11k
Intent Analysis (IA)	2	2-4	1-6	1k-9k

Table 6: Real data statistics by application area.

7.3 Empirical Results

Table 5 presents AUC to measure the overall efficiency-quality tradeoff given by various methods. Our method achieves the highest AUC in 8/10 use cases with particularly strong performance in high-task-number settings, highlighting the benefit of incorporating task dependency. HAIC ranks as the second-best method, consistent with our simulation findings. Among the remaining three methods, performance varies by use case, though the two DS variants slightly outperform MV overall.

8 Conclusion

In this work, we introduced CalCo, a comprehensive hierarchical Bayesian framework designed to address the challenges of integrating LLMs into industrial-scale human annotation workflows. Our approach explicitly models the complex dependency structures inherent in multi-task labeling. By capturing inter-task correlations via Logistic-Normal priors, CalCo enables the system to "borrow" statistical strength from data-rich tasks to improve inference on correlated, data-poor tasks. Simultaneously, our hierarchical treatment of AI agents through latent archetypes allows the model to robustly account for the shared error profiles common among LLM families, preventing the overconfidence that plagues traditional aggregation methods. Furthermore, the incorporation of Amortized Variational Inference via a neural encoder significantly improves scalability, decoupling the number of variational parameters from the dataset size and enabling efficient inference on large-scale production data.

Empirical evaluations on both simulated data and diverse real-world datasets demonstrate that CalCo consistently outperforms established baselines. We showed that CalCo achieves a better trade-off between labeling efficiency and model quality. Finally, we demonstrated the flexibility of the framework in supporting various stopping policies, allowing practitioners to dynamically optimize for specific business constraints, whether prioritizing granular task-level savings or strict job-level quality guarantees.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] J. Aitchison and S. M. Shen. 1980. Logistic-Normal Distributions: Some Properties and Uses. *Biometrika* 67, 2 (1980), 261–272.

[3] Ekaterina Artemova, Akim Tsvigun, Dominik Schlechtweg, et al. 2025. Hands-On Tutorial: Labeling with LLM and Human-in-the-Loop. *arXiv preprint arXiv:2411.04637* (2025).

[4] Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of Human Judges? A Large Scale Empirical Study across 20 NLP Evaluation Tasks. In *Proceedings of ACL (Volume 2: Short Papers)*. 238–255.

[5] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. 2019. Pyro: Deep universal probabilistic programming. *Journal of machine learning research* 20, 28 (2019), 1–6.

[6] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* (1979).

[7] Ankush Ganguly, Sanjana Jain, and Ukrit Watchareeruetai. 2023. Amortized variational inference: A systematic review. *Journal of Artificial Intelligence Research* 78 (2023), 167–215.

[8] Jiahui Geng, Fengyu Cai, Yuxia Wang, et al. 2024. A Survey of Confidence Estimation and Calibration in Large Language Models. In *Proceedings of NAACL (Volume 1: Long Papers)*. 6577–6595.

[9] Kristina Gligoric, Tijana Zrnic, Cinoo Lee, et al. 2025. Can Unconfident LLM Annotations Be Used for Confident Conclusions?. In *Proceedings of NAACL (Volume 1: Long Papers)*. 3514–3533.

[10] Xingwei He, Zhenghao Lin, Yeyun Gong, et al. 2024. AnnoLLM: Making Large Language Models to Be Better Crowdsourced Annotators. In *Proceedings of NAACL (Volume 6: Industry Track)*. 165–190.
[11] Larry V Hedges and Ingram Olkin. 2014. *Statistical methods for meta-analysis*. Academic press.
[12] Markelle Kelly, Alex James Boyd, Sam Showalter, Mark Steyvers, and Padhraic Smyth. 2025. Bayesian Inference for Correlated Human Experts and Classifiers. In *Forty-second International Conference on Machine Learning*.
[13] Gavin Kerrigan, Padhraic Smyth, and Mark Steyvers. 2021. Combining Human Predictions with Model Probabilities via Confusion Matrices and Calibration. In *Advances in Neural Information Processing Systems*.
[14] Hannah Kim, Kushan Mitra, Rafael Li Chen, et al. 2024. MEGAnno+: A Human-LLM Collaborative Annotation System. In *Proceedings of EACL: System Demonstrations*. 168–176.
[15] John Lafferty and David Blei. 2005. Correlated Topic Models. In *Advances in Neural Information Processing Systems*, Vol. 18.
[16] Chungpa Lee, Thomas Zeng, Jongwon Jeong, Jy yong Sohn, and Kangwook Lee. 2026. How to Correctly Report LLM-as-a-Judge Evaluations. In *Proceedings of ICML*.
[17] Jiyi Li. 2024. Human-LLM Hybrid Text Answer Aggregation for Crowd Annotations. In *Proceedings of EMNLP*. 15609–15622.
[18] Minzhi Li, Taiwei Shi, Caleb Ziems, et al. 2023. CoAnnotating: Uncertainty-Guided Work Allocation between Human and Large Language Models for Data Annotation. In *EMNLP*.
[19] Shudong Liu, Zhaocong Li, Xuebo Liu, et al. 2024. Can LLMs Learn Uncertainty on Their Own? Expressing Uncertainty Effectively in A Self-Training Manner. In *Proceedings of EMNLP*. 21635–21645.
[20] Shuai Ma, Ying Lei, Xinru Wang, et al. 2023. Who Should I Trust: AI or Myself? Leveraging Human and AI Correctness Likelihood to Promote Appropriate Trust in AI-Assisted Decision-Making. In *Proceedings of CHI*. 1–19.
[21] Charles C. Margossian and David M. Blei. 2024. Amortized variational inference: when and why?. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence (UAI '24)*. Article 115.
[22] Anders Giovanni Møller, Arianna Pera, Jacob Dalsgaard, and Luca Aiello. 2024. The Parrot Dilemma: Human-Labeled vs. LLM-augmented Data in Classification Tasks. In *Proceedings of EACL (Volume 2: Short Papers)*. 179–192.
[23] Hussein Mozannar, Hunter Lang, Dennis Wei, et al. 2023. Who Should Predict? Exact Algorithms For Learning to Defer to Humans. In *Proceedings of AISTATS*. 10520–10545.
[24] Viet-An Nguyen, Peibei Shi, Jagdish Ramakrishnan, Narjes Torabi, Nimar S. Arora, Udi Weinsberg, and Michael Tingley. 2022. Crowdsourcing with Contextual Uncertainty. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3645–3655.
[25] Rebecca J Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics* 2 (2014), 311–326.
[26] Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian Models of Annotation. *TACL* (2018).
[27] Gregor Pirš and Erik Štrumbelj. 2019. Bayesian Combination of Probabilistic Classifiers using Multivariate Normal Mixtures. *Journal of Machine Learning Research* 20, 51 (2019), 1–18.
[28] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *ICML*. 889–896.
[29] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning From Crowds. *JMLR* 11 (Aug. 2010), 1297–1322.
[30] Filipe Rodrigues, Francisco C. Pereira, and Bernardete Ribeiro. 2014. Gaussian Process Classification and Active Learning with Multiple Annotators. In *ICML*.
[31] Pablo Ruiz, Pablo Morales-Álvarez, Rafael Molina, and Aggelos K Katsaggelos. 2019. Learning from crowds with variational Gaussian processes. *Pattern Recognition* 88 (2019), 298–311.
[32] Samuel Showalter, Alex J Boyd, Padhraic Smyth, and Mark Steyvers. 2024. Bayesian online learning for consensus prediction. In *International Conference on Artificial Intelligence and Statistics*. 2539–2547.
[33] Mark Steyvers, Heliodoro Tejeda, Gavin Kerrigan, and Padhraic Smyth. 2022. Bayesian modeling of human–AI complementarity. *Proceedings of the National Academy of Sciences* 119, 11 (2022).
[34] Zhen Tan, Dawei Li, Song Wang, et al. 2024. Large Language Models for Data Annotation and Synthesis: A Survey. In *Proceedings of EMNLP*. 930–957.
[35] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).
[36] Katherine Tian, Eric Mitchell, Allan Zhou, et al. 2023. Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback. In *Proceedings of EMNLP*. 5433–5442.
[37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
[38] Susanne Trick and Constantin Rothkopf. 2022. Bayesian Classifier Fusion with an Explicit Model of Correlation. In *Proceedings of AISTATS*. 2282–2310.
[39] Michelle Vaccaro, Abdullah Almaatouq, and Thomas Malone. 2024. When combinations of humans and AI are useful: A systematic review and meta-analysis. *Nature Human Behaviour* 8, 12 (2024), 2293–2303.
[40] Peiqi Wang, Barbara D. Lam, Yingcheng Liu, Ameneh Asgari-Targhi, Rameswar Panda, William M Wells, Tina Kapur, and Polina Golland. 2025. Calibrating Expressions of Certainty. In *The Thirteenth International Conference on Learning Representations*.
[41] Xinru Wang, Hannah Kim, Sajjadur Rahman, et al. 2024. Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels. In *Proceedings of CHI*.
[42] Miao Xiong, Zhiyuan Hu, Xinyang Lu, et al. 2024. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. In *ICLR*.

9 Appendix

9.1 Notations

Table 7 describes in more detail all the notations used in our paper.

Notation	Description
	Observed Data
$\mathbf{s}_{i,t,m} \in \Delta^{K_t-1}$	Continuous score vector from AI agent m
$r_{i,t,j} \in \{1,\dots,K_t\}$	j-th discrete label for task t of item i
$a_{i,j} \in \{1,\dots,A\}$	The ID of the Human agent providing the j-th annotation for item i
	Latent Variables
$y_{i,t} \in \{1,\dots,K_t\}$	Latent true label for item i, task t
$\theta_{i,t} \in \Delta^{K_t-1}$	Item-specific class prevalence vector
$q_{i,t} \in \mathbb{R}^{K_t-1}$	Item-specific class prevalence vector in ALT space
$\phi_{a,t}$	Confusion matrix for human agent a on task t
$\psi_{m,t}$	Confusion matrix for AI agent m on task t
$\bar{\psi}_{f,t}$	Confusion matrix for archetype f on task t
$z_m \in \{1,\dots,F\}$	Index assignment indicating which archetype that model m belongs to
	Hyperparameters
$\mu_\theta, \sigma_\theta, \eta_\theta$	Hyperparameters (Mean, Scale, LKJ shape) for the Multivariate Normal prior governing task prevalence correlations
β_ϕ	Dirichlet concentration parameter for Human Agent confusion matrices
β_ψ	Dirichlet concentration parameter for Archetype confusion matrices
γ	Precision parameter controlling the deviation of an AI Agent's confusion matrix from its archetype
α	Dirichlet concentration parameter for the archetype mixing distribution π
λ	Scaling concentration parameter for the AI Agent score likelihood (Dirichlet distribution)

Table 7: Notations used in this paper.

Algorithm 1: Amortized SVI for posterior inference

Input: Data $\mathcal{D} = \{(\mathbf{r}_i, \mathbf{s}_i)\}_{i=1}^N$, number of archetypes F, batch size B, learning rate η
Output: Variational parameters ξ, Encoder weights ν
Initialize: Variational parameters for global (Σ_θ), agents (ϕ, ψ), and archetypes ($\boldsymbol{\pi}, \bar{\Psi}$); Initialize Encoder ν.
Initialize Σ_θ using Empirical Bayes from $\mathcal{D}$
while *not converged* **do**
  `// Sample Global & Agent Variational Parameters`
  Sample correlations $\hat{\Sigma}_\theta \sim \boldsymbol{Q}(\Sigma_\theta)$
  Sample archetype weights $\hat{\boldsymbol{\pi}} \sim \boldsymbol{Q}(\boldsymbol{\pi})$ and base matrices $\hat{\bar{\Psi}} \sim \boldsymbol{Q}(\bar{\Psi})$
  Sample agent confusion matrices $\hat{\phi}_a \sim \boldsymbol{Q}(\phi_a)$ and $\hat{\psi}_m \sim \boldsymbol{Q}(\psi_m)$
  `// Process Mini-Batch (Amortized Step)`
  Sample batch $\mathcal{B} \subset \{1, \ldots, N\}$
  $\mathcal{L}_{\text{batch}} \leftarrow 0$
  for *each item* $i \in \mathcal{B}$ **do**
    Extract features $\mathbf{x}_i \leftarrow \text{Flatten}(\text{Avg}(\mathbf{s}_i))$
    Predict latent mean: $\hat{\boldsymbol{\mu}}_{q,i} \leftarrow \text{Encoder}_\nu(\mathbf{x}_i)$
    Sample prevalence (ALT): $\mathbf{q}_i \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{q,i}, \hat{\Sigma}_\theta)$
    Compute simplex: $\theta_{i,t} \leftarrow \text{Softmax}(\mathbf{q}_{i,t})$
    `// Marginalize ground truth` y
    $\log p(\text{obs}_i) \leftarrow \sum_t \text{LogSumExp}_k \Big[\log p(y_{i,t} = k \mid \theta_{i,t}) + \log p(\text{obs}_{i,t} \mid y_{i,t} = k, \hat{\phi}, \hat{\psi}) \Big]$
    $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \log p(\text{obs}_i)$
  `// Marginalize archetype assignments` z_m
  $\mathcal{L}_{\text{model}} \leftarrow \sum_m \log \left(\sum_{f=1}^F \hat{\pi}_f \cdot p(\hat{\psi}_m \mid \hat{\bar{\psi}}_f) \right)$
  $\mathcal{L}_{\text{KL}} \leftarrow \text{KL}\left(\boldsymbol{Q}(\text{Global}) \parallel p(\text{Global})\right) + \frac{N}{B} \sum_{i \in \mathcal{B}} \text{KL}\left(\boldsymbol{Q}(\mathbf{q}_i) \parallel p(\mathbf{q}_i)\right)$
  $\mathcal{L} \leftarrow \frac{N}{B}\mathcal{L}_{\text{batch}} + \mathcal{L}_{\text{model}} - \mathcal{L}_{\text{KL}}$
  Update parameters $\xi, \nu \leftarrow \text{Adam}(\nabla \mathcal{L})$

9.2 Proof of Theorem 1

Proof. We proceed in three steps.

Step 1 (Decomposition). By linearity of covariance,

$$\boldsymbol{V}_{\text{true}} = \sum_{m=1}^M w_m^2 \boldsymbol{V}_m + \sum_{m \neq n} w_m w_n \boldsymbol{V}_{mn} = \boldsymbol{V}_{\text{naive}} + \sum_{m \neq n} w_m w_n \boldsymbol{V}_{mn}. \quad (19)$$

Thus,

$$\boldsymbol{\Delta}_M = \sum_{m \neq n} w_m w_n \boldsymbol{V}_{mn} = 2 \sum_{m<n} w_m w_n \cdot \text{sym}(\boldsymbol{V}_{mn}), \quad (20)$$

where $\text{sym}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\top)$.

Step 2 (Lower bound via correlation assumption). By condition (C1),

$$\boldsymbol{\Delta}_M \succeq 2\tau_{\min} \boldsymbol{V}_{\min} \sum_{m<n} w_m w_n. \quad (21)$$

Using the algebraic identity

$$\sum_{m<n} w_m w_n = \frac{1}{2}\left[\left(\sum_{m=1}^M w_m \right)^2 - \sum_{m=1}^M w_m^2 \right] = \frac{1}{2}\left(1 - \sum_{m=1}^M w_m^2 \right), \quad (22)$$

we obtain

$$\boldsymbol{\Delta}_M \succeq \tau_{\min} \boldsymbol{V}_{\min} \left(1 - \sum_{m=1}^M w_m^2 \right). \quad (23)$$

Step 3 (Asymptotic limit). Under conditions (R3),

$$\sum_{m=1}^M w_m^2 \leq M \cdot \frac{C^2}{M^2} = \frac{C^2}{M} \xrightarrow{M \to \infty} 0. \quad (24)$$

Therefore,

$$\lim_{M \to \infty} \boldsymbol{\Delta}_M \succeq \tau_{\min} \boldsymbol{V}_{\min}(1 - 0) = \tau_{\min} \boldsymbol{V}_{\min} \succ \mathbf{0}. \quad (25)$$

□

9.3 Posterior Inference

Algorithm 1 summarizes the Amortized Stochastic Variational Inference (SVI) procedure detailed in Section 5, implemented using the Pyro probabilistic programming language [5].

9.4 Synthetic Data Generation Process

To evaluate CalCo's ability to recover latent parameters and transfer statistical strength, we developed a simulation engine that generates multi-task annotations with configurable correlation structures. Unlike the inference model which places priors on correlations, the simulator accepts fixed ground-truth covariance structures to benchmark performance against known dependencies.

The generation process is governed by two distinct correlation matrices: Ω_θ, which controls the semantic dependencies between tasks (e.g., "Politics" implies "Toxic"), and Ω_ψ, which controls the error correlation between AI agents (e.g., shared failure modes among model families).

9.4.1 Correlated Ground Truth Generation. We generate item-specific prevalence vectors that exhibit inter-task correlations. We accept a task correlation matrix $\Omega_\theta \in \mathbb{R}^{(K-T)\times(K-T)}$ as input. We construct the task covariance matrix $\Sigma_\theta = \text{diag}(\boldsymbol{\sigma}_\theta)\Omega_\theta\text{diag}(\boldsymbol{\sigma}_\theta)$, where $\boldsymbol{\sigma}_\theta$ determines the variance of the prevalence.

For each item i, we sample a latent prevalence vector $\mathbf{q}_i$ in ALT space:

$$\mathbf{q}_i \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \Sigma_\theta) \quad (26)$$

These vectors are partitioned per task into sub-vectors $\mathbf{q}_{i,t}$ and transformed into valid probability simplices $\theta_{i,t}$ via the softmax function: $\theta_{i,t} = \text{Softmax}([\mathbf{q}_{i,t}, 0])$. Finally, the true label $y_{i,t}$ is sampled from the resulting categorical distribution: $y_{i,t} \sim \text{Categorical}(\theta_{i,t})$.

9.4.2 Correlated AI Agent Performance. To simulate realistic ensembles where AI agents share error profiles (e.g., GPT-4 and GPT-4-Turbo), we generate model confusion matrices using a model correlation matrix $\Omega_\psi \in \mathbb{R}^{M\times M}$. We employ a Gaussian Copula method to sample correlated accuracies:

(1) We sample latent ability vectors $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \Omega_\psi)$.
(2) We map these values to the desired accuracy range $[p_{min}, p_{max}]$ using the standard normal cumulative distribution function Λ:

$$\alpha_m = p_{min} + (p_{max} - p_{min}) \cdot \Lambda(b_m) \quad (27)$$

The confusion matrix $\psi_{m,t}$ for model m is constructed such that the diagonal (correct class probability) equals α_m, while off-diagonal errors are distributed uniformly. Model scores $\mathbf{s}_{i,t,m}$ are then sampled from a Dirichlet distribution centered on the row of the confusion matrix corresponding to the true label: $\mathbf{s}_{i,t,m} \sim \text{Dirichlet}(\lambda \cdot \psi_{m,t,y_{i,t}})$.

For human annotations, we simulate a sparse labeling setting. For each item i, we sample a random subset of annotators $\mathcal{A}_i \subset \{1, \ldots, A\}$ of size J_i. For $j \in \{1, \cdots, J_i\}$, a label $r_{i,t,j}$, provided by human agent $a_{i,j}$ is sampled according to their individual confusion matrix $\phi_{a_{i,j},t}$, which is generated independently for each annotator.

The complete generative procedure is summarized in Algorithm 2.

Algorithm 2: Synthetic Multi-Task Data Generation

Input: Number of items N, Task Correlation Ω_θ, Model Correlation Ω_ψ
Output: Dataset $\mathcal{D} = \{(\mathbf{r}_i, \mathbf{s}_i, \mathbf{y}_i)\}_{i=1}^N$
// 1. Initialize Global Parameters
Construct task covariance Σ_θ from Ω_θ and $\boldsymbol{\sigma}_\theta$
// 2. Generate Correlated Model Accuracies
Sample latent abilities $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \Omega_\psi)$
for *each model* $m \in \{1, \ldots, M\}$ **do**
  Compute accuracy $\alpha_m \leftarrow p_{min} + (p_{max} - p_{min}) \cdot \Lambda(b_m)$
  Construct model confusion matrices $\psi_{m,t}$ using α_m (diagonal dominance)
Generate human confusion matrices $\{\phi_{a,t}\}$ (sampled independently)
// 3. Item-Level Generation
for *each item* $i = 1$ **to** N **do**
  // Sample latent context
  Sample ALT vector $\mathbf{q}_i \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \Sigma_\theta)$
  for *each task* $t = 1$ **to** T **do**
    Extract task sub-vector $\mathbf{q}_{i,t}$ from $\mathbf{q}_i$
    Compute prevalence $\theta_{i,t} = \text{Softmax}([\mathbf{q}_{i,t}, 0])$
    Sample true label $y_{i,t} \sim \text{Categorical}(\theta_{i,t})$
    // Generate Sparse Human Labels
    Sample subset of reviewers $\mathcal{A}_i$ of size J_i
    for $j = 1$ **to** J_i **do**
      Sample label $r_{i,t,j} \sim \text{Categorical}(\phi_{a_{i,j},t,y_{i,t}})$
    // Generate Model Scores
    for *each model* $m = 1$ **to** M **do**
      Sample scores $\mathbf{s}_{i,t,m} \sim \text{Dirichlet}(\lambda \cdot \psi_{m,t,y_{i,t}})$
return $\mathcal{D}$