From Labels to Decisions: A Mapping-Aware Annotator Model

Evan Yao
MIT

Jagdish Ramakrishnan
Meta

Xu Chen
Meta

Viet-An Nguyen
Meta

Udi Weinsberg
Meta

ABSTRACT

Online platforms regularly rely on human annotators to make real-time operational decisions for tasks such as content moderation. While crowdsourcing models have been proposed for aggregating noisy labels, they do not generalize well when annotators produce a labels in a large space, e.g., generated from complex review trees. We study a novel crowdsourcing setting with D possible operational decisions or outcomes, but annotators produce labels in a larger space of size $L > D$ which are mapped to decisions through a known mapping function. For content moderation, such labels can correspond to violation reasons (e.g. nudity, violence), while the space of decisions is binary: remove the content or keep it up. In this setting, it is more important to make the right decision rather than estimating the correct underlying label. Existing methods typically separate out the labels to decisions mapping from the modeling of annotators, leading to sub-optimal statistical inference efficiency and excessive computation complexity. We propose a novel confusion matrix model for each annotator that leverages this mapping. Our model is parameterized in a hierarchical manner with both population parameters shared across annotators to model shared confusions and individual parameters to admit heterogeneity among annotators. With extensive numerical experiments, we demonstrate that the proposed model substantially improves accuracy over existing methods and scales well for moderate and large L. In a real-world application on content moderation at Meta, the proposed method offers a 13% improvement in AUC over prior methods, including Meta's existing model in production.

CCS CONCEPTS

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

KEYWORDS

Crowdsourcing, confusion matrix, content moderation

ACM Reference Format:
Evan Yao, Jagdish Ramakrishnan, Xu Chen, Viet-An Nguyen, and Udi Weinsberg. 2023. From Labels to Decisions: A Mapping-Aware Annotator Model. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10,*

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599828

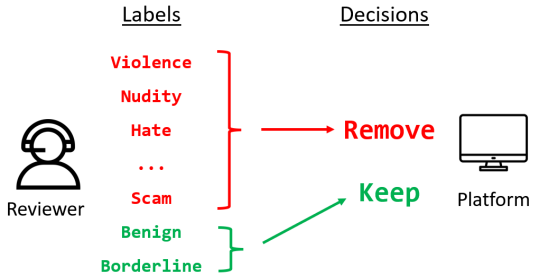

Figure 1: Example of this mapping $u(\cdot)$ for content moderation. The large space of L labels produced by the human reviewer (left) is mapped to one of $D = 2$ operational decisions to be taken by the platform (right)

2023, Long Beach, CA, USA. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599828

1 INTRODUCTION

Despite the widespread use of machine learning for content moderation in online platforms, determining whether a content is policy-violating is often a difficult task requiring human review. At Meta, millions of pieces of content are reviewed by thousands of human annotators across the globe on a daily basis [2]. Such reviewers are trained to follow a protocol, with specific instructions on how to review and classify specific violation types. These instructions are typically codified in a complex decision tree that reviewers use to determine whether a piece of content violates Meta's community standards [19, 20] so it can be promptly removed. Throughout our work, we will use the terms annotators and reviewers interchangeably.

Due to various nuances in violations, e.g., determining a specific type of slur or the context in which it is used, reviewers typically produce labels in a large space indicating the type of policy violation or lack thereof. This detailed label is then mapped to a specific operational decision: remove the content or keep it up (see Figure 1). In this paper, we consider this common setup at Meta, where we have a large label space with few operational decisions. Some examples of such a review paradigm include: 1) a reviewer marks the operational decision, e.g., remove, and provides additional detailed context on reason why the content was violating, e.g., graphic violence. The large space of labels is the space of possible reasons or additional context provided by the reviewer, 2) a reviewer answers a sequence of questions on the content (e.g., does it contain a slur?) and the answers correspond to violating or non-violating by a mapping function the reviewer may be unaware of. Here, the large label space consists of all possible sequences of answers given by the reviewer.

More generally, we consider a setting with L possible labels each of which are mapped into one of $D < L$ operational decisions by a given mapping function $u(\cdot)$. For each item which requires an operational decision, we collect multiple labels from different annotators and must aggregate them into a final decision. While annotators choose labels from among L possible labels, the platform is only interested in making the correct decision and quantifying its uncertainty. To do so, a common approach from the crowdsourcing literature is to model each annotator's labeling behavior with a *confusion matrix* (see e.g., [8, 27, 30]) and each content with some latent state. The most natural approach would be to model each content with a true label among the L possible labels and each reviewer with a $L \times L$ confusion matrix, where each row is a categorical distribution over the label space, describing this annotator's labeling behavior when reviewing content with some true label. Such a model allows us to quantify our uncertainty of each content's true label conditioned on the annotator's labels. We can then quantify our uncertainty over decisions by applying $u(\cdot)$.

While such $L \times L$ confusion matrices work well in capturing any reviewer-specific features across the labels (e.g., an annotator who is especially inaccurate at identifying one particular label or is often biased towards another label), estimating $L \times L$ parameters for *each* annotator is not scalable. A natural approach to reduce the parameter space is to leverage the fact that we are only interested estimating the correct decision. We first apply the mapping function $u(\cdot)$ to reduce all observed labels down to their corresponding decisions. Then, we can apply a similar modeling idea as before except over the D decisions: each content has a true decision from D possible decisions and each reviewer's behavior with a $D \times D$ confusion matrix. While such an approach is scalable (as D is usually small), by collapsing all labels that map to the same decision, valuable information from the label space itself may be lost. For example, consider the two labels `violence` and `nudity` from the content moderation example in Figure 1. Even though both map to the decision `remove`, one may believe that a label of `nudity` results in a *more confident* decision to remove the content compared to violence because violence is inherently a more a subjective violation.

Our work develops and validates a novel approach that achieves the *best of both worlds*, capturing the nuances of the L labels for only a moderate increase in parameter space. We describe our contributions as follows:

(1) **Novel Confusion Matrix Model.** We propose a novel crowdsourcing model that learns $D \times L$ rectangular confusion matrices, where some parameters are shared amongst reviewers, while others are personalized. Our approach incorporates the mapping $u(\cdot)$ in the model to improve accuracy and scalability when L is large. The aforementioned $L \times L$ and $D \times D$ approaches do not leverage the mapping function $u(\cdot)$ as part of the model, but rather apply $u(\cdot)$ either before or after the model inference itself. We are not aware of prior methods that specifically focus on this setting with a large space of labels mapping to a few operational decisions.
(2) **Insights from Simulations.** We show through simulations when our model is most beneficial compared to three benchmark approaches including the aforementioned $L \times L$ and $D \times D$ approaches (denoted `multi` and `binary` respectively). Our model significantly outperforms benchmarks when labels mapping to the same decision are heterogeneous, training data per annotator is limited, and L is large.
(3) **Application to Content Moderation at Meta.** Based on data from community operations at Meta, we show that our proposed approach is effective on a large-scale dataset by utilizing the label-specific information while reducing model complexity and inference time. Our model achieves a 13% improvement in AUC after a single review, and over 6% after two reviews.

Aim and Scope. We describe an approach that is effective for a real-world content moderation application at Meta, improving upon a method previously deployed [24]. In addition, while our method is motivated by this application, it is generally applicable to multiple domains, where detailed reviewer labels are mapped to a smaller space of decisions. For example, other domains that could benefit from our approach include image or text classification where descriptive tags are used, or medical diagnosis where detailed labels on the location of abnormalities in images can inform the presence of a disease.

Reproducibility. Technical details about our model and simulations can be found in Appendix A and B. The code can be found at `https://github.com/facebookresearch/clara/tree/main/mapping-aware-model`

2 RELATED WORK

Aggregating labels provided by non-experts to infer the correct answer has been the focus of much research in the crowdsourcing literature [1, 7, 16, 26, 29, 36, 37]. Specifically, our work is broadly related to the following areas:

Multi-Class Crowdsourced Labeling. For multi-class labeling tasks where each reviewer provides a label by picking one out of L possible options, a common modeling paradigm is to assume that each reviewer is characterized by an $L \times L$ confusion matrix [8, 14, 18, 28]. To accurately estimate the confusion matrices when L is large, many extensions have been proposed. One direction is to share information across confusion matrices by mixing a single common population-wide confusion matrix with per-reviewer's confusion matrices [4, 6, 13]; this is the benchmark model `common` described in Section 4.1. Another direction is to model and capture the correlation among reviewers' confusion matrices [3, 14, 17, 22]. Although this approach does not reduce the number of parameters estimated, it does allow for insights from one reviewer to generalize to another.

To reduce the parameter space, many modeling approaches perform clustering to group "similar" reviewers into communities [12, 21, 27, 30, 33]. In addition, to address the sparsity in the observed labels, another line of work incorporates items'

features to estimate feature-dependent confusion matrices using logistic regression [15, 34, 35] or Gaussian processes [23]. In this work, we leverage the additional information from the label-to-decision mapping to capture the confusion matrices more effectively and accurately.

Hierarchical Crowdsourced Labeling. In many applications, especially in annotating images and text for training machine learning algorithms, reviewers will be asked to produce a hierarchical label such as `animal, dog, golden retriever` in object detection. The main work on aggregating hierarchical labels from different reviewers comes from Otani et. al. [25] Motivated by an existing model in probabilistic label aggregation [32] and item response theory [31], Otani et. al. model a hierarchical classification task as a sequence of independent multi-class tasks, each with their own difficulty level, in addition to a set of reviewers with their own skill level. While not directly related to aggregating labels, other work in hierarchical crowdsourcing has focused on incomplete annotations [10] (ones that don't stretch down to the leaf) and constructing hierarchical structures out of flat multi-class labels [9]. Our setting in this paper is different from 2-tier hierarchical classification in that we are interested in estimating the high-level decisions rather than the low-level ones, which was the goal of previous work.

3 RECTANGULAR ANNOTATOR MODEL

In this section, we describe the basic setup of our problem setting and our annotator model `rectangular`. Throughout our paper, we focus on applications with binary decisions (i.e., $D = 2$ such as remove or keep), but our model is described generally for any $D \geq 2$.

3.1 Problem Setup

Label Space. Consider a crowdsourcing setting with D possible decisions or outcomes (e.g. remove or keep) and L possible labels assigned by human annotators. Let the ordered set of possible decisions and labels be $\mathcal{D}$ and $\mathcal{L}$ respectively, with $|\mathcal{D}| = D$ and $|\mathcal{L}| = L$. For notational simplicity, we will also refer to a label $\ell \in \mathcal{L}$ by its index, i.e. $\ell \in [L]$, where $[n] = \{1, 2, \ldots, n\}$ for any positive integer n. A fixed mapping $u : \mathcal{L} \to \mathcal{D}$ is provided which maps each label into a corresponding decision. We also define $\mathcal{L}_d \subset \mathcal{L}$ to be the set of labels that map to decision $d \in \mathcal{D}$, i.e. $\mathcal{L}_d = \{\ell \in \mathcal{L} \mid u(\ell) = d\}$.

Annotators and Observed Labels. There are I items requiring labels, each of which receive labels from a small subset of A annotators. Items and annotators are indexed by $[I]$ and $[A]$ respectively. Each of the I items is annotated by N_i annotators, each of whom produces a label in $\mathcal{L}$. Let $\{a_{i,j}\}_{i\in[I],j\in N_i}$ and $\{\ell_{i,j}\}_{i\in[I],j\in N_i}$ denote the set of annotators and labels, respectively. Here, $a_{i,j} \in [A]$ is the index of the annotator who made the jth review for content $i \in [I]$, and $\ell_{i,j} \in \mathcal{L}$ is the label they produced.

3.2 Generative Model: `rectangular`

We now describe our data generative model `rectangular` and highlight how it leverages the hierarchical structure induced by $u(\cdot)$. A formal generative process and plate diagram can be found in Figures 9 and 10 in Appendix A.

3.2.1 Basic Generative Process. The generative process consists of 3 main components: true decisions ($y_i \in \mathcal{D}$), confusion matrices ($\psi^{(a)} \in \mathbb{M}^{D\times L}$)[1], and observed labels $\ell_{i,j} \in \mathcal{L}$. Note that N_i and $a_{i,j}$'s are fixed from the observed data and we do not model how they are generated. The first two components are latent (unobserved) variables while the last is observed.

True Decisions. We model each item with a *true decision* $y_i \in \mathcal{D}$ capturing the true or correct decisions if annotators were perfect. First, we draw $\theta \in \Delta^{D-1}$ from a Dirichlet distribution with concentration vector all-ones [2]. Each θ_d represents the proportion of items in the system with true decision $d \in \mathcal{D}$. For each item $i \in [I]$, we draw its true decision y_i independently according to a categorical distribution over $\mathcal{D}$ with probabilities given by $\theta \in \Delta^{D-1}$. We denote this as $y_i \overset{\text{iid}}{\sim} \text{Cat}(\theta)$.

Annotator Confusion Matrices. Each annotator $a \in [A]$ has an associated rectangular confusion matrix $\psi^{(a)} \in \mathbb{M}^{D\times L}$ where each row $\psi_d^{(a)}$ is a L-dimensional probability vector over $\mathcal{L}$. Intuitively, the confusion matrix fully captures the stochasticity in each annotator's labeling behavior. Conditioned on a particular correct decision, the corresponding row of a annotator's confusion matrix gives a categorical distribution over observed labels and the annotator's label is drawn accordingly. The novelty of our model is captured in the following two ways regarding confusion matrices:

(1) **Rectangular Design**. Since we are only interested in making correct decisions (and not estimating the correct label), modeling our latent state with D decisions reduces our model complexity over modeling the latent state with L labels, resulting in $D \times L$ rectangular confusion matrices as opposed to more traditional square matrices of size $D \times D$ or $L \times L$.
(2) **Joint Distribution**. We generate $\{\psi^{(a)}\}_{a\in[A]}$ in a way which allows information to be *shared* across annotators in a way that leverages the hierarchical structure induced by $u(\cdot)$. We describe this process in detail in Section 3.2.2.

Observed Labels. Fixing the values of $\{y_i\}_{i\in[I]}$ and $\{\psi^{(a)}\}_{a\in[A]}$, for each item $i \in [I]$, we draw the observed labels $\{\ell_{i,j}\}_{j\in[N_i]}$ independently as follows:

$$\left(\{\ell_{i,j}\}_{j\in[N_i]} \mid y_i, \psi^{(a)}\right) \sim \text{Cat}\left(\psi_{y_i}^{(a_{i,j})}\right)$$

Conditioned on y_i and $\psi^{(a)}$, the labels for each item are drawn independently according to the confusion matrix given by annotator $a_{i,j} \in [A]$ and the row corresponding to y_i.

Example 1 (Rectangular Confusion Matrix). Figure 2 shows a rectangular confusion matrix $\psi^{(a_{i,j})}$ for annotator $a_{i,j} \in [A]$. The two decisions are $\mathcal{D} =$ `{A, B}` while the four labels are $\mathcal{L} =$ `{A1, A2, B3, B4}` and the mapping between them is

[1] Here $\mathbb{M}^{p\times q} \subset \mathbb{R}^{p\times q}$ is the set of all p by q confusion matrices, which are matrices where every row is a probability distribution.
[2] Δ^{n-1} denotes the n-dimensional simplex, i.e. probability distributions of length n.

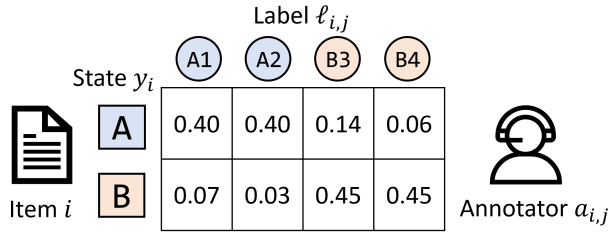

Figure 2: Rectangular confusion matrix $\psi^{(a_{i,j})}$ with $D = 2$ latent states {A, B} in squares and $L = 4$ labels {A1, A2, B3, B4} in circles. Each row of the matrix defines a categorical distribution over {A1, A2, B3, B4}

$u(\texttt{A1}) = u(\texttt{A2}) = \texttt{A}$ and $u(\texttt{B3}) = u(\texttt{B4}) = \texttt{B}$. Given $y_i \in \{\texttt{A}, \texttt{B}\}$, this annotator generates a label based on a categorical distribution given by each row of Figure 2.

3.2.2 Jointly Drawing Confusion Matrices. A naive way to draw $\{\psi^{(a)}\}_{a\in[A]}$ would be to draw each $\psi^{(a)}$ independently from some prior distribution over $\mathbb{M}^{D\times L}$. Such a model formulation leads to $O(ADL)$ parameters, which can be quite large and hence difficult to estimate in the inference process. More importantly though, as described in Section 1, reviewers are trained for the purpose of quality control and possess shared characteristics. Therefore, we assume instead that the set of confusion matrices $\{\psi^{(a)}\}_{a\in[A]}$ is parameterized with a smaller number of parameters, including one set of *individual* parameters and one set of *shared* parameters

$$\left\{\phi^{(a)} \in \mathbb{M}^{D\times D}\right\}_{a\in[A]} \text{ and } \left\{\eta_{d,d',\ell} \in [0,1]\right\}_{d,d'\in\mathcal{D}^2,\ell\in\mathcal{L}_{d'}}.$$

We model each annotator's labeling process step in two steps: choosing a decision $d \in \mathcal{D}$ and then producing a label ℓ from $\mathcal{L}_d$. The confusion matrix $\phi^{(a)}$, specific to each individual annotator, models how each annotator first chooses $d \in \mathcal{D}$. Fixing this decision ud, the actual label ℓ is drawn from $\mathcal{L}_d$ according to a categorical distribution with probability vector $\eta_{d,u(\ell)}$ [3] that is shared across all annotators. With this hierarchical model specification induced by the decision mapping u, the actual labeling process of an annotator can be decomposed into two steps: the annotator produces the decision and then the label within that decision. Mathematically, the probability that annotator a produces label ℓ when the ground truth is d is decomposed into the product of $\phi^{(a)}_{d,u(\ell)}$ for the annotator to produce a decision $u(\ell)$, and, $\eta_{d,u(\ell),\ell}$ for them to produce label ℓ out of $\mathcal{L}_{u(\ell)}$. Specifically, we have:

$$\psi^{(a)}_{d,\ell} = \phi^{(a)}_{d,u(\ell)} \cdot \eta_{d,u(\ell),\ell} \tag{1}$$

Example 2 (Deriving ψ from ϕ and η). $\psi^{(a_{i,j})}$ from Figure 2 can be derived from $\phi^{(a)}$ and η given in Figure 3. The two trees represent the decision-making process when the latent state y_i is A or B. The first level of the tree (black solid lines) has probabilities that come from $\phi^{(a)}$, unique to each annotator, while the second level (red dashed lines) come from the shared η's. The final rectangular confusion matrix is given by the product of the two probabilities along the tree's branches down to the leaves.

3.2.3 Intuition: Label Ambiguity Score. While $\phi^{(a)}$ captures the general skill level of each annotator, the shared parameter η captures the *ambiguities* of each label. We define formally this concept of label ambiguity score below and claim in Proposition 1 that `rectangular` assumes all annotators share the same *ordering* of label ambiguity scores.

Definition 3 (Label Ambiguity Score). For any confusion matrix $\psi \in \mathbb{M}^{D\times L}$, the *ambiguity score* of label $\ell \in \mathcal{L}$ with respect to decision $d \in \mathcal{D} \setminus \{u(\ell)\}$ is defined as:

$$b^{\psi}_{\ell,d} := \frac{\psi_{d,\ell}}{\psi_{u(\ell),\ell}}$$

Intuitively, a label ℓ has a high ambiguity score with respect to decision d if there is a relatively high chance of seeing label ℓ when latent state (i.e. y_i) is $d \neq u(\ell)$ versus $u(\ell)$. Such a label ℓ is unreliable for concluding that $y_i = u(\ell)$ as it is likely to be generated even when $y_i = d \neq u(\ell)$.

Example 4 (Ambiguity Score). Consider again the example from Figure 2. The ambiguity score for A1 and A2 are $0.07/0.40 = 0.175$ and $0.03/0.40 = 0.075$ respectively. This implies that label A1 is more ambiguous than label A2 as both labels have a 40% chance of being drawn from their correct decision of A, but label A2 is likely to be drawn from state B.

Our hierarchical model is constructed so that across our confusion matrices $\psi^{(a)}$ for $a \in [A]$, there are commonalities between label ambiguity scores.

PROPOSITION 1. *Consider two annotator $a, a' \in [A]$ with their confusion matrices $\psi^{(a)}$ and $\psi^{(a')}$ drawn according to equation* (1)*. For any two labels $\ell, \ell \in \mathcal{L}$ with $u(\ell) = u(\ell')$ and decision $d \in \mathcal{D}$ with $d \neq u(\ell)$, we have the following:*

$$b^{\psi^{(a)}}_{\ell,d} \leq b^{\psi^{(a)}}_{\ell',d} \quad \Leftrightarrow \quad b^{\psi^{(a')}}_{\ell,d} \leq b^{\psi^{(a')}}_{\ell',d} \tag{2}$$

In other words, our generative process ensures that the *ordering* of ambiguity scores across different labels $\ell \in \mathcal{L}$ for a given decision $d \in \mathcal{D} \setminus \{u(\ell)\}$ is fixed across reviews. Consider Example 4: label A1 has an ambiguity score that is $0.175/0.075 = 2.33$ times higher than that of label A2. By Proposition 2, we conclude that label A1 is more ambiguous than label A2 for all annotators. The proof of Proposition 1 can be found in Appendix A.1.

3.2.4 Complexity. Our model consists of $D \times D$ confusion matrices for each annotator $\psi^{(a)}$ as well as $O(D \cdot L)$ parameters from η, which is non-zero only for $d, d' \in \mathcal{D}^2$ and $\ell \in \mathcal{L}_{d'}$. Therefore, the total number of parameters for our model is $O(AD^2 + DL)$.

3.2.5 Limitations. While our model is simple and intuitive, there are indeed limitations. First, we acknowledge that by reducing the latent space from L dimensions to D, this may

[3] We define $\eta_{d,d',\ell} = 0$ if $\ell \notin \mathcal{L}_{d'}$.

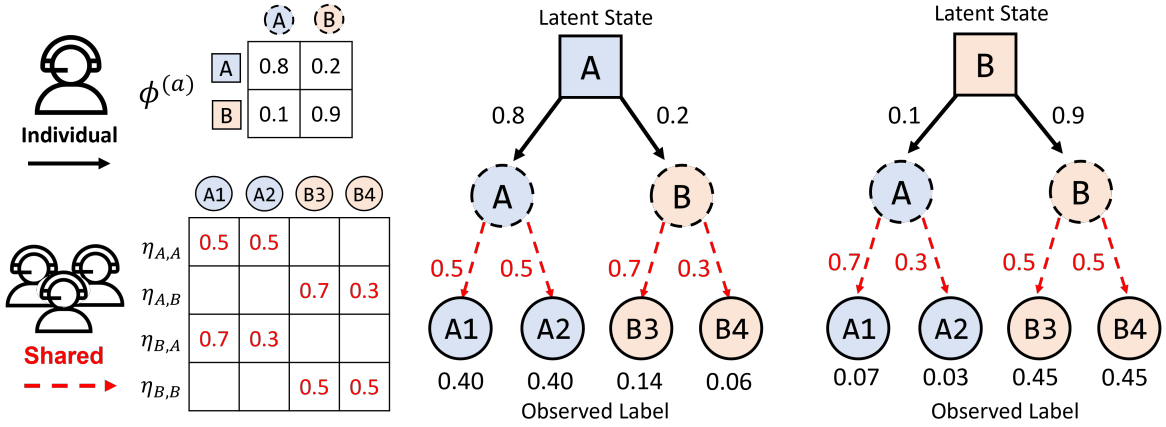

Figure 3: Decomposition of rectangular confusion matrix from Figure 2 into $\phi^{(a)}$ (individual) and η (shared).

not accurately capture the underlying data generation process. Second, we also acknowledge Proposition 1 may not hold in practice as some annotators may find certain labels relatively ambiguous while other annotators find those labels quite unambiguous.

4 BENCHMARKS AND EVALUATION

In this section, we define the benchmark modeling approaches that `rectangular` will be evaluated against.

4.1 Benchmark Approaches

We compare our generative model `rectangular` against three benchmark approaches: `binary`, `multi` and `common`. Similar to `rectangular`, each benchmark approach follows the general structure from Dawid & Skene [8] in modeling each item $i \in [I]$ with some latent state y_i and each annotator $a \in [A]$'s behavior using confusion matrices $\psi^{(a)}$. Given y_i and $\psi^{(a)}$, reviews are independently generated from a categorical distribution given by the y_ith row of $\psi^{(a)}$.

- **`multi`.** We model each review as a L-way multi-class labeling problem by modeling the latent state $y_i \in [L]$ (drawn independently for each item $i \in [I]$) and $\psi^{(a)} \in \mathbb{M}^{L\times L}$ (drawn independently for each annotator $a \in [A]$). Note that `multi` ignores the mapping function $u(\cdot)$ in the model itself. Rather, the function $u(\cdot)$ is used when we apply the model to estimate item i's uncertainty with respect to $\mathcal{D}$, i.e. $P(u(y_i) = d)$, which can be computed as $\Sigma_{\ell\in\mathcal{L}_d} P(y_i = \ell)$. While `multi` is able to capture any type of confusions among the L labels, estimating a $L \times L$ confusion matrix for each annotator can be challenging when L is large and data per annotator is limited. The following two benchmark approaches are two ways of mitigating this issue: `binary` reduces the latent space from L to D, while `common` learns common mistakes shared between $\psi^{(a)}$'s.
- **`binary`.** First, transform $\{x_{i,j} \in \mathcal{L}\}_{i\in[I],j\in[N_i]}$ by applying $u(\cdot)$, thus making $x_{i,j} \in \mathcal{D}$. Model each latent state y_i by one of D possible values and $\psi^{(a)}$ as a $D \times D$ confusion matrix, which are drawn independently from a fixed prior for each annotator. This model ignores the original label space $\mathcal{L}$ and models each annotator's behavior as directly producing decisions in $\mathcal{D}$. While $D \times D$ confusion matrices are often practical to estimate for each annotator, such an approach loses valuable information from $\mathcal{L}$ themselves.
- **`common`.** Presented in [6, 13], `common` is an extension of `multi` where we add in a shared population confusion matrix $\psi^c \in \mathbb{M}^{L\times L}$ that is used to model common confusions shared among all annotators. For example, ψ^c could capture the fact that *all* annotators are often confused between "spam" and "scam" labels in content moderation. Each reviewer's final confusion matrix $\psi^{(a)} \in \mathbb{M}^{L\times L}$ is computed as

$$\psi^{(a)} = \omega_a \cdot \tilde{\psi}^{(a)} + (1 - \omega_a) \cdot \psi^{(c)}$$

where $\tilde{\psi}$ captures each reviewer's own characteristics and the mixing factor $\omega_a \in [0, 1]$ is a learned parameter. The advantage of this approach is that common confusions among annotators will not need to be learned separately for each annotator, reducing the amount of data needed. The disadvantage is that the number of model parameters is still large overall and the addition of ψ^c makes optimization more challenging as ψ^c jointly affects all annotator's behaviors.

Our model `rectangular` uses $D \times D$ confusion matrices for each annotator, similar to `binary`, and shares the parameter η across all annotators similar to `common`. However, unlike those two approaches, our model is able to both capture the information from $\mathcal{L}$ and remain practical to estimate. Figure 4 summarizes the dimensionality of each of the 4 modeling approaches as well as any shared parameters, while 1 summarizes the advantages and disadvantages.

4.2 Evaluation Metrics

4.2.1 Notation. To evaluate the quality of each model $\mathcal{M}$, we partition a dataset of I items into a training set and testing set, with $I_{\text{train}} \sqcup I_{\text{test}} = [I]$. We fit each of our four generative models to the observed training data $\{\ell_{i,j}, a_{i,j}\}_{i\in[I_{\text{train}}],j\in[N_i]}$ using standard posterior inference techniques and obtain a maximum a posteriori (MAP) estimate of the latent variables, in particular $\hat{\theta}, \hat{y}_i$ and $\hat{\psi}^{(a)}$. In all our experiments, posterior

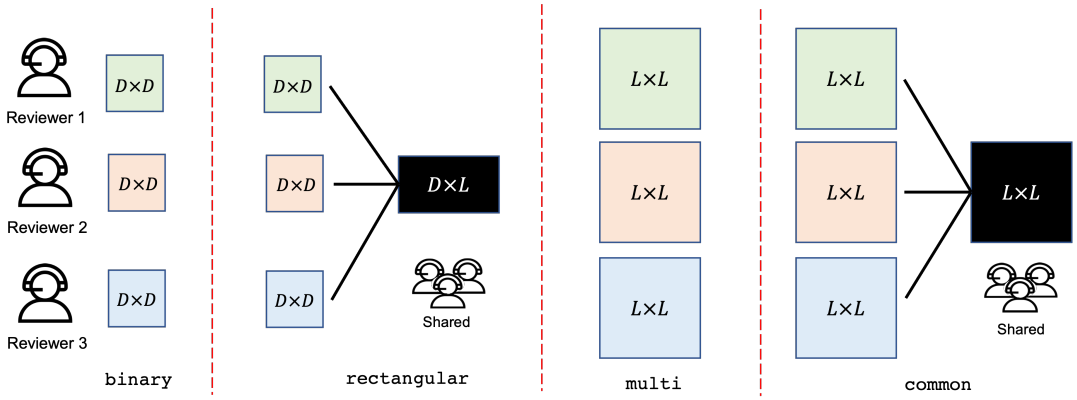

Figure 4: Dimensionality of each reviewer's confusion matrix and shared parameters.

Model	y_i Size	$\psi^{(a)}$ Dim	Parameters	Advantages	Disadvantages
`binary`	D	$D \times D$	$O(AD^2)$	Small number of parameters	Loses information from $\mathcal{L}$
`rectangular`	D	$D \times L$	$O(AD^2 + DL)$	Small number of parameters and captures information in $\mathcal{L}$	Assumes label ambiguity scores are consistent across annotators
`multi`	L	$L \times L$	$O(AL^2)$	Simple and utilizes $\mathcal{L}$	Large number of parameters
`common`	L	$L \times L$	$O(AL^2)$	Utilizes $\mathcal{L}$ while reducing amount of data needed	Large number of parameters and difficult to optimize

Table 1: Comparing the Four Models, Ordered by Complexity

inference was performed using Stan [5] with the L-BFGS optimization algorithm.

For each item $i \in I_{\text{test}}, n \in [N_i]$ and decision $d \in \mathcal{D}$, we use our MAP estimate to compute the following posterior probability:

$$p_{i,n}^d := \begin{cases} P\left(y_i = d \mid \{a_{i,j}, \ell_{i,j}\}_{j\in[n]}\right) & \hat{\phi}^{(a)} \in \mathbb{M}^{D\times L} \\ \Sigma_{\ell\in\mathcal{L}_d} P\left(y_i = \ell \mid \{a_{i,j}, \ell_{i,j}\}_{j\in[n]}\right) & \hat{\phi}^{(a)} \in \mathbb{M}^{L\times L} \end{cases} \tag{3}$$

where $p_{i,n}^d$ is defined as the posterior probability of decision d being the true decision for item i after observing just the first n out of the N_i total reviews. Depending on whether $\mathcal{M}$ models y_i with D or K dimensions, either obtain $p_{i,j}^d$ immediately from y_i or a sum over all labels in $\mathcal{L}_d$. We consider the N_i reviews as happening in sequence, with $p_{i,n}^d$ capturing our posterior after partial information. For notational convenience, let z_i for $i \in I_{\text{test}}$ be the decision with the highest posterior after all N_i reviews, i.e. $z_i := \text{argmax}_{d\in\mathcal{D}} p_{i,N_i}^d$. We can think of z_i as the *ultimate* decision after observing all N_i reviews. Figure 5 below provides a visualization of both $p_{i,n}^d$ and z_i evolving from two annotations.

4.2.2 Metric Definitions. For any item $i \in I_{\text{test}}$, when reviews are plentiful (i.e. N_i is large) and annotators are generally accurate, then any model's estimate of z_i will be close to the true correct decision. However, a strong model can obtain a *early signal* of z_i from just the first few reviews. Namely, we evaluate a model based on how well it can perform the following:

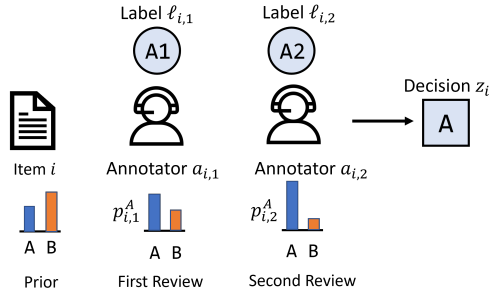

Figure 5: Sequential labeling process with 2 annotators

- After just the first review, distinguish between items for which the final decision is most likely to differ from that of the first review.
- After the second review, given that the first two reviews conflict, prioritize more reviews for items where the final decision is more likely to be a decision of interest (e.g. violating, if we want to quickly remove violating content from the platform).

We define these two metrics formally.

Definition 5. **First Review AUC** for decision $d \in \mathcal{D}$ measures how well a model is able to determine whether $z_i = d$ given that $u(x_{i,1}) = d$. It is calculated by:

$$\texttt{roc\_auc}\left(\{p_{i,1}^d, \mathbb{1}(z_i = d)\}_{i\in[I_{\text{test}}]:u(x_{i,1})=d}\right)$$

where $\texttt{roc\_auc}(\{x_i, y_i\})$ is the area under the receiver operating curve (ROC) with scores x_i and true binary labels y_i.

This metric measure how well our algorithm can decide whether to *trust* the first label of $\ell_{i,1}$. Naively, we could just always trust the first reviewer's decision is correct even though

sometimes further review would have overturned that decision. A strong model would assign a smaller $p^d_{i,1}$ value to items i where it is the case that $z_i \neq u(\ell_{i,1})$ (i.e. the first review is overturned by future reviews). We note that an algorithm which always trusts the first review will receive a first-review AUC score of 0.50.

We can define a similar metric after two reviews, but in the more challenging case where the two reviews conflict.

Definition 6. **Two Conflicting Reviews AUC** measures how well a model is able to estimate the final decision after 2 conflicting reviews. For a decision $d \in \mathcal{D}$, this metric is defined as:

$$\texttt{roc\_auc}\left(\{p^d_{i,2}, \mathbb{1}\{z_i = d\}\}_{i \in [I]\,:\, u(x_{i,1}) \neq u(x_{i,2})}\right)$$

This AUC metric is perfect if $p^d_{i,2}$ is close to 1 when $z_i = d$ and close to 0 when $z_i \neq d$. This would imply that after the second review, $p^d_{i,2}$ is good at differentiating whether $z_i = d$ or not. For example, for the decision to remove violating content, we could allocate our limited labeling resources to items with a high score, thereby removing violating content more quickly.

5 SIMULATION STUDY

We design a simulation study that highlights the advantages of `rectangular` over the three benchmark approaches.

5.1 Data Generative Process

Basics. Our simulation study consists of $L \in \{4, 10, 20\}$ label classes and $D = 2$ decisions, where labels 1 through $L/2$ map to decision `A` and labels $L/2 + 1$ through L map to decision `B`. There are a total of $I \in \{1000, 5000, 10000, 15000, 20000\}$ items and a fixed $A = 50$ annotators. Each annotator $a \in [A]$ has a confusion matrix $\psi^{(a)} \in \mathbb{M}^{L \times L}$, which we describe in detail below. Each item $i \in [I]$ has a ground truth label $y_i \in \mathcal{L}$, which is generated from a uniform distribution over L classes, and receives 3 reviews from annotators chosen uniformly at random from $[A]$. After choosing annotators $a_{i,1}, a_{i,2}, a_{i,3}$ and ground truth y_i, we draw the labels $\ell_{i,1}, \ell_{i,2}, \ell_{i,3}$ independently with $\ell_{i,j}$ being drawn from the y_ith row of $\psi^{(a_{i,j})}$.

Confusion Matrices. We generate confusion matrices according to `common`. Recall that `common`'s generative process involves generating individual confusion matrices $\tilde{\psi}^{(a)}$ for each annotator $a \in [A]$, as well as a common confusion matrix ψ^c that is shared among all annotators. Annotator a produces their labels according to the matrix $\psi^{(a)} = \omega_a \cdot \tilde{\psi}^{(a)} + (1 - \omega_a) \cdot \psi^c$. In our simulation study, we fix $\omega_a = 0.5$ and generate $\tilde{\psi}^{(a)}$ and ψ^c as follow:

- $\tilde{\psi}^{(a)}$ is chosen such that annotators $1, 2, \ldots, A$ are ordered in ascending order of accuracy. For each label $\ell \in \mathcal{L}$, annotator $a \in [A]$ gets it correctly with probability $0.7 + \frac{a}{A} \cdot 0.3$ on average. Otherwise, they make mistakes uniformly at random across the other $L - 1$ labels. The left side of Figure 6 shows an example with $A = 3$ where annotators 1, 2 and 3 are in increasing order of skill.
- ψ^c is controlled by a *label heterogeneity factor* $h \in [0, 1]$, which intuitively captures how much labels vary in terms of label ambiguity score (Definition 3). Consider the two shared matrices on the right half of Figure 6. The bottom matrix is an example for $h = 1.0$ in which there is high heterogeneity in label ambiguity: labels `A1` and `B3` are ambiguous while labels `A2` and `B4` are unambiguous (this is similar to Example 4). On the other hand, the top matrix in the right half of Figure 6 represents the case where $h = 0.0$, meaning that all labels mapping to the same decision behave the same and thus there is no reason to distinguish between such labels.

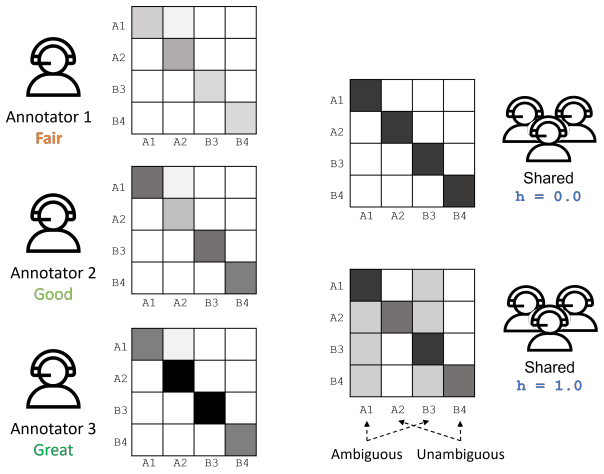

Figure 6: Simulation setup example with $L = 4$. $\tilde{\psi}^{(a)}$ is shown on the left, while possible values of ψ^c is on the right. In the matrices shown, darker square represent values closer to 1

5.2 Results

Figure 7 provides the results of our simulation study. Graphs in the first row plot the first review metric from Definition 5 while the second row corresponds to second review conflicting metric from Definition 6[4]. The x-axis of all graphs is the number of items in the training set, i.e. I, while all metrics were calculated on an independently drawn testing set also of size I. Each column corresponds to a different combination of $h \in \{0.0, 1.0\}$ and $L \in \{4, 10, 20\}$ indicated at the top of each column. Each point is the average of 10 trials with means shown by the lines and 95% confidence intervals shown by the highlighted area around each line.

Figure 7 (left) Varying h. When $h = 0.0$, `binary` performs the best because reducing the labels to decisions as a preprocessing step loses no information (recall when $h = 0.0$ the labels that map to a particular decision are indistinguishable). Our method `rectangular` performs only marginally worse than `binary`, but noticeably better than `common` and `multi` after both the first and second reviews. `common` and `multi` perform poorly because they attempt to estimate $L \times L = 16$ parameters for each annotator when a simple $D \times D = 4$ matrix would have been sufficient. When $h = 1.0$, all methods besides

[4]Note that since our two decisions are symmetric, we show the average of the first-review AUC metrics with respect to either decision.

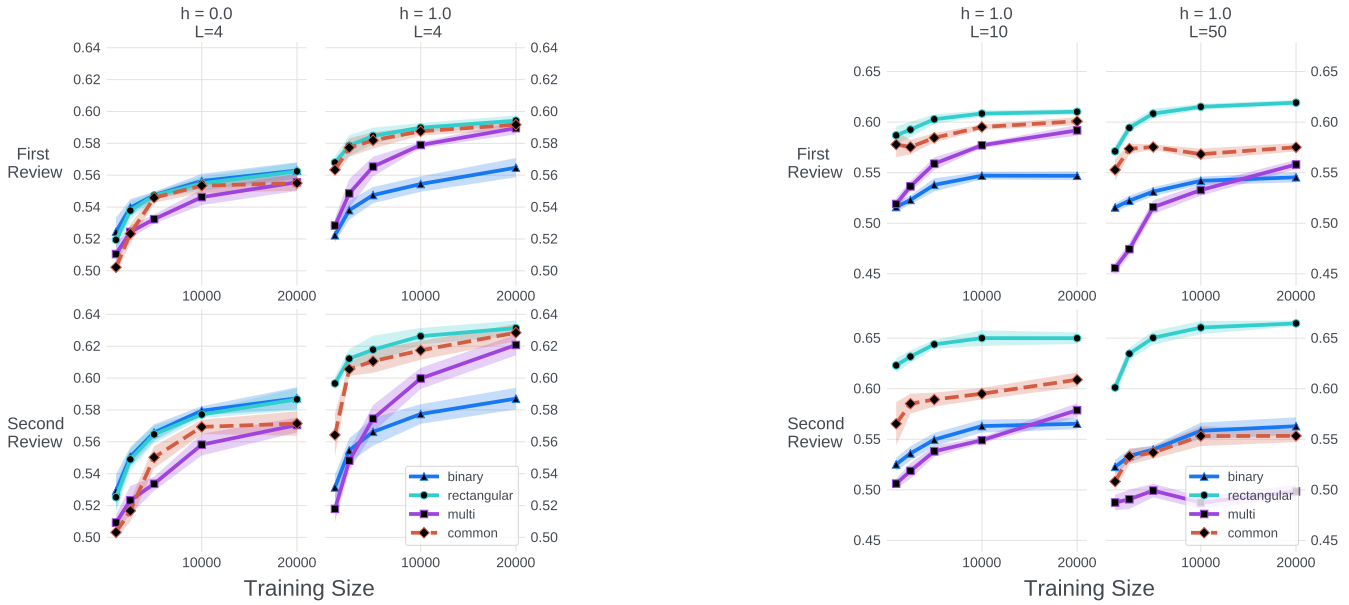

Figure 7: Simulation results when varying the heterogeneity factor h (left) and label space size L (right)

`binary` perform better by taking advantage of the heterogeneity in L labels. We see that `rectangular` is able to achieve the better AUC's compared `common` with fewer training examples, despite the underlying data being generated from `common`.

When labels are indistinguishable (and thus not useful), `rectangular` performs only slightly worse than `binary`, but when labels are heterogeneous, `rectangular` significantly outperforms `multi` and `common` for the same amount of training data. *In this sense,* `rectangular` *is able to achieve the best of both worlds: it performs well regardless of whether the labels are useful for decision-making.*

Figure 7 (right) Varying L. : From the second column of Figure 7 (a), we saw that when $L = 4$ and $h = 1.0$, with enough training examples, all methods that utilize the L labels (i.e. all except `binary`) achieve the same performance. In Figure 7 (b), we hold $h = 1.0$ constant and increase L to 10 and 50. As expected, `binary`'s performance stays the same regardless of L since the larger label space does not affect it. However, as L increases, there a substantial gap appears between `rectangular` and `common`/`multi` in both the first review and second review metrics. For $L = 50$, `common` and `multi` must estimate 50×50 entry confusion matrices for each of 50 annotators, leading to at least 125,000 parameters from just the confusion matrices alone. Running a posterior inference algorithm with so many parameters to estimate is already challenging task, let alone dealing doing so with such limited data. On the other hand, `rectangular` and `binary` must only estimate $50 \times 4 = 200$ annotator-specific parameters, making them much more scalable, but only `rectangular` can also take advantage of the label-specific information. *The low parameter complexity of* `rectangular` *makes it more scalable to large values of L while still taking advantage of the label space itself.*

6 META COMMUNITY OPERATIONS

We demonstrate the value of `rectangular` on real-world data from community operations at Meta. On this large-scale dataset, `rectangular` achieves a 13% improvement in first review AUC and a 6% improvement in second review AUC.

6.1 Dataset Description

Our dataset consists of a sample of 300,000 pieces of content at Meta over a 60 day window, each of which has an average of 2.3 labels from reviewers contracted by Meta. Among these 300,000 pieces of content, the set of reviewers has size $A = 10771$ for an average of around 72 reviews per reviewer Each reviewer provides one of $L = 14$ labels, 13 of which correspond to various violation types such as violence or nudity [5], while the last indicates a lack of violating content. The $D = 2$ decisions are whether the content is violating and needs to be removed from the platform or non-violating and can be kept.

6.2 Results

Results from the Meta content moderation dataset are shown in Figure 8. On the x-axis, we explore how the performance of our algorithms vary with the number of training examples by splitting the 300,000 items into a training set of proportion $x \in \{0.05, 0.1, 0.25, 0.5\}$ and the remaining $1 - x$ items into the testing set. All experiments are averaged over 10 random train-test splits. Means and 95% confidence intervals are shown by the line and shaded regions respectively. The first three panels show the first review and second review AUC metrics, while the last panel shows the training time on a log scale.

Our method `rectangular` significantly outperforms the other three methods in terms of the first review violating and second review conflicting AUC metrics, while achieving nearly

[5]Even though a piece of content could potentially have multiple violation types, we only ask reviewers for the most prominent violation.

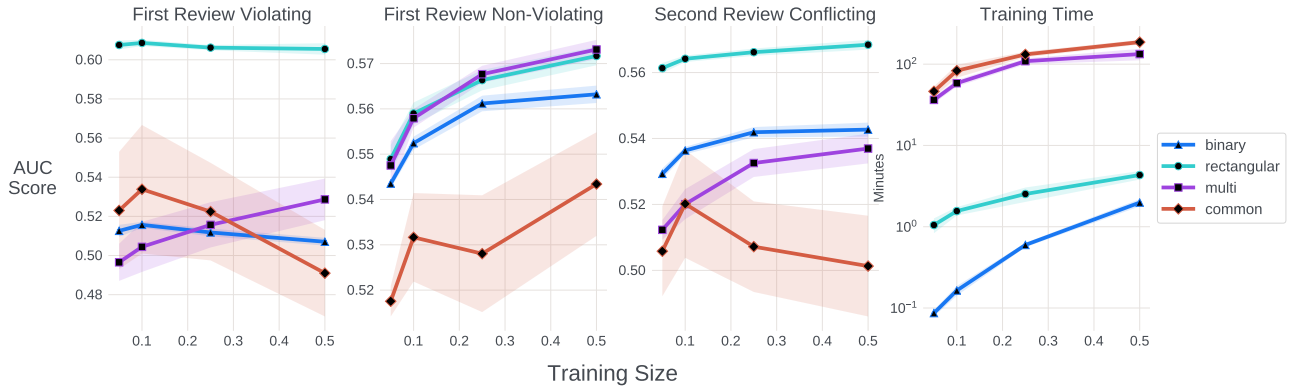

Figure 8: AUC and training time comparison on Meta community operations dataset

the best performance in first review non-violating. In particular, we highlight that the current model used in production at Meta, described in [24] is exactly the benchmark model `binary`. At a 50% training size, `rectangular` outperforms the next best method by around 13% in first reviewer violating AUC, and by 6% for the second review conflicting AUC. Furthermore, `rectangular` is able to achieve nearly its peak AUC score even with a training size of just 5%, while `multi` and `binary` require more data to increase in performance. On the flip side, the method `common` has very high variation across trials and achieves poor mean performance, even decreasing in AUC with an increase in training size. Even though `common` only has a single additional confusion matrix compared to `multi`, the fact that this confusion matrix simultaneous affects all annotator's behavior increases the difficulty of finding the MAP estimate.

Review Efficiency Implication. An increase in first review or second review AUC allows for better prioritization of reviews. Consider the first-review AUC metric for the violating decision. To better allocate scarce human reviews, we would like to prioritize additional reviews for content whose final decision is *most likely to change to non-violating*. For such content, had we trusted the first review only and not performed multiple reviews, we would have created a *false positive*. In one extreme, we could simply trust the first review and not allocate any additional reviews, resulting in the lowest cost but also worst quality as all false positives go undetected. On the other, we could choose to multi-review all content, which has the highest cost, but avoids 100% of false positives. In between, we could choose to multi-review only α proportion of first-review violating content which is *most likely to be non-violating*. Table 2 shows this cost-accuracy trade-off for our four models for $\alpha \in \{0.25, 0.5, 0.75\}$, as well as the improvement of `rectangular` over the next best approach.

After the second review, for content where the first two reviews agree, there is not much uncertainty about the true decision. However, when the first two reviews conflict, additional reviews are needed. By having a high second review conflicting AUC metric, we can better prioritize additional reviews for content that is most likely to be violating, thus reducing the amount of time such content stays on the platform.

Model Type	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$
`binary`	0.274	0.542	0.798
`rectangular`	**0.374**	**0.640**	**0.838**
`multi`	0.271	0.531	0.753
`common`	0.291	0.536	0.759
Improvement	28%	18%	5.0%

Table 2: Proportion of False Positives Avoided when Multi-Reviewing α Proportion

Training Time. The last panel of Figure 8 shows training times for the four models. The training time was determined by tracking the fitting time for L-BFGS optimization in Stan over 10 runs, parallelized with Meta's FBLearner platform [11] We see that the $O(AL^2)$ methods `multi` and `common` take over 100 minutes to train, while `binary` and `rectangular` take orders to magnitude less. While `rectangular` does take more time to train than `binary`, the increase is reasonable at less than 10 minutes and the gap shrinks as the training size increases. While our experiments were run on moderate training sizes, in practice models are trained on millions of examples and so training efficiency is quite important. Overall, `rectangular` provides a significant increase in performance for a modest increase in training time.

7 CONCLUSION

We studied a novel crowdsourcing setting where reviewers produce labels in a space larger than the set of possible outcomes or decisions. In such a setting, estimating large confusion matrices based on the label space is not feasible, yet we still want to leverage the label space to aid our decision-making. We proposed a model `rectangular` which achieves the best of both worlds. Our model works in a 2-tier hierarchical fashion, where the top-level decision is personalized to each reviewer, while the low-level label is generated according to a common distribution shared among all reviewers. Through both simulation studies and real-world content moderation data at Meta, we demonstrate that our approach does a better job of modeling annotators and dynamically prioritizing multiple reviews.

REFERENCES

[1] Omar Alonso. 2019. The practice of crowdsourcing. *Synthesis lectures on information concepts, retrieval, and services* 11, 1 (2019), 1–149.

[2] Vashist Avadhanula, Omar Abdul Baki, Hamsa Bastani, Osbert Bastani, Caner Gocmen, Daniel Haimovich, Darren Hwang, Dima Karamshuk, Thomas Leeper, Jiayuan Ma, Gregory Macnamara, Jake Mullett, Christopher Palow, Sung Park, Varun S Rajagopal, Kevin Schaeffer, Parikshit Shah, Deeksha Sinha, Nicolas Stier-Moses, and Peng Xu. 2022. Bandits for Online Calibration: An Application to Content Moderation on Social Media Platforms.

[3] Peng Cao, Yilun Xu, Yuqing Kong, and Yizhou Wang. 2019. Max-MIG: an Information Theoretic Approach for Joint Learning from Crowds. In *International Conference on Learning Representations*.

[4] Bob Carpenter. 2008. Multilevel Bayesian models of categorical data annotation. *Unpublished manuscript* 17, 122 (2008), 45–50.

[5] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. Stan: A probabilistic programming language. *Journal of Statistical Software*.

[6] Zhendong Chu, Jing Ma, and Hongning Wang. 2021. Learning from Crowds by Modeling Common Confusions. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 7 (May 2021), 5832–5840.

[7] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 1–40.

[8] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28, 1 (1979), 20–28.

[9] Xiaoni Duan and Keishi Tajima. 2019. Improving Multiclass Classification in Crowdsourcing by Using Hierarchical Schemes. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. 2694–2700.

[10] Masafumi Enomoto, Kunihiro Takeoka, Yuyang Dong, Masafumi Oyamada, and Takeshi Okadome. 2021. Quality Control for Hierarchical Classification with Incomplete Annotations. In *Advances in Knowledge Discovery and Data Mining*, Kamal Karlapalem, Hong Cheng, Naren Ramakrishnan, R. K. Agrawal, P. Krishna Reddy, Jaideep Srivastava, and Tanmoy Chakraborty (Eds.). Springer International Publishing, Cham, 219–230.

[11] Facebook. 2016. *Introducing FBLearner Flow: Facebook's AI backbone*. https://code.fb.com/core-data/introducing-fblearner-flow-facebook-s-ai-backbone/

[12] Xiawei Guo and James T. Kwok. 2016. Aggregating Crowdsourced Ordinal Labels via Bayesian Clustering. In *Machine Learning and Knowledge Discovery in Databases*, Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken (Eds.). Springer International Publishing, Cham, 426–442.

[13] Ece Kamar, Ashish Kapoor, and Eric Horvitz. 2015. Identifying and Accounting for Task-Dependent Bias in Crowdsourcing. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 3, 1 (Sep. 2015), 92–101.

[14] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics (AISTATS)*. 619–627.

[15] Jingzheng Li, Hailong Sun, and Jiyi Li. 2022. Beyond confusion matrix: learning from multiple annotators with awareness of instance features. *Machine Learning* (2022), 1–23.

[16] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A Survey on Truth Discovery. *SIGKDD Explor. Newsl.* 17, 2 (Feb. 2016), 1–16.

[17] Yuan Li, Benjamin Rubinstein, and Trevor Cohn. 2019. Exploiting Worker Correlation for Label Aggregation in Crowdsourcing. In *ICML*. 3886–3895.

[18] Qiang Liu, Jian Peng, and Alexander Ihler. 2012. Variational Inference for Crowdsourcing. In *Advances in Neural Information Processing Systems*. 692–700.

[19] Meta. 2022. *Community Standards Enforcement Report*. https://transparency.fb.com/data/community-standards-enforcement

[20] Meta. 2022. *How Meta enforces its policies*. https://transparency.fb.com/enforcement/

[21] Pablo G. Moreno, Antonio Artés-Rodríguez, Yee Whye Teh, and Fernando Perez-Cruz. 2015. Bayesian Nonparametric Crowdsourcing. *JMLR* (2015), 1607–1627.

[22] An T. Nguyen, Byron C. Wallace, and Matthew Lease. 2016. A Correlated Worker Model for Grouped, Imbalanced and Multitask Data. In *UAI*. 537–546.

[23] 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.

[24] Viet-An Nguyen, Peibei Shi, Jagdish Ramakrishnan, Udi Weinsberg, Henry C Lin, Steve Metz, Neil Chandra, Jane Jing, and Dimitris Kalimeris. 2020. CLARA: confidence of labels and raters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2542–2552.

[25] Naoki Otani, Yukino Baba, and Hisashi Kashima. 2016. Quality control of crowdsourced classification using hierarchical class structures. *Expert Systems with Applications* 58 (2016), 155–163. https://doi.org/10.1016/j.eswa.2016.04.009

[26] Silviu Paun, Ron Artstein, and Massimo Poesio. 2022. Statistical Methods for Annotation Analysis. *Synthesis Lectures on Human Language Technologies* 15, 1 (2022), 1–217.

[27] Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian Models of Annotation. *Transactions of the Association for Computational Linguistics* 6, 0 (2018), 571–585. https://transacl.org/index.php/tacl/article/view/1430

[28] 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.

[29] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (Honolulu, Hawaii). Association for Computational Linguistics, 254–263.

[30] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-Based Bayesian Aggregation Models for Crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web* (Seoul, Korea) *(WWW '14)*. Association for Computing Machinery, New York, NY, USA, 155–164. https://doi.org/10.1145/2566486.2567989

[31] Norman D Verhelst, Cornelis AW Glas, and HH De Vries. 1997. A steps model to analyze partial credit. In *Handbook of modern item response theory*. Springer, 123–138.

[32] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Eds.), Vol. 22. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2009/file/f899139df5e1059396431415e770c6dd-Paper.pdf

[33] Gongqing Wu, Liangzhu Zhou, Jiazhu Xia, Lei Li, Xianyu Bao, and Xindong Wu. 2022. Crowdsourcing Truth Inference Based on Label Confidence Clustering. *ACM Trans. Knowl. Discov. Data* (aug 2022). https://doi.org/10.1145/3556545 Just Accepted.

[34] Yan Yan, Rómer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy. 2010. Modeling annotator expertise: Learning when everybody knows a bit of something. In *Artificial Intelligence and Statistics (AISTATS)*. 932–939.

[35] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. 2014. Learning from Multiple Annotators with Varying Expertise. *Mach. Learn.* 95, 3 (June 2014), 291–327.

[36] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. 2016. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. *JMLR* 17, 1 (Jan. 2016), 3537–3580.

[37] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth Inference in Crowdsourcing: Is the Problem Solved? *VLDB* (2017), 541–552.

True Labels. The generative process for the true labels y_i is given below.

- $\theta \sim \text{Dirichlet}(1_D)$, where 1_D is the D-dimensional all ones vector.
- $\gamma \sim \text{Unif}(0.5, 1)$.
- $y_i \sim \text{Categorical}(\theta)$ over decisions $\{1, 2, \ldots, D\}$.

Generating Confusion Matrices $\psi^{(a)}$. Recall that each $\psi^{(a)}$ is comprised of two components: $\phi^{(a)}$ (individual) and $\eta_{d,d',\ell}$ (shared). We describe the generative process for each of these two components.

$\phi^{(a)}$: For each reviewer $a \in [A]$, we draw $\phi^{(a)}$ by drawing for each $d \in \mathcal{D}$ the values

$$\begin{aligned} r_d &\sim \text{Unif}(\gamma, 1) && d \in \mathcal{D} \\ p_d &\sim \text{Dirichlet}(1_D) && d \in \mathcal{D} \\ \phi_d^{(a)} &= r_d \cdot e_d + (1 - r_d) \cdot p_d && d \in \mathcal{D} \end{aligned}$$

where e_d is a length D vector with 0 in all indices and a 1 in the dth index. Intuitively, $\phi^{(a)}$ is a random confusion matrix with diagonal entries guaranteed to be at least γ.

Generating $\eta_{d,d}$. We define $\eta_{d,d} \in \Delta^{L-1}$ as a vector over the L labels in which we fix $\eta_{d,d,\ell} = 0$ for $\ell \notin \mathcal{L}_d$. The remaining entries, i.e. the subvector $\{\eta_{d,d,\ell}\}_{\ell \in \mathcal{L}_d}$, is drawn according to a Dirichlet distribution with the all ones vector.

Generating $\eta_{d,d'}$ for $d \neq d'$. Having drawn $\eta_{d,d}$ for $d \in \mathcal{D}$, we now draw $\eta_{d,d',\ell}$ when $d \neq d'$ by leveraging an auxiliary variable $\rho_{d,d'}$, which is a L dimensional vector that is zero for indices $\ell \notin \mathcal{L}_{d'}$.

- First, draw $\rho_{d,d'}$ independently in each of its $L_{d'}$ non-zero dimensions.

$$\rho_{d,d',\ell} \sim \text{Unif}\left(1, \frac{\gamma}{1-\gamma}\right) \qquad \ell \in [L_{d'}]$$

- Having already chosen $\eta_{d,d}$ and $\rho_{d,d'}$, we set $\eta_{d,d'}$ for $d \neq d'$ in the following way:

$$\eta_{d,d',\ell} = \frac{\rho_{d,d',\ell} \cdot \eta_{d',d',\ell}}{\Sigma_{\ell' \in [L_{d'}]} \left(\rho_{d,d',\ell'} \cdot \eta_{d',d',\ell'}\right)} \qquad \ell \in [L_{d'}]$$

For each $d \neq d'$, $\eta_{d,d'}$ is the normalized version of the element-wise product between $\rho_{d,d'}$ and $\eta_{d,d}$, and hence η_d is a valid probability distribution. See Proposition 2 for a discussion as to why we must choose $\eta_{d,d'}$ in this fashion.

Rectangular Confusion Matrix. We now construct $\psi_{d,\ell}^{(a)}$ by multiplying $\phi^{(a)}$ and η according to equation 1:

$$\psi_{d,\ell}^{(a)} = \phi_{d,u(\ell)}^{(a)} \cdot \eta_{d,u(\ell),\ell}$$

Generate Labels. Finally, for each item $i \in [I]$ and review $j \in [N_i]$ with true state $y_i \in [D]$ and reviewer $a_{i,j}$, we draw the label $\ell_{i,j}$ *independently* for each $j \in [N_i]$ using $\psi^{(a_{i,j})}$:

$$\ell_{i,j} \mid y_i, \psi^{(a_{i,j})} \sim \text{Cat}\left(\psi_{d,\ell}^{(a_{i,j})}\right)$$

Figure 9: Generative process for `rectangular`

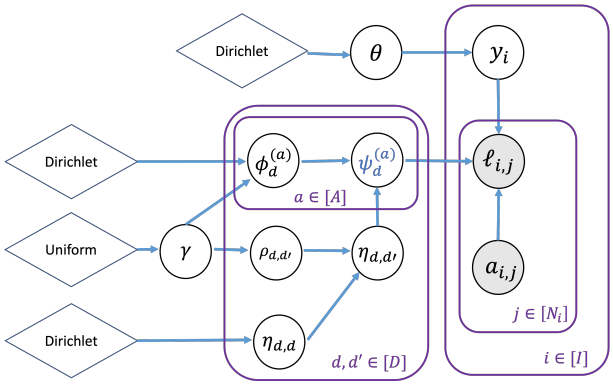

Figure 10: Plate diagram for `rectangular`

Notation	Description
$D \in \mathbb{N}$	Size of decision or outcome space
$L \in \mathbb{N}$	Size of label space
$u : [L] \to [D]$	Map from a label to its decision
L_d	The set of labels which map to decision $d \in [D]$.
$I \in \mathbb{N}$	Number of items being reviewed
$N_i \in \mathbb{N}$	Number of reviews for item $i \in [I]$.
$A \in \mathbb{N}$	Number of unique reviewers
$a_{i,j} \in [A]$	jth reviewer for item i
$\ell_{i,j} \in [L]$	Label provided in the jth review of item i (by reviewer $a_{i,j}$)
$y_i \in [D]$	True decision for item $i \in [I]$
θ	Prevalence over the D decisions (Probability vector over D items)
$\gamma \in (0.5, 1]$	Minimum for diagonal entries of each reviewer's personal the $D \times D$ decision confusion matrix
$\psi^{(a)}$	Individual confusion matrix for labeler $a \in [A]$ over the D categories ($D \times D$ matrix where each row is a probability vector).
$\eta_{d,d'}$	When the true decision is d and a reviewer produces decision d', $\eta_{d,d'}$ is the distribution of which label is produced. (Probability distribution of length L with support over $L_{d'}$).
$\rho_{d,d'} \in \left[1, \frac{\gamma}{1-\gamma}\right]$	Auxiliary variable which controls how different $\eta_{d,d'}$ and $\eta_{d',d'}$ are.

Table 3: Notation Overview for `rectangular`

A DETAILED MODEL DESCRIPTION

In this supplementary section, we provide the formal description of our model `rectangular`. Table 3 gives an overview of the notation in our model while Figure 9 provides a detailed generative process and plate diagram. We also provides the proof of Proposition 1 as well as discusses a technicality around model identifiability.

A.1 Proof of Proposition 1

PROOF. By the way that $\psi_{(a)}$ factors in equation (1), we have that:

$$\frac{b^{\psi^{(a)}}_{\ell,d}}{b^{\psi^{(a)}}_{\ell',d}} = \frac{\psi^{(a)}_{d,\ell}}{\psi^{(a)}_{d,\ell'}} \cdot \frac{\psi^{(a)}_{u(\ell'),\ell'}}{\psi^{(a)}_{u(\ell),\ell}}$$
$$= \frac{\phi^{(a)}_{d,u(\ell)} \cdot \eta_{d,u(\ell),\ell} \cdot \phi^{(a)}_{u(\ell'),u(\ell')} \cdot \eta_{u(\ell'),u(\ell'),\ell'}}{\phi^{(a)}_{d,u(\ell')} \cdot \eta_{u(\ell'),u(\ell'),\ell'} \cdot \phi^{(a)}_{u(\ell),u(\ell)} \cdot \eta_{u(\ell),u(\ell),\ell}}$$

which does not depend on a because $u(\ell) = u(\ell')$ and hence all the terms involving $\phi^{(a)}$ cancel out. □

A.2 Model Technicality: Identifiability

We now discuss a technical aspect of our data generation process. In our formal generative process, we drew our variables in a way that respects identifiability. The way we have informally described our model so far, there is nothing stopping the model from estimating that even though $u(\texttt{A1}) = \texttt{A}$, that the label A1 is actually more likely to be seen when the true decision is B. In other words, our model is currently just as valid if we switch A and B. When we draw our variables η in the formal generative process, we must ensure that the model is uniquely identifiable

PROPOSITION 2 (IDENTIFIABILITY). *Our data generating process above guarantees that for any reviewer $a \in [A]$, an observed label $\ell \in [L]$ is more likely to be generated when the true decision is $d = u(\ell)$ than $d' \neq u(\ell)$.*

$$P(\ell_{i,j} = \ell \mid y_i = d) \geq P(\ell_{i,j} = \ell \mid y_i = d') \quad (4)$$
$$\forall (\ell, d, d') : d = u(\ell), d' \neq u(\ell)$$

PROOF. Based on how labels $\ell_{i,j}$ are generated in our model, we have that equation (4) is equivalent to: $\psi^{(a)}_{d,d} \cdot \eta_{d,d,\ell} \geq \psi^{(a)}_{d',d} \cdot \eta_{d',d,\ell}$ Since $\psi^{(a)}_{d,d} \geq \gamma$ and $\psi^{(a)}_{d',d} \leq 1-\gamma$, it is sufficient to show that $\eta_{d',d,\ell} \leq \eta_{d,d,\ell} \cdot \frac{\gamma}{1-\gamma}$. This is true based on the way we drew $\eta_{d',d,\ell}$ since when $d' \neq d$.

$$\eta_{d',d,\ell} = \frac{\rho_{d',d,\ell} \cdot \eta_{d,d,\ell}}{\Sigma_{\ell' \in [L_d]} \left(\rho_{d',d,\ell'} \cdot \eta_{d,d,\ell'}\right)} \leq \rho_{d',d,\ell} \cdot \eta_{d,d,\ell} \leq \left(\frac{\gamma}{1-\gamma}\right) \cdot \eta_{d,d,\ell}$$

Here, we heavily rely on the fact that $\rho_{d',d,\ell}$ is drawn to be between 1 and $\frac{\gamma}{1-\gamma}$. □

B SIMULATION DATA GENERATION

Our simulation study has $A = 50$ (number of reviewers), $L \in \{4, 10, 20\}$ (number of label classes) and $D = 2$ with the first $L/2$ labels mapping to one decision and the remaining $L/2$ mapping to the other. There are $I \in \{1000, 2500, 5000, 10000, 20000\}$ items each with 3 reviews. Let $\gamma = 0.70$ be a lower bound on the diagonal entries of all confusion matrices we present. A heterogeneity factor $h \in [0, 1]$ is used to control how different the ambiguities of the $L/2$ labels are.

Each reviewer $a \in [A]$ has a confusion matrix $\psi^{(a)} \in \mathbb{R}^{L \times L}$, which is the average of the population confusion matrix ψ^c and their own individual confusion matrix $\tilde{\psi}^{(a)}$, i.e. $\psi_a = 0.5 \cdot \tilde{\psi}^{(a)} + 0.5 \cdot \psi^c$. We describe each of these two components:

- **Population Confusion Matrix ψ^c:** The population confusion matrix is made up of two components:

  $$\psi^c = \gamma \cdot I_L + (1-\gamma) \cdot M$$

  where I_L is the $L \times L$ identity matrix, and M is matrix where each row is the same *error* probability distribution p_M. With probability γ regardless of the what the true label is, the observed label is correct. With probability $1-\gamma$, we will randomly draw an entry from the error distribution p_M. The error distribution p_M is calculated using the heterogeneity factor and is calculated as a convex combination of p_S (S for same) and p_D (D for different).

  $$p_S = \frac{1}{L} \cdot [\underbrace{1, 1, \ldots, 1}_{\frac{L}{2}}, \underbrace{1, 1, \ldots, 1}_{\frac{L}{2}}]$$
  $$p_D = \frac{1}{2 \cdot \Sigma_{i=1}^{L/2} i^2} [\underbrace{1^2, 2^2 \ldots, (L/2)^2}_{\frac{L}{2}}, \underbrace{1^2, 2^2 \ldots, (L/2)^2}_{\frac{L}{2}}]$$
  $$p_M = h \cdot p_S + (1-h) \cdot p_D$$

  Recall that the first $L/2$ labels correspond to one decision and the remaining $L/2$ labels correspond to another. When $h = 0$, $p_M = p_S$ meaning that we make mistakes uniformly at random, which implies that all labels are equally ambiguous (no label is more likely to appear as a mistake). On the other hand, when $h = 1$, then the distribution is very lop-sided, with labels 1 and $L/2+1$ being very unlikely to occur from a mistake, whereas labels $L/2-1$ and $L-1$ are very likely to be generated when the underlying label does not map to the correct decision.
- **Individual Confusion Matrix $\tilde{\psi}^{(a)}$:** Reviewer $a \in [A]$ has an individual confusion matrix $\psi^{(a)}$ with the following entries:

  $$\gamma_a = 0.7 + \frac{a}{A} \cdot 0.3 \qquad a \in [A]$$
  $$\tilde{\psi}^{(a)}_{\ell,\ell} \sim \text{Unif}(\gamma_a - 0.1, \min(1, \gamma_a + 0.1)) \qquad \ell \in [L]$$
  $$\tilde{\psi}^{(a)}_{\ell,\ell'} = 1 - \frac{\tilde{\psi}^{(A)}_{\ell,\ell}}{L-1} \qquad \ell \neq \ell'$$

  In other words, $\tilde{\psi}^{(a)}$ is a diagonal matrix with the same value $\gamma + \frac{a}{A} \cdot (1-\gamma)$ on and uniform entries off the diagonal. Each reviewer $a \in [A]$ has different entries on the diagonal with larger reviewer indices receiving higher diagonal entries on average.

Given each reviewer's final confusion matrix $\psi^{(a)} = 0.5 \cdot \tilde{\psi}^{(a)} + 0.5 \cdot \psi^c$, for each of the I items, we draw its true label $y_i \in [L]$ uniformly at random, and then draw 3 reviewers for this item, $a_{i,1}, a_{i,2}, a_{i,3}$, uniformly at random with replacement. The observed reviews are then chosen according to the y_ith row of the confusion matrices of these 3 reviewers:

$$\ell_{i,j} \mid y_i, \psi^{(a_{i,j})} \sim \text{Cat}\left(\psi^{(a_{i,j})}_{y_i}\right) \qquad j \in [3]$$