Training Complex Models with Multi-Task Weak Supervision

Alexander Ratner[†] Braden Hancock[†] Jared Dunnmon[†] Frederic Sala[†]
Shreyash Pandey[†] Christopher Ré[†]

[†]Department of Computer Science, Stanford University
{ajratner, bradenjh, jdunnmon, fredsala, shreyash, chrismre}@stanford.edu

September 6, 2018

Abstract

As machine learning models continue to increase in complexity, collecting large hand-labeled training sets has become one of the biggest roadblocks in practice. Instead, weaker forms of supervision that provide noisier but cheaper labels are often used. However, these weak supervision sources have diverse and unknown accuracies, may output correlated labels, and may label different tasks or apply at different levels of granularity. We propose a framework for integrating and modeling such weak supervision sources by viewing them as labeling different related sub-tasks of a problem, which we refer to as the *multi-task weak supervision* setting. We show that by solving a matrix completion-style problem, we can recover the accuracies of these *multi-task* sources given their dependency structure, but without any labeled data, leading to higher-quality supervision for training an end model. Theoretically, we show that the generalization error of models trained with this approach improves with the number of *unlabeled* data points, and characterize the scaling with respect to the task and dependency structures. On three fine-grained classification problems, we show that our approach leads to average gains of 20.2 points in accuracy over a traditional supervised approach, 6.8 points over a majority vote baseline, and 4.1 points over a previously proposed weak supervision method that models tasks separately.

1 Introduction

One of the greatest roadblocks to using modern machine learning models is collecting hand-labeled training data at the massive scale they require. In real-world settings where domain expertise is needed and modeling goals change frequently, hand-labeling training sets is prohibitively slow, expensive, and static. For these reasons, practitioners are increasingly turning to weak supervision techniques wherein noisier, often programmatically-generated labels are used instead. Common *weak supervision sources* include external knowledge bases [26; 39; 8; 33], heuristic patterns [14; 29], feature annotations [25; 38], and noisy crowd labels [17; 11]. The use of these sources has led to state-of-the-art results in a range of domains [39; 37]. A theme of weak supervision is that using the full diversity of available sources is critical to training high-quality models [29; 39].

The key technical difficulty of weak supervision is determining how to combine the labels of multiple sources which have different, unknown accuracies, may be correlated, and may label at different levels of granularity. In our experience with users in academia and industry, the complexity of real world weak supervision sources makes this integration phase the key time sink and stumbling block. For example, if we are training a model to classify entities in text, we may have one available source of high-quality but coarse-grained labels—e.g. "Person" vs. "Location"—and one source that provides lower-quality but finer-grained labels; moreover, these sources might be correlated due to some shared component or data source [2; 35]. Handling such diversity requires solving a core technical challenge: estimating the unknown accuracies of multi-granular and potentially correlated supervision sources without any labeled data.

We propose `MeTaL`, a framework for modeling and integrating weak supervision sources with different unknown accuracies, correlations, and granularities. In `MeTaL`, we view each source as labeling one of several related sub-tasks of a problem—we refer to this as the *multi-task weak supervision* setting. We then show that given the dependency structure of the sources, we can use their observed agreement and disagreement rates to recover their unknown accuracies by solving a matrix-completion-style problem. Moreover, we exploit the relationship structure between tasks to observe additional cross-task agreements and disagreements, effectively providing extra signal to learn from. In contrast to previous approaches based on sampling from the posterior of a graphical model directly [30; 2], we are able to apply strong matrix concentration bounds [34], and obtain a simple algorithm for learning and modeling the accuracies of these diverse weak supervision sources. Given their accuracies, we

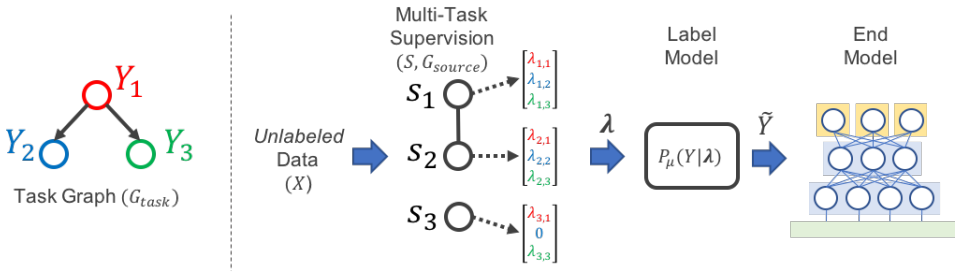

Figure 1: A schematic of the MeTaL pipeline. To generate training data for an *end model*, the user inputs a *task graph* G_{task} defining the relationships between *task labels* $Y_1, ..., Y_t$; a set of *unlabeled* data points X; a set of *multi-task weak supervision sources* $S = \{s_1, ..., s_m\}$ that each output one or more task labels for X; and the dependency structure between these sources, G_{source}. We train a *label model* to learn the accuracies of the sources, outputting a vector of probabilistic training labels $\tilde{\mathbf{Y}}$ for training an end multi-task model.

combine their labels to produce training data which can then be used to supervise arbitrary multi-task learning models [6; 31].

Compared to previous methods which only handled the single-task setting [30; 29], and generally only considered conditionally-independent sources [1; 11], we demonstrate that our multi-task aware approach leads to average gains of 4.1 points in accuracy in our experiments, and has at least three additional benefits. First, many dependency structures between weak supervision sources may lead to non-identifiable models of their accuracies, where a unique solution cannot be recovered. We provide a compiler-like check to establish identifiability—i.e. the existence of a unique set of source accuracies—for arbitrary dependency structures, without resorting to the standard assumption of non-adversarial sources [11], alerting users to this potential stumbling block that we have observed in practice. Next, we provide sharper sample complexity bounds that characterize the benefit of adding additional unlabeled data, and the scaling with respect to the user-specified task and dependency structure. While previous approaches required thousands of sources to give non-vacuous bounds, we capture regimes with small numbers of sources, better mirroring the real-world uses of weak supervision we have observed. Finally, we are able to solve our proposed problem directly with SGD, leading to over $100\times$ faster runtimes compared to prior Gibbs-sampling based approaches [30; 28], and enabling simple implementation using libraries like PyTorch.

We validate our framework on three fine-grained classification tasks in named entity recognition, relation extraction, and medical document classification, for which we have diverse weak supervision sources at multiple levels of granularity. We show that by modeling them as labeling hierarchically-related sub-tasks and utilizing unlabeled data, we can get an average improvement of 20.2 points in accuracy over a traditional supervised approach, 6.8 points over a basic majority voting weak supervision baseline, and 4.1 points over data programming [30], an existing weak supervision approach in the literature that is not multi-task-aware. We also extend our framework to handle unipolar sources that only label one class, a critical aspect of weak supervision in practice that leads to an average 2.8 point contribution to our gains over majority vote. From a practical standpoint, we argue that our framework represents an efficient way for practitioners to supervise modern machine learning models for complex tasks by opportunistically using the diverse weak supervision sources available to them.

2 Related Work

Our work builds on and extends various settings studied in machine learning:

Weak Supervision: We draw motivation from recent work which models and integrates weak supervision using generative models [30; 29; 2] and other methods [13; 19]. These approaches, however, do not handle multi-granularity or multi-task weak supervision, require expensive sampling-based techniques that may lead to non-identifiable solutions, and leave room for sharper theoretical characterization of weak supervision scaling properties. More generally, our work is motivated by a wide range of specific weak supervision techniques, which includes traditional distant supervision approaches [26; 8; 39; 15; 33], co-training methods [4], pattern-based supervision [14; 39], and feature-annotation techniques [25; 38; 23].

Crowdsourcing: Our approach also has connections to the crowdsourcing literature [17; 11], and in particular to spectral and method of moments-based approaches [40; 9; 12; 1]. In contrast, the goal of our work is to support and explore settings not covered by crowdsourcing work, such as sources with correlated outputs, the proposed multi-task supervision setting, and regimes wherein a small number of labelers (weak supervision sources) each

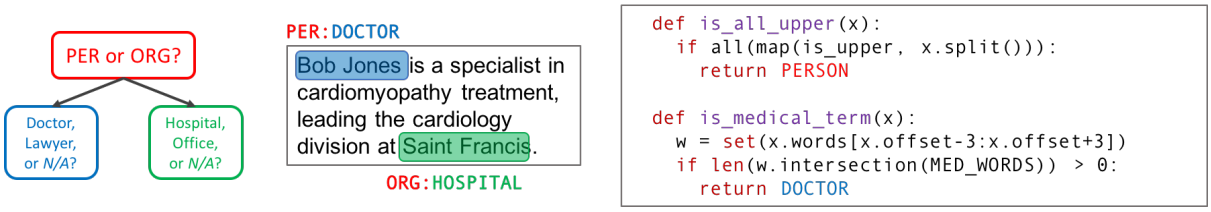

Figure 2: An example fine-grained entity classification problem, where weak supervision sources label three sub-tasks of different granularities: (i) `Person` vs. `Organization`, (ii) `Doctor` vs. `Lawyer` (or *N/A*), (iii) `Hospital` vs. `Office` (or *N/A*). The example weak supervision sources use a pattern-based heuristic and dictionary lookup respectively.

label a large number of items (data points). Moreover, we theoretically characterize the generalization performance of an end model trained with the weakly labeled data.

Multi-Task Learning: Our proposed approach is motivated by recent progress on multi-task learning models [6; 31; 32], in particular their need for multiple large hand-labeled training datasets. We note that the focus of our paper is on generating supervision for these models, not on the particular multi-task learning model being trained, which we seek to control for by fixing a simple architecture in our experiments.

Our work is also related to recent techniques for estimating classifier accuracies without labeled data in the presence of structural constraints [28]. We use matrix structure [24] and concentration bounds [34] for our core results.

3 Programming Machine Learning with Weak Supervision

As modern machine learning models become both more complex and more performant on a range of tasks, developers increasingly interact with them by programmatically generating noisier or *weak* supervision. These approaches of effectively *programming* machine learning models [18] by programmatically generating training labels generally proceed as follows [30; 29]: First, users provide one or more *weak supervision sources*, which are applied to unlabeled data to generate a set of noisy labels. These labels overlap and conflict; we model and combine them via a *label model* in order to produce a set of training labels. These weak labels are then used to train a discriminative model, which we refer to as the *end model*.

In our experiences with users from science and industry, we have found it critical to utilize all available sources of weak supervision for complex modeling problems, including ones which label at multiple levels of *granularity*. However, this diverse, multi-granular weak supervision does not easily fit into existing paradigms. We propose a formulation where each weak supervision source labels some sub-task of a problem, which we refer to as the *multi-task weak supervision* setting. We consider an example:

Example 1 *A developer wants to train a fine-grained Named Entity Recognition (NER) model to classify mentions of entities in the news (Figure 2). She has a multitude of available weak supervision sources which she believes have relevant signal for her problem—for example, pattern matchers, dictionaries, and pre-trained generic NER taggers. However, it is unclear how to properly use and combine them: some of them label phrases coarsely as* `PERSON` *versus* `ORGANIZATION`*, while others classify specific fine-grained types of people or organizations, with a range of unknown accuracies. In our framework, she can represent them as labeling tasks of different granularities—e.g.* $Y_1 = \{\texttt{Person}, \texttt{Org}\}$, $Y_2 = \{\texttt{Doctor}, \texttt{Lawyer}, \texttt{N/A}\}$, $Y_3 = \{\texttt{Hospital}, \texttt{Office}, \texttt{N/A}\}$, *where the label* `N/A` *applies when for example when the type-of-person task is applied to an organization.*

In our proposed multi-task supervision setting, the user specifies a set of structurally-related *tasks*, and then provides a set of *weak supervision sources* which are user-defined functions that label each data point for each task or abstain, and may have some user-specified dependency structure. Our goal is to estimate the unknown accuracies of these sources, combine their outputs, and use the resulting labels to train an end model.

4 Modeling Multi-Task Weak Supervision

The core technical challenge of the *multi-task weak supervision* setting is recovering the unknown *accuracies* of weak supervision sources given their dependency structure and a schema of the tasks they label, but without any ground-truth labeled data. We define a new algorithm for recovering the accuracies in this setting using a low-rank matrix completion approach. We establish conditions under which the resulting estimator returns a unique

solution, and show how the estimation error affects the generalization performance of the end model we aim to train. Afterwards, we analyze the sample complexity of our estimator, characterizing its scaling with respect to the amount of *unlabeled data*, as well as the task schema and dependency structure. Finally, we show how it can be extended to handle abstentions and *unipolar* sources, two critical scenarios in the weak supervision setting.

4.1 A Multi-Task Weak Supervision Estimator

Problem Setup Let $X \in \mathcal{X}$ be a data point and $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_t]^T$ be a vector of categorical *task labels*, $Y_i \in \{1, \ldots, k_i\}$, corresponding to t tasks, where $(X, \mathbf{Y})$ is drawn i.i.d. from a distribution $\mathcal{D}$.[1] The user provides a specification of how these tasks relate to each other; we denote this schema as the *task structure* G_{task}, which defines a feasible set of label vectors $\mathcal{Y}$, such that $\mathbf{Y} \in \mathcal{Y}$. For example, Figure 2 illustrates a hierarchical task structure over three tasks of different granularities that are related by logical implication relationships: if $Y_1 = \texttt{PERSON}$, then $Y_3 = \texttt{N/A}$, since the source Y_3 labeling types organizations is not applicable to persons. Thus, in this task structure, $\mathbf{Y} = [\texttt{PERSON}, \texttt{DOCTOR}, \texttt{N/A}]$ is in $\mathcal{Y}$ while $\mathbf{Y} = [\texttt{PERSON}, \texttt{N/A}, \texttt{HOSPITAL}]$ is not.

In our setting, rather than observing the true label $\mathbf{Y}$, we have access to m *multi-task weak supervision* sources $s_i \in S$, which emit label vectors $\boldsymbol{\lambda}_i$ that contain labels for some subset of the t tasks. Let 0 denote a null or abstaining label, and let the *coverage set* $\tau_i \subseteq \{1, \ldots, t\}$ be the fixed set of tasks for which source s_i emits non-zero labels, such that $s_i : \mathcal{X} \mapsto \mathcal{Y}_{\tau_i}$. For example, a coarse-grained source s_i from our previous example might have a coverage set $\tau_i = \{1, 3\}$, emitting labels such as $\boldsymbol{\lambda}_i = [\texttt{PERSON}, 0, \texttt{N/A}]$. Note that sources often label multiple tasks implicitly due to the constraints of the task structure; for example, a source that labels types of people (Y_2) also implicitly labels people vs. organizations ($Y_1 = \texttt{PERSON}$), and types of organizations (as $Y_3 = \texttt{N/A}$). Thus sources tailored to different tasks still have agreements and disagreements; we use this extra signal in our approach.

The user also provides the conditional dependency structure of the sources as a graph G_{source}. Specifically, if (i, j) is not an edge in G_{source}, this means that $\boldsymbol{\lambda}_i$ is independent of $\boldsymbol{\lambda}_j$ conditioned on $\mathbf{Y}$ and the other source labels. Importantly, we do not know anything about the sources' accuracies, or the strengths of their correlations. Note that if G_{source} is unknown, it can be estimated using statistical techniques such as [2].

Our overall goal is to apply the sources S to an unlabeled dataset $\mathcal{X}_U$ consisting of n data points, then use the resulting weakly-labeled training set to supervise an *end model* $f_w : \mathcal{X} \mapsto \mathcal{Y}$ (Figure 1). To do this, we will learn a *label model* $P_\mu(\mathbf{Y}|\boldsymbol{\lambda})$ which takes as input the noisy labels and outputs a single probabilistic label $\tilde{\mathbf{Y}}$ for each X. Succinctly, given a user-provided tuple $(\mathcal{X}_U, S, G_{\text{source}}, G_{\text{task}})$, our goal is to recover the parameters μ. The key technical challenge is then estimating μ without access to ground truth labels $\mathbf{Y}$.

Modeling Multi-Task Sources We start by considering a simple generative model of our noisy labeling process. We model each source s_i with a single conditional accuracy parameter, $\alpha_i := P(\boldsymbol{\lambda}_i = (y)_{\tau_i} | \mathbf{Y} = y)$. Here, $(\cdot)_{\tau_i}$ only includes entries in the coverage set τ_i, as these are the only tasks that s_i labels. In other words, we use the same accuracy parameter for all labels $\mathbf{Y}$, and assume that each source labels incorrectly with probability $1 - \alpha_i$ uniformly over the incorrect classes—a commonly considered model [11; 30]. In the Appendix, we cover more general models that our framework can handle, including incorporating abstentions and *task*- or *label*-conditional models.

Next, we introduce a series of random variables that encode the desired accuracies α_i while simultaneously having observable products. The general thrust of our approach is to factor these observed products to recover the accuracies. We define a random variable row vector $\phi_i \in \{-1, 0, 1\}^{1 \times r}$ for each source s_i, where $r = |\mathcal{Y}|$, such that:

$$(\phi_i)_s = \mathbb{1}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\} \mathbb{1}^{\pm}\left\{\mathbf{Y} = y_s\right\}$$

for $y_s \in \mathcal{Y}$, and where $\mathbb{1}^{\pm}\{a = b\} = \mathbb{1}\{a = b\} - \mathbb{1}\{a \neq b\}$. We additionally define an extended set of random variables to allow us to capture the statistics of the cliques $C \in \mathcal{C}$ of G_{source}, $\phi \in \{-1, 0, 1\}^{|\mathcal{C}| \times r}$, which includes random variables covering all cliques; we detail the explicit form in the Appendix. Our goal then is to estimate the parameters of our label model, $\mu = \mathbb{E}[\phi]$, which capture the source accuracies α_i.

Note that while ϕ_i is not observable due to its dependence on $\mathbf{Y}$, $\phi_i \phi_j := \phi_i \phi_j^T$ is in fact observable, acting as an indicator for when sources i, j agree (scaled by the observable size of their overlap):

$$\phi_i \phi_j = \sum_{y \in \mathcal{Y}} \mathbb{1}\left\{\boldsymbol{\lambda}_i = (y)_{\tau_i}\right\} \mathbb{1}\left\{\boldsymbol{\lambda}_j = (y)_{\tau_j}\right\}.$$

[1] The variables we introduce throughout this section are summarized in a glossary in the Appendix.

The fact that $\phi_i \phi_j$ is observable undergirds our approach, providing enough signal to recover μ without observing the true label $\mathbf{Y}$.

Our Approach We proceed by analyzing the covariance matrix of ϕ, leading to a low-rank matrix completion approach for recovering μ. We leverage two pieces of information: (i) the observability of $\phi_i \phi_j$ mentioned in the previous section, and (ii) a result from Loh & Wainwright [24] which states that the inverse covariance matrix is structured according to G_{source}. Specifically, if there is no edge between s_i and s_j in G_{source}, then the corresponding entry of the inverse covariance matrix is 0.

We start with the covariance matrix of ϕ:

$$\Sigma = \mathbf{Cov}\left[\phi\right] = \mathbb{E}\left[\phi\phi^T\right] - \mu\mu^T. \tag{1}$$

While $\mu = \mathbb{E}\left[\phi\right] \in \mathbb{R}^{|\mathcal{C}| \times r}$, given our choice to model each source with a single parameter, by simple algebra we can reduce the rank r term $\mu\mu^T$ to a rank-one term plus an observable additive term, $\mu\mu^T = \mu'\mu'^T + O_{\text{rank-one}}$ where $\mu' \in \mathbb{R}^{|\mathcal{C}| \times 1}$; we can then directly recover μ from μ' (see Appendix). In this way we reduce (1) to:

$$\Sigma = O - \mu'\mu'^T, \tag{2}$$

where we denote $O = \mathbb{E}\left[\phi\phi^T\right] + O_{\text{rank-one}}$ as the *overlaps matrix*. Crucially, O is empirically observable. Equation (2) is therefore close to a rank-one matrix factorization objective, except that we cannot observe Σ. Using the Sherman-Morrison formula we write:

$$\Sigma^{-1} = \left(O - \mu'\mu'^T\right)^{-1} = O^{-1} + zz^T, \tag{3}$$

where $c = (1 - \mu'^T O^{-1} \mu')^{-1}$ and $z = \sqrt{c} O^{-1} \mu'$. This suggests an algorithmic approach: We first observe O and compute O^{-1}. Since Σ^{-1} is graph structured according to G_{source} by [24], we know the zero entries of Σ^{-1}, enabling us to estimate z, and thereby recover an estimate of μ' and μ (Algorithm 2). In more detail: let Ω be the set of indices (i, j) where $\Sigma^{-1}_{i,j} = 0$, determined by G_{source}, yielding a system of equations,

$$0 = O^{-1}_{i,j} + \left(zz^T\right)_{i,j} \text{ for } (i, j) \in \Omega. \tag{4}$$

Define $||A||_\Omega$ as the Frobenius norm of A with entries not in Ω set to zero; then we can rewrite (4) as $\left|\left|O^{-1} + zz^T\right|\right|_\Omega = 0$. We solve this equation to estimate z, and thereby recover μ', and thus μ.

Checking for Identifiability A first question is which dependency structures G_{source} lead to unique solutions of μ? This question presents a stumbling block for users, who might attempt to use non-identifiable sets of correlated weak supervision sources.

We provide a simple, testable condition for identifiability. Let G_{inv} be the inverse graph of G_{source}; note that Ω is the edge set of G_{inv}. Then, let M_Ω be a matrix with dimensions $|\Omega| \times m$ such that each row in M_Ω corresponds to a pair $(i, j) \in \Omega$ with 1's in positions i and j and 0's elsewhere.

Taking the log of the squared entries of (4), we get a system of linear equations $M_\Omega l = q_\Omega$, where $l_i = \log(z_i^2)$ and $q_{(i,j)} = \log((O^{-1}_{i,j})^2)$. Assuming we can solve this system of linear equations (which we can always ensure by adding sources; see Appendix), this yields z_i^2, meaning our model is identifiable *up to sign*.

Given estimates of z_i^2, the sign of a single z_i determines the sign of all other z_j reachable from z_i in G_{inv}. Thus to ensure a unique solution, we only need to pick a sign for each connected component in G_{inv}. In the case the sources are assumed to be independent, e.g., [10; 40; 11], it suffices to make the assumption that the sources are *on average* non-adversarial, i.e. select the sign of the z_i that leads to the higher average accuracy μ_i. Even a single source that is conditionally independent from all the other sources will cause G_{inv} to be fully connected, meaning we can use this symmetry breaking assumption in the majority of cases even with correlated sources. Otherwise, a sufficient condition is the standard one of non-adversarial sources.

Source Accuracy Estimation Algorithm Now that we know when a set of sources with correlation structure G_{source} is identifiable, yielding z, we can estimate the accuracies μ with Algorithm 2. In Figure 3, we plot the performance of our algorithm on synthetic data, showing its scaling with n, density of pairwise correlation structure G_{source}, and runtime performance as compared to prior approaches. Next, we theoretically analyze the scaling of the error $||\hat{\mu} - \mu^*||$.

Algorithm 1 Source Accuracy Estimation for Multi-Task Weak Supervision

Input: Empirical overlaps matrix $\hat{O} \in \mathbb{R}^{|\mathcal{C}| \times |\mathcal{C}|}$, correlation sparsity structure Ω

$\hat{z} \leftarrow \text{argmin}_z \left|\left|\hat{O}^{-1} + zz^T\right|\right|_\Omega$

$\hat{c} \leftarrow 1 + \hat{z}^T\hat{O}\hat{z}$, $\hat{\mu} \leftarrow \hat{O}\hat{z}/\sqrt{\hat{c}}$

return $\hat{\mu}$

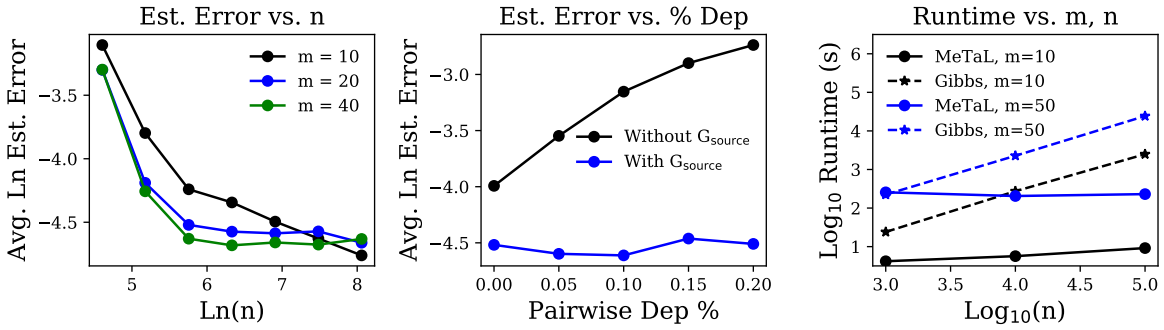

Figure 3: (Left) Estimation error $||\hat{\mu} - \mu^*||$ decreases with increasing n. (Middle) Given G_{source}, our model successfully recovers the source accuracies even with many pairwise dependencies among sources, where a naive conditionally-independent model fails. (Right) The runtime of `MeTaL` is independent of n after an initial matrix multiply, and can thus be multiple orders of magnitude faster than Gibbs sampling-based approaches [30].

4.2 Theoretical Analysis: Scaling with Diverse Multi-Task Supervision

Our ultimate goal is to train an *end model* using the source labels, denoised and combined by the label model $\hat{\mu}$ we have estimated. We connect the generalization error of this end model to the estimation error of Algorithm 2, ultimately showing that the generalization error scales as $n^{-\frac{1}{2}}$, where n is the number of unlabeled data points. This key result establishes the same asymptotic scaling as traditionally supervised learning methods, but with respect to *unlabeled* data points.

Let $P_{\hat{\mu}}(\tilde{\mathbf{Y}} \mid \boldsymbol{\lambda})$ be the probabilistic label predicted by our estimated label model, given the source labels $\boldsymbol{\lambda}$ as input, which we compute using the estimated $\hat{\mu}$. We then train an *end* multi-task discriminative model $f_w : \mathcal{X} \mapsto \mathcal{Y}$ parameterized by w, by minimizing the expected loss with respect to the label model over n unlabeled data points. Let $l(w, X, \mathbf{Y}) = \frac{1}{t}\sum_{s=1}^{t} l_t(w, X, \mathbf{Y}_s)$ be a bounded multi-task loss function such that without loss of generality $l(w, X, \mathbf{Y}) \leq 1$; then we minimize the empirical *noise aware loss*:

$$\hat{w} = \text{argmin}_w \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\tilde{\mathbf{Y}} \sim P_{\hat{\mu}}(\cdot|\boldsymbol{\lambda})} \left[l(w, X_i, \tilde{\mathbf{Y}}) \right], \tag{5}$$

and let $\tilde{w}$ be the w that minimizes the true noise-aware loss. This minimization can be performed by standard methods and is not the focus of our paper; let the solution $\hat{w}$ satisfy $\mathbb{E}\left[\|\hat{w} - \tilde{w}\|^2\right] \leq \gamma$. We make several assumptions, following [30]: (1) that for some label model parameters μ^*, sampling $(\boldsymbol{\lambda}, \mathbf{Y}) \sim P_{\mu^*}(\cdot)$ is the same as sampling from the true distribution, $(\boldsymbol{\lambda}, \mathbf{Y}) \sim \mathcal{D}$; and (2) that the task labels Y_s are independent of the features of the end model features given $\boldsymbol{\lambda}$ sampled from $P_{\mu^*}(\cdot)$, that is, the output of the optimal label model provides sufficient information to discern the true label. We also rely on two technical conditions: an assumption that sources vote based on the unlabeled data (rather than entirely independently), and a simple marginal coherency condition for observability. These conditions are described in depth in the Appendix. Then we have the following result:

Theorem 1 *Let $\tilde{w}$ minimize the expected noise aware loss, using weak supervision source parameters $\hat{\mu}$ estimated with Algorithm 2. Let $\hat{w}$ minimize the empirical noise aware loss with $\mathbb{E}\left[\|\hat{w} - \tilde{w}\|^2\right] \leq \gamma$, $w^* = \min_w l(w, X, \boldsymbol{Y})$, and let the assumptions above hold. Then the generalization error is bounded by:*

$$\mathbb{E}\left[l(\hat{w}, X, \boldsymbol{Y}) - l(w^*, X, \boldsymbol{Y})\right] \leq \gamma + 4|\mathcal{Y}|\,||\hat{\mu} - \mu^*||\,.$$

Thus, to control the generalization error, we must control $||\hat{\mu} - \mu^*||$, which we do in Theorem 2:

Theorem 2 *Let $\hat{\mu}$ be an estimate of μ^* produced by Algorithm 2 run over n unlabeled data points. Let $a :=$*

	NER	RE	Doc	Average
Gold (Dev)	63.7 ± 2.1	28.4 ± 2.3	62.7 ± 4.5	51.6
MV	76.9 ± 2.6	43.9 ± 2.6	74.2 ± 1.2	65.0
DP [30]	78.4 ± 1.2	49.0 ± 2.7	75.8 ± 0.9	67.7
MeTaL	**82.2** ± 0.8	**56.7** ± 2.1	**76.6** ± 0.4	**71.8**

Table 1: **Performance Comparison of Different Supervision Approaches.** We compare the micro accuracy (avg. over 10 trials) with 95% confidence intervals of an end multi-task model trained using the hand-labeled development set (Gold Dev), hierarchical majority vote (MV), data programming (DP), and our approach (MeTaL).

$\left(\frac{1}{|\mathcal{C}|} - \lambda_{\min}^{-1}(O)\right)^{-\frac{1}{2}}$ *and* $b := \frac{\|O^{-1}\|^2}{O_{\min}^{-1}}$. *Then, we have:*

$$\mathbb{E}\left[||\hat{\mu} - \mu^*||\right] \leq |\mathcal{C}|^2 \sqrt{\frac{32\pi}{n}} \left[(3\sqrt{|\mathcal{C}|}a\lambda_{\min}^{-1}(O) + 1) \times \left(2\sqrt{2}ab\sigma_{\max}(M_\Omega^+)\left[\kappa(O) + \lambda_{\min}^{-1}(O)\right]\right)\right].$$

Interpreting the Bound We briefly explain the key terms controlling the bound in Theorem 2; more detail is found in the Appendix. Our key result is that the estimation error scales as $n^{-\frac{1}{2}}$. Next, $\sigma_{\max}(M_\Omega^+)$, the largest singular value of the pseudoinverse M_Ω^+, has a deep connection to the density of the graph G_{inv}. The smaller this quantity, the more information we have about G_{inv}, and the easier it is to estimate the accuracies. Next, $\lambda_{\min}(O)$, the smallest eigenvalue of the observed matrix, reflects the conditioning of O; better conditioning yields easier estimation. Finally, $O_{\min}^{-1}$, the smallest entry of the inverse observed matrix, reflects the smallest non-zero correlation between source accuracies; distinguishing between small correlations and independencies requires more samples.

4.3 Extensions: Abstentions & Unipolar Sources

Handling Abstentions One fundamental aspect of the weak supervision setting is that sources may abstain from labeling a data point entirely—that is, they may have incomplete and differing coverage [29; 10]. We can easily deal with this case by extending the coverage ranges $\mathcal{Y}_{\tau_i}$ of the sources to include the vector of all zeros, $\vec{0}$, and we do so in the experiments.

Handling Unipolar Sources Finally, we consider the extension of our framework to modeling separate *class conditional* source accuracies, in particular motivated by the case we have frequently observed in practice of *unipolar* weak supervision sources, i.e. sources that each only label a single class or abstain. In this approach, we aim to learn a separate source accuracy parameter $\alpha_{i,y} := P(\boldsymbol{\lambda}_i = (y)_{\tau_i} | \mathbf{Y} = y)$ for each $\mathbf{Y}$. Finally, we also consider the fully general case, where we model the probabilities of sources emitting each possible output, conditioned on the various choices of $\mathbf{Y}$. These approaches are detailed in the Appendix.

This joint-class formulation requires an estimate of p, the vector of class balances $p(\mathbf{Y})$ for $\mathbf{Y} \in \mathcal{Y}$ to determine the class-conditional accuracies. While we can often simply estimate p from a small labeled sample, in the Appendix we provide a three-way tensor decomposition approach for estimating p. In the unipolar setting, we can use this approach to yield an improvement of 2.8 points in accuracy in our experiments.

5 Experiments

We validate our approach on three fine-grained classification problems—entity classification, relation classification, and document classification—where weak supervision sources are available at both coarser and finer-grained levels (e.g. as in Figure 2). We evaluated the predictive accuracy of end models supervised with training data produced by several approaches, finding that our approach outperforms traditional hand-labeled supervision by 20.2 points, a baseline majority vote weak supervision approach by 6.8 points, and a prior weak supervision denoising approach [30] that is not multi-task-aware by 4.1 points.

Datasets Each dataset consists of a large (3k-63k) amount of unlabeled training data and a small (200-350) amount of labeled data which we refer to as the *development set*, which we use for (a) a traditional supervision baseline, and (b) for hyperparameter tuning of the end model (see Appendix). The average number of sources per task was 13, with sources expressed as Python functions, averaging 4 lines of code and comprising a mix of pattern matching, external knowledge base or dictionary lookup, and pre-trained models.

Named Entity Recognition (NER): We represent a fine-grained named entity recognition problem—i.e. tagging entity mentions in text documents—as a hierarchy of three sub-tasks over the OntoNotes dataset [36]: $Y_1 \in \{\text{Person}, \text{Organization}\}$, $Y_2 \in \{\text{Businessperson}, \text{Other Person}, \textit{N/A}\}$, $Y_3 \in \{\text{Company}, \text{Other Org}, \textit{N/A}\}$, where again we use *N/A* to represent "not applicable".

Relation Extraction (RE): We represent a relation extraction problem—i.e. classifying entity-entity relation mentions in text documents—as a hierarchy of six sub-tasks which either concern labeling the subject, object, or subject-object pair of a *candidate* relation in the TACRED dataset [41]. For example, we might classify a relation as having a Person subject, Location object, and Place-of-Residence relation type.

Medical Document Classification (Doc): We represent a radiology report triaging—i.e. document classification—problem from the OpenI dataset [27] as a hierarchy of three sub-tasks: $Y_1 \in \{\text{Acute}, \text{Non-Acute}\}$, $Y_2 \in \{\text{Urgent}, \text{Emergent}, \textit{N/A}\}$, $Y_3 \in \{\text{Normal}, \text{Non-Urgent}, \textit{N/A}\}$.

End Model Protocol Our goal was to select a basic multi-task end model class, to test its performance with training labels produced by various different approaches. We use an architecture consisting of a bidirectional LSTM input layer with pre-trained embeddings, d linear intermediate layers, and a final linear layer ("task head") for each supervision task, attached to the intermediate layer corresponding to its level in the problem task structure—thus mirroring the structure of G_{task}. A hyperparameter search was initially performed for each application over layer sizes, embedding types, and dropout and regularization, then fixed for the experiments.

Core Validation We compare the accuracy of an end multi-task model trained with labels from our approach versus those trained with labels from three baseline approaches (Table 1):

- *Traditional Supervision* **[Gold (Dev)]**: We train the end model using the hand-labeled data points in the development set.
- *Hierarchical Majority Vote* **[MV]**: We use a hierarchical majority vote of the weak supervision source labels: i.e. for each data point, for each task we take the majority vote and proceed down the task tree accordingly. This procedure can be thought of as a hard decision tree, or a cascade of if-then statements that might occur in a rule-based approach.
- *Data Programming* **[DP]**: We model each task separately using the data programming approach for denoising weak supervision [29].

In all settings, we used the same end model architecture as described above. Note that while we choose to model these problems as consisting of multiple sub-tasks, we evaluate with respect to the broad primary task of fine-grained classification (for subtask-specific scores, see Appendix). We observe in Table 1 that our approach of leveraging multi-granularity weak supervision leads to large gains—20.2 points over traditional supervision with the development set, 6.8 points over hierarchical majority vote, and 4.1 points over data programming.

Ablations We examine individual factors:

Unipolar Correction: Modeling unipolar sources (Sec 4.3), which we find to be especially common when fine-grained tasks are involved, leads to an average gain of 2.8 points of accuracy in `MeTaL` performance.

Joint Task Modeling: Next, we use our algorithm to estimate the accuracies of sources for each task separately, to observe the empirical impact of modeling the multi-task setting jointly as proposed. We see average gains of 1.3 points in accuracy (see Appendix).

End Model Generalization: Though not possible in many settings, in our experiments we can directly apply the label model to make predictions. In Table 5, we show that the end model improves performance by an average 3.4 points in accuracy, validating that the models trained do indeed learn to generalize beyond the provided weak supervision. Moreover, the largest generalization gain came from the dataset with the most available unlabeled data (n=63k), demonstrating scaling consistent with the predictions of our theory (Fig. 4). This ability to leverage additional unlabeled data and more sophisticated models are key advantages of the weak supervision approach in practice.

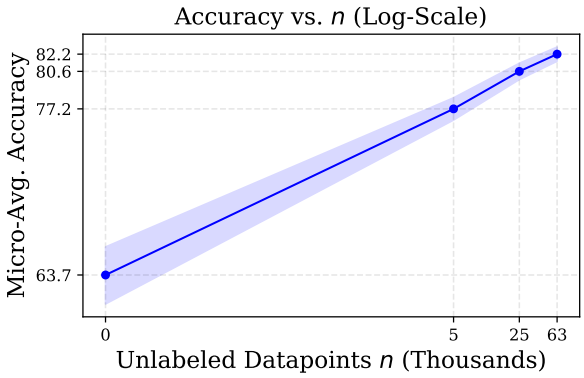

Figure 4: In the OntoNotes dataset, quality scales with the amount of available *unlabeled* data.

	# Train	LM	EM	*Gain*
NER	62,547	75.2	82.2	*7.0*
RE	9,090	55.3	57.4	*2.1*
Doc	2,630	75.6	76.6	*1.0*

Figure 5: Using the label model (LM) predictions directly does not perform as well as using them to train an end model (EM).

6 Conclusion

We presented `MeTaL`, a framework for training models with weak supervision from diverse, *multi-task* sources with different granularities, accuracies, and correlations. We tackle the core challenge of recovering the unknown source accuracies via a matrix-completion style approach, introduced a scalable algorithm with sharper theoretical bounds and empirical gains on real-world datasets. In future work, we hope to learn the dependency structure and cover a broader range of settings where labeled training data is a bottleneck.

References

[1] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[2] S. H. Bach, B. He, A. J. Ratner, and C. Ré. Learning the structure of generative models without labeled data, 2017.

[3] A. Bhaskara, M. Charikar, and A. Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability, 2014.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training, 1998.

[5] L. Cambier and P.-A. Absil. Robust low-rank matrix completion by riemannian optimization. *SIAM Journal on Scientific Computing*, 2016.

[6] R. Caruana. Multitask learning: A knowledge-based source of inductive bias, 1993.

[7] F. R. K. Chung. Laplacians of graphs and cheeger inequalities. 1996.

[8] M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources, 1999.

[9] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi. Aggregating crowdsourced binary ratings, 2013.

[10] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi. Aggregating crowdsourced binary ratings, 2013.

[11] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[12] A. Ghosh, S. Kale, and P. McAfee. Who moderates the moderators?: Crowdsourcing abuse detection in user-generated content, 2011.

[13] M. Y. Guan, V. Gulshan, A. M. Dai, and G. E. Hinton. Who said what: Modeling individual labelers improves classification. *arXiv preprint arXiv:1703.08774*, 2017.

[14] S. Gupta and C. D. Manning. Improved pattern learning for bootstrapped entity extraction., 2014.

[15] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations, 2011.

[16] J. Honorio. Lipschitz parametrization of probabilistic graphical models. *arXiv preprint arXiv:1202.3733*, 2012.

[17] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems, 2011.

[18] A. Karpathy. Software 2.0. `medium.com/@karpathy/software-2-0-a64152b37c35`.

[19] A. Khetan, Z. C. Lipton, and A. Anandkumar. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.

[20] F. Király and R. Tomioka. A combinatorial algebraic approach for the identifiability of low-rank matrix completion. *arXiv preprint arXiv:1206.6470*, 2012.

[21] O. Klopp, K. Lounici, and A. B. Tsybakov. Robust matrix completion. *arXiv preprint arXiv:1610.08123*, 2016.

[22] J. B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

[23] P. Liang, M. I. Jordan, and D. Klein. Learning from measurements in exponential families, 2009.

[24] P.-L. Loh and M. J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses, 2012.

[25] G. S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 11(Feb):955–984, 2010.

[26] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data, 2009.

[27] National Institutes of Health. Open-i. 2017.

[28] E. Platanios, H. Poon, T. M. Mitchell, and E. J. Horvitz. Estimating accuracy from unlabeled data: A probabilistic logic approach, 2017.

[29] A. Ratner, S. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision, 2018.

[30] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly, 2016.

[31] S. Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[32] A. Søgaard and Y. Goldberg. Deep multi-task learning with low level tasks supervised at lower layers, 2016.

[33] S. Takamatsu, I. Sato, and H. Nakagawa. Reducing wrong labels in distant supervision for relation extraction, 2012.

[34] J. A. Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[35] P. Varma, B. D. He, P. Bajaj, N. Khandwala, I. Banerjee, D. Rubin, and C. Ré. Inferring generative model structure with static analysis, 2017.

[36] R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue. Ontonotes: A large training corpus for enhanced processing. *Handbook of Natural Language Processing and Machine Translation. Springer*, 2011.

[37] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification, 2015.

[38] O. F. Zaidan and J. Eisner. Modeling annotators: A generative approach to learning from annotator rationales, 2008.

[39] C. Zhang, C. Ré, M. Cafarella, C. De Sa, A. Ratner, J. Shin, F. Wang, and S. Wu. DeepDive: Declarative knowledge base construction. *Commun. ACM*, 60(5):93–102, 2017.

[40] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing, 2014.

[41] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning. Position-aware attention and supervised data improve slot filling, 2017.

A Problem Setup & Modeling Approach

In Section A, we review our problem setup and modeling approach in longform. In Section B, we provide an overview, additional interpretation, and the proofs of our main theoretical results. Finally, in Section C, we go over additional details of our experimental setup.

We begin in Section A.1 with a glossary of the symbols and notation used throughout this paper. Then, in Section A.2 we present the setup of our multi-task weak supervision problem, and in Section A.3.1 we present our approach for modeling multi-task weak supervision, and the *masked* low-rank matrix approximation approach used to estimate the model parameters. Finally, in Section A.4, we present in more detail the subcase of hierarchical tasks considered in the main body of the paper.

A.1 Glossary of Symbols

Symbol	Used for
X	Data point, $X \in \mathcal{X}$
n	Number of data points
t	Number of tasks
Y_s	Label for one of the t classification tasks, $Y_s \in \{1, \ldots, k_s\}$
$\mathbf{Y}$	Vector of task labels $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_t]^T$
r	Cardinality of the output space, $r = \lvert\mathcal{Y}\rvert$
G_{task}	Task structure graph
$\mathcal{Y}$	Output space of allowable task labels defined by G_{task}, $\mathbf{Y} \in \mathcal{Y}$
$\mathcal{D}$	Distribution from which we assume $(X, \mathbf{Y})$ data points are sampled i.i.d.
$s_i(x), s_i$	Weak supervision source, a function mapping X to a label vector
m	Number of sources
$\boldsymbol{\lambda}_i$	Label vector $\boldsymbol{\lambda}_i \in \mathcal{Y}$ output by the ith source for X
$\boldsymbol{\lambda}$	$m \times t$ matrix of labels output by the m sources for X
$\mathcal{Y}_0$	Source output space, which is $\mathcal{Y}$ augmented to include elements set to zero
τ_i	Coverage set of $\boldsymbol{\lambda}_i$- the tasks s_i gives non-zero labels to
$\mathcal{Y}_{\tau_i}$	The output space for $\boldsymbol{\lambda}_i$ given coverage set τ_i
$\phi(i, y)$	$1 \times r$ row vector indicator random variable for $\boldsymbol{\lambda}_i = y$ *and* different values of $\mathbf{Y}$
$\phi_{\mathbf{Y}}$	$1 \times r$ row vector indicator random variable for different values of $\mathbf{Y}$
G_{source}	Source dependency graph
$\mathcal{C}$	Cliqueset (maximal and non-maximal) of G_{source}
ψ	Augmented vector of G_{source} clique statistics
Σ	Generalized covariance matrix $\Sigma \equiv \mathbf{Cov}[\psi]$
μ	Label model parameters, $\mu = \mathbb{E}[\psi]$
θ	Log parameters $\log(\mu)$
α	Conditional parameters $(\alpha_{C,y_c})_y = P(\boldsymbol{\lambda}_C = y_C \lvert \mathbf{Y} = y) = (\mu_{C,y_C})_y / P(\mathbf{Y} = y)$
O	Observable overlaps matrix $O = \mathbb{E}\left[\psi\psi^T\right]$
Ω	Inverse augmented edge set: Cliques $(A, B) \in \Omega$ iff A, B not part of the same maximal clique
P	Diagonal matrix of class prior probabilities, $P(\mathbf{Y})$
$P_\mu(\mathbf{Y}, \boldsymbol{\lambda})$	The *label model* parameterized by μ
$\tilde{\mathbf{Y}}$	The probabilistic training label, i.e. $P_\mu(\mathbf{Y} \lvert \boldsymbol{\lambda})$
$f_w(X)$	The *end model* trained using $(X, \tilde{\mathbf{Y}})$

Table 2: Glossary of variables and symbols used in this paper. Note that some symbols above are provided as "X, x", where x is the shorthand version used for notational cleanliness below.

A.2 Problem Setup

Let $X \in \mathcal{X}$ be a data point and $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_t]^T$ be a vector of *task labels* corresponding to t tasks. We consider categorical task labels, $Y_s \in \mathcal{Y}_s$, with cardinality k, for $s \in \{1, \ldots, t\}$. We assume $(X, \mathbf{Y})$ pairs are sampled i.i.d. from distribution $\mathcal{D}$; to keep the notation manageable, we do not place subscripts on the sample tuples.

Task Structure The tasks are related by a *task graph* G_{task}. In full generality, we consider the task graph as a structure which defines a feasible set of task vectors, $\mathcal{Y}$, such that $\mathbf{Y} \in \mathcal{Y}$. We let $r = |\mathcal{Y}|$ be the number of feasible task vectors. In section A.4, we consider the particular subcase of a *hierarchical* task structure as used in the experiments section of the paper.

Multi-Task Sources We now consider *multi-task* weak supervision sources $s_i \in S$, which represent noisy and potentially incomplete sources of labels, which have unknown accuracies and correlations. Each source s_i outputs label vectors $\boldsymbol{\lambda}_i$, which contain non-zero labels for *some* of the tasks, such that $\boldsymbol{\lambda}_i$ is in the feasible set $\mathcal{Y}$ but potentially with some elements set to zero, denoting a null vote or abstention for that task. Let $\mathcal{Y}_0$ denote this extended set which includes certain task labels set to zero.

We also assume that each source has a fixed *task coverage set* τ_i, such that $(\boldsymbol{\lambda}_i)_s \neq 0$ for $s \in \tau_i$, and $(\boldsymbol{\lambda}_i)_s = 0$ for $s \notin \tau_i$; let $\mathcal{Y}_{\tau_i} \subseteq \mathcal{Y}_0$ be the range of $\boldsymbol{\lambda}_i$ given coverage set τ_i. The intuitive idea of the task coverage set is that some labelers may choose not to label certain tasks; Example 2 illustrates this notion.

Thus we have: $s_i : \mathcal{X} \mapsto \{0\} \cup \mathcal{Y}_{\tau_i}$, where, again, $\boldsymbol{\lambda}_i$ denotes the output of the function s_i.

Problem Statement Our overall goal is to use the noisy or *weak*, *multi-task* supervision from the m sources, $s_1, \ldots, s_m$, to supervise an *end model* $f_w : \mathcal{X} \mapsto \mathcal{Y}$. Since the sources have unknown accuracies, and will generally output noisy and incomplete labels that will overlap and conflict, our intermediate goal is to first learn a *label model* $P_\mu : \boldsymbol{\lambda} \mapsto [0,1]^{|\mathcal{Y}|}$ which takes as input the source labels and outputs a set of probabilistic label vectors, which can then be used to train the end model.

The key technical challenge in this approach then consists of learning the parameters of this label model—corresponding to the conditional accuracies of the sources (and, for technical reasons we shall shortly explain, cliques of correlated sources)—given that *we do not have access to the ground truth labels* $\mathbf{Y}$. We discuss our approach to overcoming this core technical challenge in the subsequent section.

A.3 Our Approach: Modeling Multi-Task Sources

Our goal is to estimate the parameters μ of a *label model* that produces probabilistic training labels given the observed source outputs, $\tilde{\mathbf{Y}} = P_\mu(\mathbf{Y}|\boldsymbol{\lambda})$, *without access to the ground truth labels* $\mathbf{Y}$. We do this in three steps:

1. First, we define random variables $\phi(i, y)$, which are *vectors* that encode indicators of particular events $(\boldsymbol{\lambda}_i, \mathbf{Y})$ occurring; then, given a conditional independence structure between the sources, we define a graphical model over the $\phi(i, y)$ and a random variable $\phi_{\mathbf{Y}}$ representing the unobserved true label $\mathbf{Y}$.
2. Next, we analyze the covariance matrix of an augmented sufficient statistics vector for this model, ψ:

$$\textbf{Cov}\left[\psi\right] = \Sigma = \mathbb{E}\left[\psi\psi^T\right] - \mu\mu^T, \tag{6}$$

   where we define $\mu = \mathbb{E}\left[\psi\right]$. We establish that given our formulation, $O = \mathbb{E}\left[\psi\psi^T\right]$ is observable. We then apply a result by Loh and Wainwright [24] to establish an approximate sparsity pattern of Σ^{-1}. This allows us to invert (6) and solve for μ using a matrix approximation-style algorithm.
3. Next, we describe how to recover the class balance $P(\mathbf{Y})$; with this and the estimate of μ, we then describe how to compute the probabilistic training labels $\tilde{Y} = P_\mu(\mathbf{Y}|\boldsymbol{\lambda})$.

Finally, we describe the simplified model we use in the main body and for our theoretical results, where we model each source with a single parameter, leading to a rank-one version of our algorithm.

A.3.1 Defining a Multi-Task Source Model

Now we are ready to define our source model.

Source Label-True Label Indicator RVs In the most general formulation of our approach, we encode the label-specific accuracies of the sources using the vector-valued random variable $\phi(i, y)$. Recall that $\mathcal{Y}$ is the feasible

set of task label vectors defined by our task graph G_{task}, and let $r = |\mathcal{Y}|$. Then, we define $\phi(i, y)$ as a $1 \times r$ row vector of indicator random variables for a source s_i emitting a label $y \in \mathcal{Y}_{\tau_i}$ *and* for values of the true label $\mathbf{Y}$:

$$\phi(i, y) = [\mathbb{1}\{\boldsymbol{\lambda}_i = y, \mathbf{Y} = y_1\}, \ldots, \mathbb{1}\{\boldsymbol{\lambda}_i = y, \mathbf{Y} = y_r\}]. \tag{7}$$

Note that

$$\phi(i, y) \in \{0_r^T, e_1^T, e_2^T, \ldots, e_r^T\}.$$

Here, 0_r is an all 0's vector of length r, while e_i is the ith unit basis vector of length r. It is easy to see why this holds: each indicator includes a different value of $\mathbf{Y}$, so at most one can fire at once.

The product of these random variables is simply the dot product of the vectors, so that for $y_i \in \mathcal{Y}_{\tau_i}$, $y_j \in \mathcal{Y}_{\tau_j}$:

$$\phi(i, y_i)\phi(j, y_j) := \phi(i, y_i)\phi(j, y_j)^T. \tag{8}$$

Note that here the ϕ's are row vectors, so that the product is a scalar random variable. This enables us to define a covariance matrix over the ϕ's.

Conditional Independence Structure We assume that we know the dependency structure of the sources *conditioned on* $\boldsymbol{Y}$, that is, we know which pairs or groups of sources are conditionally independent conditioned on all other sources and the true label $\mathbf{Y}$. To take advantage of this structure, we will create a graphical model as follows. We introduce a random variable representing $\mathbf{Y}$, which we also represent as a vector. Note that $y_1, y_2, \ldots, y_r$ are all the possible labels $\mathbf{Y}$:

$$\phi_{\mathbf{Y}} = [\mathbb{1}\{\mathbf{Y} = y_1\}, \mathbb{1}\{\mathbf{Y} = y_2\}, \ldots, \mathbb{1}\{\mathbf{Y} = y_r\}]. \tag{9}$$

We now have a model over random variables ϕ along with $\phi_{\mathbf{Y}}$, for which we know the dependency structure.

In other words, we know the dependency edges, provided as G_{source}. If there is no edge between source i and j in G_{source}, then we assume that $\phi(i, y) \perp \phi(j, y)$ conditioned on all the other sources and $\mathbf{Y}$ (for $i \neq j$). That is, $(i, j) \in G_{\text{source}}$ means that there is a group of edges in the probabilistic graphical model between the variables $\phi(i, y)$ and $\phi(j, y)$. Note that if G_{source} is unknown, there are various existing techniques for estimating it statistically [2] or even from static analysis if the sources are heuristic functions [35].

Augmented Sufficient Statistics Finally, we extend the random variables ϕ by defining a matrix of indicator statistics over all cliques in G_{source}, in order to estimate all the parameters needed for our label model P_μ. We assume that the provided G_{source} is *chordal*, meaning it has no chordless cycles of length greater than three; if not, the graph can easily be *triangulated* to satisfy this property, in which case we work with this augmented version. Let $\mathcal{C}$ be the set of maximal and non-maximal cliques in the chordal graph G_{source}. Let $d = \sum_{C \in \mathcal{C}} \prod_{i \in C} |\mathcal{Y}_{\tau_i}|$, which is the total number of combinations that sources in each clique can take, summed over all the cliques, and not counting abstains (since given the other statistics, these can be inferred). Recall that we defined $r = |\mathcal{Y}|$. Then, let $\psi \in \{0, 1\}^{d \times r}$ be the matrix of indicator variables for all possible values of each clique, conditioned on all possible true labels $\mathbf{Y} \in \mathcal{Y}$, so that

$$\psi(C, y_C) = [\mathbb{1}\{\cap_{i \in C} \boldsymbol{\lambda}_i = (y_C)_i, \mathbf{Y} = y_1\}, \ldots, \mathbb{1}\{\cap_{i \in C} \boldsymbol{\lambda}_i = (y_C)_i, \mathbf{Y} = y_r\}], \tag{10}$$

where $(y_C)_i \in \mathcal{Y}_{\tau_i}$ for $i \in C$, $C \in \mathcal{C}$. We also define the product $\psi(C, y_C)\psi(D, y_D) = \psi(C, y_C)\psi(D, y_D)^T$ just as with ϕ, and let $\psi_{\mathbf{Y}} = \phi_{\mathbf{Y}}$. That is, ψ is the extended version of ϕ which includes indicators for all *cliques* of sources in G_{source}.[2] Our model estimation goal is now stated simply: we wish to estimate $\mu = \mathbb{E}[\psi]$, *without access to the ground truth labels* $\boldsymbol{Y}$.

A.3.2 Model Estimation without Ground Truth Using Inverse Covariance Structure

Our goal now is to estimate $\mu = \mathbb{E}[\psi]$; this, along with the class balance $P(\mathbf{Y})$, is sufficient information to compute $P_\mu(\mathbf{Y}|\boldsymbol{\lambda})$. If had access to a large enough set of ground truth labels $\mathbf{Y}$, we could simply take the empirical expectation $\hat{\mathbb{E}}[\psi]$; however in our setting we cannot directly observe this. Instead, we proceed by analyzing the covariance matrix, which corresponds to the *generalized covariance matrix* of our graphical model as in [24] (the exact connection is explained in the following):

$$\mathbf{Cov}[\psi] \equiv \Sigma = \mathbb{E}\left[\psi\psi^T\right] - \mu\mu^T. \tag{11}$$

[2] As we note later, we actually only need to account for a subset of $\mathcal{C}$, which consists of the maximal and separator set cliques in the junction tree representation of G_{source}.

Leveraging Observable Pairwise Products We start by noting that given our definition of ψ, we have:

$$\begin{aligned}
\mathbb{E}\left[\psi(C, y_C)\psi(D, y_D)\right] &= \mathbb{E}\left[\psi(C, y_C)\psi(D, y_D)^T\right] \\
&= \mathbb{E}\left[\left(\sum_{y' \in \mathcal{Y}} \mathbb{1}\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i, \mathbf{Y} = y'\right\}\right)\left(\sum_{y'' \in \mathcal{Y}} \mathbb{1}\left\{\cap_{j \in D} \boldsymbol{\lambda}_j = y_j, \mathbf{Y} = y''\right\}\right)\right] \\
&= \mathbb{E}\left[\sum_{y' \in \mathcal{Y}} \mathbb{1}\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i, \cap_{j \in D} \boldsymbol{\lambda}_j = y_j, \mathbf{Y} = y'\right\}\right] \\
&= \sum_{y' \in \mathcal{Y}} P\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i, \cap_{j \in D} \boldsymbol{\lambda}_j = y_j, \mathbf{Y} = y'\right\} \\
&= P\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i, \cap_{j \in D} \boldsymbol{\lambda}_j = y_j\right\},
\end{aligned}$$

where the last step follows from the law of total probability. And similarly, we have:

$$\begin{aligned}
\mathbb{E}\left[\psi(C, y_C)\psi_{\mathbf{Y}}\right] &= \mathbb{E}\left[\psi(C, y_C)\psi_{\mathbf{Y}}^T\right] \\
&= \mathbb{E}\left[\left(\sum_{y' \in \mathcal{Y}} \mathbb{1}\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i, \mathbf{Y} = y'\right\}\right)\left(\sum_{y'' \in \mathcal{Y}} \mathbb{1}\left\{\mathbf{Y} = y''\right\}\right)\right] \\
&= \mathbb{E}\left[\sum_{y' \in \mathcal{Y}} \mathbb{1}\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i, \mathbf{Y} = y'\right\}\right] \\
&= \sum_{y' \in \mathcal{Y}} P\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i, \mathbf{Y} = y'\right\} \\
&= P\left\{\cap_{i \in C} \boldsymbol{\lambda}_i = y_i\right\}.
\end{aligned}$$

In addition, we have that $\mathbb{E}\left[\phi_{\mathbf{Y}}\phi_{\mathbf{Y}}\right] = 1$. Now we see that $\mathbb{E}\left[\psi\psi^T\right]$ *does not depend* on the unobserved true label $\mathbf{Y}$, and can therefore be directly observed. This critical idea undergirds the rest of our analysis. We call this observable matrix the *overlaps matrix* $O = \mathbb{E}\left[\psi\psi^T\right]$, and note that it can be empirically estimated simply by counting the pairwise occurences of different source values. Thus we now have:

$$\Sigma = O - \mu\mu^T, \tag{12}$$

where O is empirically observable, μ is the vector of parameters we need to estimate, and Σ is the unknown covariance.

Sparsity Structure of the Inverse Covariance Matrix We do not know Σ, but we can apply Theorem 1 from Loh & Wainwright [24] to exploit our knowledge of G_{source} to get the sparsity structure of the *inverse* covariance matrix Σ^{-1}.

We do so through a two-step process that develops a series of graphs judiciously chosen for our approach. First, we consider the *base graph* G_{base}, a graphical model related to G_{source}. The vertices in G_{base} are $V(G_{\text{base}}) = \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_M, Y$. The edges are the same as those in G_{source}, with the additional set of edges given by $(\boldsymbol{\lambda}_i, Y)$ for all $1 \leq i \leq m$.

Now, let ψ_B be the augmented set of statistics for G_{base}, defined as in [24]. Effectively, $\psi_B(C)$ includes indicator variables for every combination of values that the variables in clique C take on but one (corresponding to a minimal exponential family model). The covariance matrix $\Sigma_B = \text{Cov}(\psi_B(C))$ is the generalized covariance matrix, whose inverse is block-structured according to Theorem 1 from Loh & Wainwright [24]. That is, a block corresponding to clique C_i and clique C_j in this inverse matrix is zero if the two are not part of the same maximal clique. For unary cliques (those cliques made up of single elements), this corresponds to a zero entry indicating conditional independence.

It is not convenient to directly work with the base graph covariance matrix Σ_B. Instead, we will form a second graph and corresponding covariance matrix that is easier to work with. This graph, called G_{hc}, only includes the statistics from ψ_B that include Y, so that G_{hc} is a subgraph of G_{base} and ψ_{hc} is a subvector of ψ_B. Since we are interested in the behavior of the sources with respect to Y, it is perhaps not surprising that G_{hc} is our graph of interest.

Now consider forming the covariance matrix Σ_{hc}; this matrix forms a block in the larger matrix Σ_B. Let us write the subblocks of Σ_B^{-1} as

$$\Sigma_B^{-1} = \begin{bmatrix} K_{\text{hc}} & R^T \\ R & K_{\text{lc}} \end{bmatrix}.$$

Here, these blocks of the base inverse covariance matrix corresponds to the terms including Y (left-upper corner), those not including Y (right-lower corner), and the cross terms. Using the block matrix inversion formula (that is, the Schur complement), we have that

$$\Sigma_{\text{hc}}^{-1} = K_{\text{hc}} - RK_{\text{lc}}^{-1}R^T. \tag{13}$$

That is, Σ_{hc}^{-1} can be expressed as the difference between the corresponding subblock of Σ_B^{-1} and the term $RK_{\text{lc}}^{-1}R^T$, which effectively expresses a mixing factor equivalent to conditioning out all of the variables in ψ_B that do not include Y. From here onwards, we shall rely on the following condition, which expresses that this mixing factor is not too large, so that

$$\|\Sigma_{\text{hc}}^{-1} - K_{\text{hc}}\|_\infty \leq \delta_B, \tag{14}$$

for some constant δ_B, and where the $\|\cdot\|_\infty$ indicates the largest absolute value of a matrix entry. Note that by [24], Σ_B^{-1} is graph-structured, and thus so is its subblock K_{hc}. Therefore, Σ_{hc}^{-1} is (nearly) graph-structured.

The condition (14) has a simple interpretation. It is equivalent to stating that marginalizing over the components of the source random variables that are independent of Y does not have too large of an effect on those components that are dependent of Y. We call this condition the *source-label correlation* assumption.

We have one further step to take. Note that ψ_{hc} contains all of the terms in our original vector formulation ψ, but these are unrolled vertically. In particular, we can write

$$\Sigma = A\Sigma_{\text{hc}}A^T,$$

where A is a matrix containing 1's that sums up the corresponding rows. Here, we rely on a second condition:

$$\|\Sigma^{-1} - (A^T)^+\Sigma_{\text{hc}}^{-1}A^+\|_\infty \leq \delta_A, \tag{15}$$

for some constant δ_A. Note that then we have that

$$\|\Sigma^{-1} - (A^T)^+K_{\text{hc}}A^+\|_\infty \leq \delta_A + k\delta_B. \tag{16}$$

Moreover, K_{hc} is graph-structured, so that Σ^{-1} is indeed also nearly graph-structured; we call the condition (16) the *marginal coherency* assumption for observability.

In the remainder of the Appendix, we will work with the noiseless assumptions $\delta_A = \delta_B = 0$. For an even more general setting where noise allows for positive but small δ_A and δ_B, we can easily adapt our bound in Theorem 2, which effectively adds a noise floor, but maintains the same convergence rate up to this noise floor. In other words, in this case, the constants act similar to the noise floor introduced by working with finite precision (i.e., having some positive machine epsilon), as we do in practice.

Transforming to a Low-Rank Approximation Problem in the Inverse Form Our strategy now is to leverage the fact that Σ^{-1} is approximately graph structured. We then begin by applying the Woodbury matrix identity to transform to the inverse form:

$$\begin{aligned}\Sigma^{-1} &= \left(O - \mu\mu^T\right)^{-1} \\ &= O^{-1} + O^{-1}\mu\left(I - \mu^T O^{-1}\mu\right)^{-1}\mu^T O^{-1} \\ &= O^{-1} + zz^T,\end{aligned}$$

where $z = O^{-1}\mu J$, and $JJ^T = \left(I - \mu^T O^{-1}\mu\right)^{-1}$.

We justify the decomposition $JJ^T = \left(I - \mu^T O^{-1}\mu\right)^{-1}$ as follows. Applying the Woodbury matrix identity again:

$$\left(I - \mu^T O^{-1}\mu\right)^{-1} = I + \mu^T\left(O - \mu\mu^T\right)^{-1}\mu = I + \mu^T\Sigma^{-1}\mu.$$

Algorithm 2 Class-Conditional Source Accuracy Estimation for Multi-Task Supervision

Input: Empirical overlaps matrix $\hat{O} \in \mathbb{R}^{|\mathcal{C}| \times |\mathcal{C}|}$, correlation sparsity structure Ω
$\hat{z} \leftarrow \text{argmin}_z \left|\left|\hat{O}^{-1} + zz^T\right|\right|_\Omega$
$\hat{Q} \leftarrow \hat{O}\hat{z}(I + \hat{z}^T\hat{O}\hat{z})^{-1}\hat{z}^T\hat{O}$
$\hat{\mu} \leftarrow \text{argmin}_\mu \left\{\left|\left|\hat{Q} - \mu\mu^T\right|\right|_F + \left|\left|\mu\vec{1} - \text{diag}(\hat{O})\right|\right|_F\right\}$
return $\hat{\mu}$

Here, we know that $\Sigma \succeq 0 \implies \Sigma^{-1} \succeq 0$, and therefore we can express $\Sigma^{-1} = BB^T$ for some matrix B. Thus, the second term is equal to $(\mu^T B)(\mu^T B)^T$, which is a Gram matrix and is thus PSD. Since the sum of two PSD matrices is PSD, therefore $\left(I - \mu^T O^{-1}\mu\right)^{-1} \succeq 0$ and can thus indeed be decomposed as a Gram matrix JJ^T for some J.

Now, let Ω be the *inverse augmented edge set* which contains all the pairs of indices of ψ, i.e. all pairs of nodes (and cliques) $A, B \in \mathcal{C}$, such that A and B are *not* part of the same maximal clique. Then, let A_Ω denote a matrix A with all entries $(i, j) \notin \Omega$ masked to zero. Then, using the known sparsity structure of Σ^{-1}, which is nearly zero for all entries in Ω, we have:

$$O_\Omega^{-1} + (zz^T)_\Omega = 0. \tag{17}$$

Note that in the noisy case, we instead have the bound $\|O_\Omega^{-1} + (zz^T)_\Omega\|_\infty \leq \delta_A + k\delta_B$ using our earlier conditions; this then leads to a *noisy* low-rank matrix completion problem, which can be tackled via robust matrix completion techniques [5; 21].

Thus, given the dependency graph G_{source}, we can solve for z as a *masked* low-rank matrix approximation problem (i.e., a matrix completion problem). Defining the semi-norm $||A||_\Omega = ||A_\Omega||_F$, we can solve:

$$\hat{z} = \text{argmin}_z \left|\left|O^{-1} + zz^T\right|\right|_\Omega. \tag{18}$$

Given an estimate $\hat{z}$, we can recover $\hat{\mu}$. Since $z = O^{-1}\mu J$,

$$\mu = OzJ^{-1}.$$

To estimate J, we note that

$$\begin{aligned} &J^{-T}z^T OzJ^{-1} = \mu^T O^{-1}\mu \\ \implies &(JJ^T)^{-1} = I - J^{-T}z^T OzJ^{-1} \\ \implies &J^T J = I + z^T Oz. \end{aligned}$$

So, we can decompose our empirically estimated $I + z^T Oz$ to recover an estimate of J and thus of $\hat{\mu}$.

Full Conditional-Parameter Estimation Algorithm Note that $J \in \mathbb{R}^{r \times r}$, so in the rank-one setting, $J^T J = c \in \mathbb{R}^+$, and $J = \sqrt{c}$. However, in the general setting of $r > 1$, we can only recover J up to symmetry, since $(UJ)^T(UJ) = J^T U^T UJ = J^T J$ for any orthogonal matrix U.

Instead, using the fact from above that $J^T J = I + z^T Oz$, note that we have:

$$Q \equiv \mu\mu^T = OzJ^{-1}J^{-T}z^T O = Oz(J^T J)^{-1}z^T O = Oz(I + z^T Oz)^{-1}z^T O.$$

Thus, given an estimate of z, we can get $Q = \mu\mu^T$ as well; we can then separately recover μ from Q using additional constraints more easily applied in this form. We use two basic constraints:

$$\begin{aligned} Q - \mu\mu^T &= 0 \\ \mu\vec{1} &= \text{diag}(O). \end{aligned} \tag{19}$$

Thus our full algorithm (Algorithm 2) now has two minimization stages, but both are still fast operations with minimal memory requirements, after forming $\hat{O}$.

A.3.3 Recovering the Class Balance P & Computing $P(Y|\lambda)$

The accuracies $\hat{\mu}$ we estimated above are joint accuracies; we wish to find the conditional accuracies. Thus we now turn to the task of recovering the class balance matrix $P \in [0,1]^{|\mathcal{Y}|\times|\mathcal{Y}|}$, where P is diagonal and $P_{\mathbf{Y},\mathbf{Y}} = P(\mathbf{Y})$. In many practical settings, P can be estimated from a small labeled sample, or may be known in advance. However here, we consider using a subset of conditionally independent sources, $s_1, \ldots, s_k$ to estimate P.

We note first of all that simply taking the majority vote of these sources is a biased estimator. Instead, we consider the corresponding rank-r unary mean statistics matrix μ and empirical overlaps matrix O as before, except here defined only over this subset of conditionally independent sources. Next, let A_i be the $|\mathcal{Y}_i \cup \{\vec{0}\}| \times |\mathcal{Y}|$ block of μ s.t.

$$(A_i)_{j,k} = P(\boldsymbol{\lambda}_i = y_j | \mathbf{Y} = y_k)$$

for $y_j, y_k \in \mathcal{Y}$. Finally, we can let $S = \sqrt{P}$ elementwise, and define $B_i = A_i S$. Then for any $i \neq j$, we have:

$$O_{B(i,j)} = B_i B_j^T, \tag{20}$$

where $O_{B(i,j)}$ is the corresponding block of the observations matrix O. We could then recover P by taking:

$$P = \text{diag}\left(B_i^T \vec{1}\right)^2,$$

since summing the column of B_i corresponding to label $\mathbf{Y}$ is equal to $\sqrt{P(\mathbf{Y})}\sum_{y\in\mathcal{Y}_i} P(\boldsymbol{\lambda}_i = y|\mathbf{Y}) = \sqrt{P(\mathbf{Y})}$ by the law of total probability. However, note that $B_i U$ for any orthogonal matrix U also satisfies (20), and could thus lead to a potentially infinite number of incorrect estimates of P.

Class Balance Identifiability with Three-Way View Constraint A different approach involves considering the three-way overlaps observed as $O_{B(i,j,k)}$. This is equivalent to performing a tensor decomposition. Note that above, the problem is that matrix decomposition is typically invariant to rotations and reflections; tensor decompositions have easier-to-meet uniqueness conditions (and are thus more rigid).

We apply Kruskal's classical identifiability condition for unique 3-tensor decomposition. Consider some tensor

$$T = \sum_{r=1}^{R} A_r \otimes B_r \otimes C_r,$$

where A_r, B_r, C_r are column vectors that make up the matrices A, B, C. The Kruskal rank k_A of A is the largest k such that any k columns of A are linearly independent. Then, the decomposition above is unique if $k_A + k_B + k_C \geq 2R + 2$ [22; 3]. In our case, our triple views have $R = |\mathcal{Y}|$, and we have

$$O_{B(i,j,k)} = B_i \otimes B_j \otimes B_k.$$

Thus, if $k_{B_i} + k_{B_j} + k_{B_k} \geq 2|\mathcal{Y}| + 2$, we have identifiability. Thus, it is sufficient to have the columns of each of the B_i's be linearly independent. Note that each of the B_i's have columns with the same sum, so these columns are only linearly dependent if they are equal, which would only be the case if the sources were random voters.

Recovering the Conditional Accuracies Given $P(\mathbf{Y})$, and our estimate of the joint parameters $(\mu_{C,y_C})_y = P(\boldsymbol{\lambda}_C = y_C, \mathbf{Y} = y)$ we can then get the conditional parameters:

$$(\alpha_{C,y_C})_y = P(\boldsymbol{\lambda}_C = y_C | \mathbf{Y} = y) = \frac{(\mu_{C,y_C})_y}{P(\mathbf{Y} = y)}. \tag{21}$$

Predicting Labels with the Label Model Once we have estimated the conditional accuracy parameters α, let $\theta = \log(\mu)$ be the corresponding vector of log parameters. Then we have a label model defined over the junction tree of G_{source} (consisting of maximal cliques $\bar{\mathcal{C}}$ and separator sets $\mathcal{S}$) as:

$$\begin{aligned} P_\mu(\boldsymbol{\lambda}, \mathbf{Y} = y) &= P(\mathbf{Y} = y) P_\mu(\boldsymbol{\lambda} | \mathbf{Y} = y) \\ &= P(\mathbf{Y} = y) \exp\left\{ \sum_{C\in\bar{\mathcal{C}}} (\theta_{C,y_C})_y^T \mathbb{1}\left\{\boldsymbol{\lambda}_C = y_C\right\} - \sum_{S\in\mathcal{S}} (\theta_{S,y_S})_y^T \mathbb{1}\left\{\boldsymbol{\lambda}_S = y_S\right\} \right\}, \end{aligned} \tag{22}$$

completing our task of producing the label model. The rest of the appendix is concerned with the technical conditions under which these operations can be performed.

A.3.4 Rank-One Form: Class-Symmetric Model

The formulation presented above is very general; it uses $r \times |\mathcal{Y}_{\tau_i}|$ different parameters to model the accuracy of each source, $P(\boldsymbol{\lambda}_i = y_i, \mathbf{Y} = y)$ for every $y_i \in \mathcal{Y}_{\tau_i}$, $y \in \mathcal{Y}$. We see above that this results in a masked rank-r constraint, which we use to solve for these parameters.

There are a variety of easier-to-handle special cases that exploit additional structure. For example, we might estimate a single accuracy parameter for each source on each task. Or, we might *tie* together accuracy parameters based on the structure of the task graph G_{task}; for an example of this, see Section A.4.

In the body, we instead consider a simplified model where we learn one parameter per source, which is the conditional accuracy $\alpha_i := P(\boldsymbol{\lambda}_i = (y)_{\tau_i} | \mathbf{Y} = y)$, and assume that this is the same for all $y \in \mathcal{Y}$. We additionally assume that the probability of emitting an incorrect label is uniform over the incorrect labels. We refer to this as the *class-symmetric* model. In this setting, we define the random variable *row vector* $\phi_i \in \{-1, 0, 1\}^{1 \times r}$ for each source s_i:

$$\phi_i = [\mathbb{1}\left\{\boldsymbol{\lambda}_i = (y_1)_{\tau_i}\right\} \mathbb{1}^{\pm}\left\{\mathbf{Y} = y_1\right\}, \ldots, \mathbb{1}\left\{\boldsymbol{\lambda}_i = (y_r)_{\tau_i}\right\} \mathbb{1}^{\pm}\left\{\mathbf{Y} = y_r\right\}] \tag{23}$$

for $\mathcal{Y} = \{y_1, \ldots, y_r\}$. Each element of ϕ_i is non-zero if s_i emits a corresponding label vector $\boldsymbol{\lambda}_i$, and positive or negative depending on whether s_i is correct.

We also define the product $\phi_i \phi_j = \phi_i \phi_j^T$. Since ϕ is a row vector, $\phi_i \phi_j$ is a scalar. Note that while ϕ_i is not observable due to its dependence on $\mathbf{Y}$, $\phi_i \phi_j$ is observable! In fact, this product acts as an indicator for when sources i, j agree:

$$\phi_i \phi_j = \phi_i \phi_j^T = \sum_{y \in \mathcal{Y}} \mathbb{1}\left\{\boldsymbol{\lambda}_i = (y)_{\tau_i}\right\} \mathbb{1}\left\{\boldsymbol{\lambda}_j = (y)_{\tau_j}\right\} \mathbb{1}^{\pm}\left\{\mathbf{Y} = y\right\} \mathbb{1}^{\pm}\left\{\mathbf{Y} = y\right\} = \mathbb{1}\left\{(\boldsymbol{\lambda}_i)_{\tau_i \cap \tau_j} = (\boldsymbol{\lambda}_j)_{\tau_i \cap \tau_j}\right\}.$$

And, as in the more general setting, we also define a corresponding random variable vector for $\mathbf{Y}$, $\phi_{\mathbf{Y}} = [\mathbb{1}^{\pm}\left\{\mathbf{Y} = y_1\right\}, \ldots, \mathbb{1}^{\pm}\left\{\mathbf{Y} = y_r\right\}]$, and note that $\phi_i \phi_{\mathbf{Y}}$ is also observable (though, of course, $\mathbf{Y}$ itself is not). In this setting, we then have:

$$\begin{aligned}
(\mathbb{E}\left[(\psi_C)\right])_s &= \mathbb{E}\left[\mathbb{1}\left\{\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}\right\} \mathbb{1}^{\pm}\left\{\mathbf{Y} = y_s\right\}\right] \\
&= \mathbb{E}\left[\mathbb{1}\left\{\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}, \mathbf{Y} = y_s\right\}\right] - \mathbb{E}\left[\mathbb{1}\left\{\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}, \mathbf{Y} \neq y_s\right\}\right] \\
&= P(\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}, \mathbf{Y} = y_s) - P(\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}, \mathbf{Y} \neq y_s) \\
&= 2P(\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}, \mathbf{Y} = y_s) - P(\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}) \\
&= 2\alpha_i P(\mathbf{Y} = y_s) - P(\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}).
\end{aligned}$$

We assume temporarily that we know the class balance $P(\mathbf{Y} = y)$ (for details on estimating it without labeled data, see Section A.3.3), and we can empirically observe the label frequency $P(\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\})$, so the above gives us an expression for α_i. Note also that we are only keeping track of statistics for when the members of a clique all agree on the correct label, e.g. $(\psi_C)_s = \mathbb{1}\left\{\cap_{i \in C}\left\{\boldsymbol{\lambda}_i = (y_s)_{\tau_i}\right\}\right\} \mathbb{1}^{\pm}\left\{\mathbf{Y} = y_s\right\}$. Given our class-symmetric assumptions, however, we can recover all of the clique marginals from $\mathbb{E}\left[\psi\right]$. For example,

$$\begin{aligned}
P(\boldsymbol{\lambda}_1 = y_1, \boldsymbol{\lambda}_2 = y_2, \mathbf{Y} = y_1) &= \frac{1}{r-1}\left(P(\boldsymbol{\lambda}_1 = y_1, \mathbf{Y} = y_1) - P(\boldsymbol{\lambda}_1 = y_1, \boldsymbol{\lambda}_2 = y_1, \mathbf{Y} = y_1)\right) \\
&= \frac{1}{r-1}\left((\mathbb{E}\left[\psi_1\right])_1 - (\mathbb{E}\left[\psi_{\{1,2\}}\right])_1\right).
\end{aligned}$$

Now, we have:

$$\begin{aligned}
&(\mathbb{E}\left[\psi\right]\mathbb{E}\left[\psi\right]^T)_{i,j} \\
&= \sum_{y \in \mathcal{Y}}\left(2\alpha_i P(\mathbf{Y} = y) - P(\boldsymbol{\lambda}_i = (y)_{\tau_i})\right)\left(2\alpha_j P(\mathbf{Y} = y) - P(\boldsymbol{\lambda}_j = (y)_{\tau_j})\right) \\
&= \sum_{y \in \mathcal{Y}}\Big(4P(\mathbf{Y} = y)^2 \alpha_i \alpha_j - 2P(\mathbf{Y} = y)P(\boldsymbol{\lambda}_j = (y)_{\tau_j})\alpha_i \\
&\qquad -2P(\mathbf{Y} = y)P(\boldsymbol{\lambda}_i = (y)_{\tau_i})\alpha_j + P(\boldsymbol{\lambda}_j = (y)_{\tau_j})P(\boldsymbol{\lambda}_i = (y)_{\tau_i})\Big) \\
&= 4\sigma_{\mathbf{Y}}\alpha_i\alpha_j - 2\rho_j\alpha_i - 2\rho_i\alpha_j + \rho_{i,j} \\
&= \left(2\sigma_{\mathbf{Y}}^{\frac{1}{2}}\alpha_i - \rho_i\sigma_{\mathbf{Y}}^{-\frac{1}{2}}\right)\left(2\sigma_{\mathbf{Y}}^{\frac{1}{2}}\alpha_j - \rho_j\sigma_{\mathbf{Y}}^{-\frac{1}{2}}\right) + \rho_{i,j} - \rho_i\rho_j\sigma_{\mathbf{Y}}^{-1} \\
&\equiv \mu_i\mu_j + \tilde{O}_{i,j},
\end{aligned}$$

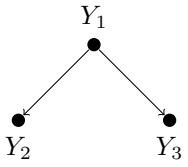

Figure 6: Example task hierarchy G_{task} for a three-task classification problem. Task Y_1 classifies a data point X as a PERSON or BUILDING. If Y_1 classifies X as a PERSON, Y_2 is used to distinguish between DOCTOR and NON-DOCTOR. Similarly, if Y_2 classifies X as a BUILDING, Y_3 is used to distinguish between HOSPITAL and NON-HOSPITAL. Tasks Y_2, Y_3 are more specific, or *finer-grained* tasks, constrained by their parent task Y_1.

where $\sigma_{\mathbf{Y}} = \sum_{y \in \mathcal{Y}} P(\mathbf{Y} = y)^2$, $\rho_i = \sum_{y \in \mathcal{Y}} P(\mathbf{Y} = y) P(\boldsymbol{\lambda}_i = (y)_{\tau_i})$, $\rho_{i,j} = \sum_{y \in \mathcal{Y}} P(\boldsymbol{\lambda}_i = (y)_{\tau_i}) P(\boldsymbol{\lambda}_j = (y)_{\tau_j})$, and $\mu \in \mathbb{R}^{(|\mathcal{C}|+1) \times 1}$. Thus we now have:

$$\Sigma = \mathbb{E}\left[\psi\psi^T\right] - \tilde{O} - \mu\mu^T \equiv O - \mu\mu^T.$$

Thus, we have reduced our main matrix approximation constraint to a rank-one problem given the class-symmetric model. We use this simplified model for our main theoretical results.

A.4 Example: Hierarchical Multi-Task Supervision

We now consider the specific case of *hierarchical* multi-task supervision, which can be thought of as consisting of coarser- and finer-grained labels, or alternatively higher- and lower-level labels, and provides a way to supervise e.g. fine-grained classification tasks at multiple levels of granularity. Specifically, consider a task label vector $\mathbf{Y} = [Y_1, \ldots, Y_t]^T$ as before, this time with $Y_s \in \{\texttt{N/A}, 1, \ldots, k_s\}$, where we will explain the meaning of the special value N/A shortly. We then assume that the tasks Y_s are related by a *task hierarchy* which is a hierarchy $G_{\text{task}} = (V, E)$ with vertex set $V = \{Y_1, Y_2, \ldots, Y_t\}$ and directed edge set E. The task structure reflects constraints imposed by higher level (more general) tasks on lower level (more specific) tasks. The following example illustrates a simple tree task structure:

Example 2 *Let Y_1 classify a data point X as either a PERSON ($Y_1 = 1$) or BUILDING ($Y_1 = 2$). If $Y_1 = 1$, indicating that X represents a PERSON, then Y_2 can further label X as a DOCTOR or NON-DOCTOR. Y_3 is used to distinguish between HOSPITAL and NON-HOSPITAL in the case that $Y_1 = 2$. The corresponding graph G_{task} is shown in Figure 6. If $Y_1 = 2$, then task Y_2 is not applicable, since Y_2 is only suitable for persons; in this case, Y_2 takes the value N/A. In this way the task hierarchy defines a feasible set of task vector values: $\mathbf{Y} = [1, 1, \texttt{N/A}]^T, [1, 2, \texttt{N/A}]^T, [2, \texttt{N/A}, 1]^T, [2, \texttt{N/A}, 2]^T$ are valid, while e.g. $\mathbf{Y} = [1, 1, 2]^T$ is not.*

As in the example, for certain configurations of $\mathbf{Y}$'s, the parent tasks logically constrain the one or more of the children tasks to be irrelevant, or rather, to have inapplicable label values. In this case, the task takes on the value N/A. In Example 2, we have that if $Y_1 = 1$, representing a building, then Y_2 is inactive (since X corresponds to a building). We define the symbol N/A (for incompatible) for this scenario, and define $\texttt{N/A} \times \texttt{N/A} = 1$ and $\texttt{N/A} \times s = -1$ for $s \neq \texttt{N/A}$. More concretely, let $\mathcal{N}(Y_i) = \{Y_j : (Y_j, Y_i) \in E\}$ be the in-neighborhood of Y_i. Then, the values of the members of $\mathcal{N}(Y_i)$ determine whether $Y_i = \texttt{N/A}$, i.e., $\mathbb{1}\{Y_j = \texttt{N/A}\}$ is deterministic conditioned on $\mathcal{N}(Y_i)$.

Hierarchical Multi-Task Sources Observe that in the mutually-exclusive task hierarchy just described, the value of a descendant task label Y_d determines the values of all other task labels in the hierarchy besides its descendants. For example, in Example 2, a label $Y_2 = 1 \implies (Y_1 = 1, Y_3 = \texttt{N/A})$; in other words, knowing that X is a DOCTOR also implies that X is a PERSON and not a BUILDING.

For a source $\boldsymbol{\lambda}_i$ with coverage set τ_i, the label it gives to the lowest task in the task hierarchy which is non-zero and non-N/A determines the entire label vector output by $\boldsymbol{\lambda}_i$. E.g. if the lowest task that $\boldsymbol{\lambda}_i$ labels in the hierarchy is $Y_1 = 1$, then this implies that it outputs vector $[1, 0, \texttt{N/A}]^T$. Thus, in this sense, we can think of each sources $\boldsymbol{\lambda}_i$ as labeling one specific task in the hierarchy, and thus can talk about coarser- and finer-grained sources.

Reduced-Rank Form: Modeling Local Accuracies In some cases, we can make slightly different modeling assumptions that reflect the nature of the task structure, and additionally result in a reduced rank form of our model. In particular, for the hierarchical setting introduced here, we can divide the conditional statistics $\mu_{\mathbf{Y}}$ into *local* and *global* subsets, and for example focus on modeling only the *local* ones to once again reduce to rank-one form.

To motivate with our running example: a finer-grained source that labels DOCTOR versus NON-DOCTOR probably is not accurate on the building type subtask; we can model this source using one accuracy parameter for the former label set (the *local* accuracy) and a different (or no parameter) for the *global* accuracy on irrelevant tasks. More specifically, for cliques involving $\boldsymbol{\lambda}_i$, we can model $\mu_{\mathbf{Y}}$ for all $\mathbf{Y}$ with only non-N/A values in the coverage set of $\boldsymbol{\lambda}_i$ using a single parameter, and call this the *local* accuracy; and we can either model $\mu_{\mathbf{Y}}$ for the other $\mathbf{Y}$ using one or more other parameters, or simply set it to a fixed value and not model it, to reduce to rank one form, as we do in the experiments. In particular, this allows us to capture our observation in practice that if a developer is writing a source to distinguish between labels at one sub-tree, they are probably not designing or testing it to be accurate on any of the other subtrees.

B Theoretical Results

In this section, we focus on theoretical results for the basic rank-one model considered in the main body of the paper. In Section B.1, we start by going through the conditions for identifiability in more detail for the rank-one case. In Section B.2, we provide additional interpretation for the expression of our primary theoretical result bounding the estimation error of the label model. In Section B.3, we then provide the proof of Theorem 1, connecting this estimation error to the generalization error of the end model; and in Section B.4, we provide the full proof of the main bound.

B.1 Conditions for Identifiability

Consider the rank one setting, where we have

$$-O_{\Omega}^{-1} = \left(zz^T\right)_{\Omega}, \tag{24}$$

where Ω is the inverse augmented edge set, i.e. a pair of indices (i, j), corresponding to elements of ψ, and therefore to cliques $A, B \in \mathcal{C}$, is in Ω if A, B are not part of the same maximal clique in G_{source} (and therefore $\Sigma_{i,j}^{-1} = 0$). This defines a set of $|\Omega|$ equations, which we can encode using a matrix M_Ω, where if (i, j) is the rth entry in Ω, then

$$(M_\Omega)_{r,s} = \begin{cases} 1 & s \in \{i, j\}, \\ 0 & \text{else.} \end{cases} \tag{25}$$

Let $l_i = \log(z_i^2)$ and $q_{(i,j)} = \log((O_{i,j}^{-1}))$; then by squaring and taking the log of both sides of 24, we get a system of linear equations:

$$M_\Omega l = q_\Omega. \tag{26}$$

Thus, we can identify z (and therefore μ) *up to sign* if M_Ω is invertible.

Notes on Invertibility of M_Ω Note that if the inverse augmented edge graph consists of a connected triangle (or any odd-numbered cycle), e.g. $\Omega = \{(i, j), (j, k), (i, k)\}$, then we can solve for the z_i up to sign, and therefore M_Ω must be invertible:

$$z_i^2 = \frac{O_{i,j}^{-1} O_{i,k}^{-1}}{O_{j,k}^{-1}},$$

and so on for z_j, z_k. Note additionally that if other z_i are connected to this triangle, then we can also solve for them up to sign as well. Therefore, if Ω contains at least one triangle (or odd-numbered cycle) per connected component, then M_Ω is invertible.

Also note that this is all in reference to the *inverse* source dependency graph, which will generally be dense (assuming the correlation structure between sources is generally sparse). For example, note that if we have one source $\boldsymbol{\lambda}_i$ that is conditionally independent of all the other sources, then Ω is fully connected, and therefore if there is a triangle in Ω, then M_Ω is invertible.

Identifying the Signs of the z_i Finally, note that if we know the sign of one z_i, then this determines the signs of every other z_j in the same connected component. Therefore, for z to be uniquely identifiable, we need only know the *sign* of one of the z_i in each connected component. As noted already, if even one source $\boldsymbol{\lambda}_i$ is conditionally independent of all the other sources, then Ω is fully connected; in this case, we can simply assume that the average source is better than random, and therefore identify the signs of z without any additional information.

Identifiability in the General Rank-r Setting In the general rank-r setting presented here, we solve a two-stage algorithm, first estimating $Q = \mu\mu^T$, and then recovering μ. We now use a result from [20] to provide a sufficient condition for identifiability in the rank-r setting, up to column permutations. First we clarify definitions. We have defined Ω as the set of indices (i, j) corresponding to rows in ψ which correspond to cliques $A, B \in \mathcal{C}$ such that A, B are not part of the same maximal clique in G_{source}. Now, we also define $\Omega_{\mathcal{C}}$, which is simply the set of (A, B) indexed directly. That is, if we had a graph of two independent sources s_1, s_2, with $r = |\mathcal{Y}| = 2$, we would have:

$$\begin{aligned}\Omega_{\mathcal{C}} &= \{(1,2)\},\\ \Omega &= \{(1,3),(1,4),(2,3),(2,4)\}.\end{aligned}$$

Now, we state the lemma:

Lemma 1 *Consider the bipartite graph $G(\Omega_{\mathcal{C}}) = (\mathcal{C}, \mathcal{C}, \Omega_{\mathcal{C}})$. Suppose that $G(\Omega_{\mathcal{C}})$ is connected, and that the columns of $\mu \in \mathbb{R}^{d\times r}$ are affinely independent. Then Algorithm 2 recovers μ up to column permutations.*

Proof: We first consider recovering the rank-r matrix ZZ^T from $O_{\Omega}^{-1} + (ZZ^T)_{\Omega} = 0$. We use Proposition 2.12 from Kiraly & Tomioka [20], which states that ZZ^T is recoverable using mask Ω if $G(\Omega)$ is r-closable. We start by using the fact that $G(\Omega_{\mathcal{C}})$ connected $\implies$ $G(\Omega_{\mathcal{C}})$ is 1-closable. This directly means that we can recursively form vertex sets of $G(\Omega_{\mathcal{C}})$ whose induced subgraphs are isomorphic to a complete 2×2 bipartite graph with one edge removed, until we have filled in all edges in $G(\Omega_{\mathcal{C}})$.

Now, consider the bipartite graph $G(\Omega) = (\{1, \ldots, d\}, \{1, \ldots, d\}, \Omega)$. Note that since we assume the conditional independence structure of our sources G_{source} is independent of the label being emitted, if $(A, B) \in \Omega_{\mathcal{C}}$ for cliques A, B, then $(i, j) \in \Omega$ for all r indices i corresponding to A and all r indices j corresponding to B. Thus, for every step of the recursive 1-closure procedure involving clique sets $\{A, A'\}, \{B, B'\}$, we can take r steps involving the corresponding indices to form the equivalent r-closure. Thus $G(\Omega)$ is r-closable, which implies that ZZ^T is recoverable.

Now, if we correctly estimate Z up to orthogonal transformations—i.e. if we estimate $\hat{Z} = ZU$, $U^TU = I$, then we can successfully recover $Q = \mu\mu^T$. From before we have:

$$\hat{Q} = O\hat{Z}(I + \hat{Z}^TO\hat{Z})^{-1}\hat{Z}^TO.$$

We now apply the Woodbury matrix identity to get:

$$\hat{Q} = O\hat{Z}\hat{Z}^TO - O\hat{Z}\hat{Z}^T(O^{-1} + \hat{Z}\hat{Z}^T)^{-1}\hat{Z}\hat{Z}^TO = OZZ^TO - OZZ^T(O^{-1} + ZZ^T)^{-1}ZZ^TO = Q.$$

Then, given Q, we estimate $\hat{\mu}$ using the constraints (1) $Q = \mu\mu^T$ and (2) $\mu\vec{1} = \text{diag}(O)$ (the law of total probability). Given constraint (1), we can recover $\hat{\mu} = \mu U$ for some orthogonal matrix U. Constraint (2) gives us:

$$\mu(U\vec{1} - \vec{1}) = 0.$$

Thus, unless U is a permutation matrix, then this implies that the columns of μ are not affinely independent. Thus, if we estimate the correct ZZ^T, and the columns of μ are affinely independent as assumed, then we can recover μ up to column permutations.□

We see that this remaining column permutation symmetry is the generalization of our previous setting where we could recover μ up to sign, and is broken in the same ways (e.g. by assuming all the sources are non-adversarial). Additionally, we can extend the above lemma to handle several disconnected components of $G(\Omega_{\mathcal{C}})$ as in the rank one setting.

B.2 Interpreting the Main Bound

We re-state Theorem 2, which bounds the average error on the estimate of the label model parameters, providing more detail on and interpreting the terms of the bound.

Theorem 2 *Let $\hat{\mu}$ be an estimate of μ^* produced by Algorithm 2 run over n unlabeled data points. Let $a := \left(\frac{1}{|\mathcal{C}|} - \lambda_{\min}^{-1}(O)\right)^{-\frac{1}{2}}$ and $b := \frac{\|O^{-1}\|^2}{O_{\min}^{-1}}$. Then, we have:*

$$\mathbb{E}\left[||\hat{\mu} - \mu^*||\right] \leq |\mathcal{C}|^2 \sqrt{\frac{32\pi}{n}} \left[(3\sqrt{|\mathcal{C}|} a \lambda_{\min}^{-1}(O) + 1) \times \left(2\sqrt{2} ab \sigma_{\max}(M_\Omega^+) \left[\kappa(O) + \lambda_{\min}^{-1}(O)\right]\right)\right].$$

Influence of $\sigma_{\max}(M_\Omega^+)$ the largest singular value of the pseudoinverse M_Ω^+. Note that $\|M_\Omega^+\|^2 = (\lambda_{\min}(M_\Omega^T M_\Omega))^{-1}$. As we shall see below, $\lambda_{\min}(M_\Omega^T M_\Omega)$ measures a quantity related to the structure of the graph G_{inv}. The smaller this quantity, the more information we have about G_{inv}, and the easier it is to estimate the accuracies. The smallest value of $\|M_\Omega^+\|^2$ (corresponding to the largest value of the eigenvalue) is $\sim \frac{1}{\sqrt{m}}$; the square of this quantity in the bound reduces the m^2 cost of estimating the covariance matrix to m.

It is not hard to see that

$$M_\Omega^T M_\Omega = \text{diag}(\deg(G_{\text{inv}})) + \text{Adj}(G_{\text{inv}}).$$

Here, $\deg(G_{\text{inv}})$ are the degrees of the nodes in G_{inv} and $\text{Adj}(G_{\text{inv}})$ is its adjacency matrix. This form closely resembles the graph Laplacian, which differs in the sign of the adjacency matrix term: $\mathcal{L}(G) = \text{diag}(\deg(G)) - \text{Adj}(G)$. We bound

$$\sigma_{\max}(M_S^+) \leq \left(d_{\min} + \lambda_{\min}(\text{Adj}(G_{\text{inv}}))\right)^{-1},$$

where $d_{\min}$ is the lowest-degree node in G_{inv} (that is, the source s with fewest appearances in Ω). In general, computing $\lambda_{\min}(\text{Adj}(G_{\text{inv}}))$ can be challenging. A closely related task can be done via *Cheeger inequalities*, which state that

$$2h_G \geq \lambda_{\min}(\mathcal{L}(G)) \geq \frac{1}{2} h_G^2,$$

where $\lambda_{\min}(\mathcal{L}(G))$ is the smallest non-zero eigenvalue of $\mathcal{L}(G)$ and

$$h_G = \min_X \frac{|E(X, \bar{X})|}{\min\left\{\sum_{x \in X} d_x, \sum_{y \in \bar{X}} d_y\right\}}$$

is the *Cheeger constant* of the graph [7]. The utility of the Cheeger constant is that it measures the presence of a bottleneck in the graph; the presence of such a bottleneck limits the graph density and is thus beneficial when estimating the structure in our case. Our Cheeger-constant like term $\sigma_{\max}(M_\Omega^+)$ acts the same way.

Now, in the easiest and most common case is that of conditionally independent sources [9; 40; 9; 17]., $\text{Adj}(G_{\text{inv}})$ has 1's everywhere but the diagonal, and we can compute explicitly that

$$\sigma_{\max}(M_S^+) = \frac{1}{\sqrt{m-2}}.$$

In the general setting, we must compute the minimal eigenvalue of the adjacency matrix, which is tractable, for example, for tree structures.

Influence of $\lambda_{\min}(O)$ the smallest eigenvalue of the observed matrix. This quantity reflects the conditioning of the observed (correlation) matrix; the better conditioned the matrix, the easier it is to estimate O.

Influence of $O_{\min}^{-1}$ the smallest entry of the inverse observed matrix. This quantity contributes to Σ^{-1}, the precision matrix; it is a measure of the smallest non-zero correlation between source accuracies (that is, the smallest correlation between non-independent source accuracies). Note that the tail bound of Theorem 2 scales as $\exp(-(O_{\min}^{-1})^2)$. This is natural, as distinguishing between small correlations and independencies requires more samples.

B.3 Proof of Theorem 1

Let $\mathcal{D}$ be the true data generating distribution, such that $(X, \mathbf{Y}) \sim \mathcal{D}$. Let $P_\mu(\mathbf{Y}|\boldsymbol{\lambda})$ be the label model parameterized by μ and conditioned on the observed source labels $\boldsymbol{\lambda}$. Furthermore, assume that:

1. For some optimal label model parameters μ^*, $P_{\mu^*}(\boldsymbol{\lambda}, \mathbf{Y}) = P(\boldsymbol{\lambda}, \mathbf{Y})$;
2. The label $\mathbf{Y}$ is independent of the features of our end model given the source labels $\boldsymbol{\lambda}$

That is, we assume that (i) the *optimal* label model, parameterized by μ^*, correctly matches the true distribution of source labels $\boldsymbol{\lambda}$ drawn from the true distribution, $(s(X), \mathbf{Y}) \sim \mathcal{D}$; and (ii) that these labels $\boldsymbol{\lambda}$ provide sufficient information to discern the label $\mathbf{Y}$. We note that these assumptions are the ones used in prior work [30], and are intended primarily to illustrate the connection between the estimation accuracy of $\hat{\mu}$, which we bound in Theorem 2, and the end model performance.

Now, suppose that we have an end model parameterized by w, and that to learn these parameters we minimize a normalized bounded loss function $l(w, X, \mathbf{Y})$, such that without loss of generality, $l(w, X, \mathbf{Y}) \leq 1$. Normally our goal would be to find parameters that minimize the expected loss, which we denote w^*:

$$L(w) = \mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[l(w, X, \mathbf{Y})\right] \tag{27}$$

However, since we do not have access to the true labels $\mathbf{Y}$, we instead minimize the expected noise-aware loss, producing an estimate $\tilde{w}$:

$$L_\mu(w) = \mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[\mathbb{E}_{\tilde{\mathbf{Y}}\sim P_\mu(\cdot|\boldsymbol{\lambda}(X))}\left[l(w, X, \tilde{\mathbf{Y}})\right]\right]. \tag{28}$$

In practice, we actually minimize the *empirical* version of the noise aware loss over an unlabeled dataset $U = \{X^{(1)}, ..., X^{(n)}\}$, producing an estimate $\hat{w}$:

$$\hat{L}_\mu(w) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\tilde{\mathbf{Y}}\sim P_\mu(\cdot|\boldsymbol{\lambda}(X^{(i)}))}\left[l(w, X^{(i)}, \tilde{\mathbf{Y}})\right]. \tag{29}$$

Let w^* be the minimizer of the expected loss L, let $\tilde{w}$ be the minimizer of the noise-aware loss for estimated label model parameters μ, L_μ, and let $\hat{w}$ be the minimizer of the empirical noise aware loss $\hat{L}_\mu$. Our goal is to bound the *generalization risk*- the difference between the expected loss of our empirically estimated parameters and of the optimal parameters,

$$L(\hat{w}) - L(w^*). \tag{30}$$

Additionally, since analyzing the empirical risk minimization error is standard and not specific to our setting, we simply assume that the error $|L_\mu(\tilde{w}) - L_\mu(\hat{w})| \leq \gamma(n)$, where $\gamma(n)$ is a decreasing function of the number of unlabeled data points n.

To start, using the law of total expectation first, followed by our assumption (2) about condtional independence, and finally using our assumption (1) about our optimal label model μ^*, we have that:

$$\begin{aligned}
L(w) &= \mathbb{E}_{(X',\mathbf{Y}')\sim\mathcal{D}}\left[L(w)\right] \\
&= \mathbb{E}_{(X',\mathbf{Y}')\sim\mathcal{D}}\left[\mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[l(w, X', \mathbf{Y})|X = X'\right]\right] \\
&= \mathbb{E}_{(X',\mathbf{Y}')\sim\mathcal{D}}\left[\mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[l(w, X', \mathbf{Y})|s(X) = s(X')\right]\right] \\
&= \mathbb{E}_{(X',\mathbf{Y}')\sim\mathcal{D}}\left[\mathbb{E}_{(\boldsymbol{\lambda},\tilde{\mathbf{Y}})\sim\mu^*}\left[l(w, X', \tilde{\mathbf{Y}})|\boldsymbol{\lambda} = s(X')\right]\right] \\
&= L_{\mu^*}(w).
\end{aligned}$$

Now, we have:

$$\begin{aligned}
L(\hat{w}) - L(w^*) &= L_{\mu^*}(\hat{w}) + L_\mu(\hat{w}) - L_\mu(\hat{w}) + L_\mu(\tilde{w}) - L_\mu(\tilde{w}) - L_{\mu^*}(w^*) \\
&\leq L_{\mu^*}(\hat{w}) + L_\mu(\hat{w}) - L_\mu(\hat{w}) + L_\mu(w^*) - L_\mu(\tilde{w}) - L_{\mu^*}(w^*) \\
&\leq |L_\mu(\hat{w}) - L_\mu(\tilde{w})| + |L_{\mu^*}(\hat{w}) - L_\mu(\hat{w})| + |L_\mu(w^*) - L_{\mu^*}(w^*)| \\
&\leq \gamma(n) + 2\max_{w'}|L_{\mu^*}(w') - L_\mu(w')|,
\end{aligned}$$

where in the first step we use our result that $L = L_{\mu^*}$ as well as add and subtract terms; and in the second step we use the fact that $L_\mu(\tilde{w}) \leq L_\mu(w^*)$. We now have our generalization risk controlled primarily by $|L_{\mu^*}(w') - L_\mu(w')|$,

which is the difference between the expected noise aware losses given the estimated label model parameters μ and the true label model parameters μ^*. Next, we see that, for any w':

$$\begin{aligned}
|L_{\mu^*}(w') - L_{\mu}(w')| &= \left|\mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[\mathbb{E}_{\tilde{\mathbf{Y}}\sim P_{\mu^*}(\cdot|\boldsymbol{\lambda})}\left[l(w,X,\tilde{\mathbf{Y}})\right] - \mathbb{E}_{\tilde{\mathbf{Y}}\sim P_{\mu}(\cdot|\boldsymbol{\lambda})}\left[l(w,X,\tilde{\mathbf{Y}})\right]\right]\right| \\
&= \left|\mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[\sum_{\mathbf{Y}'\in\mathcal{Y}} l(w,X,\mathbf{Y}')\left(P_{\mu^*}(\mathbf{Y}'|\boldsymbol{\lambda}) - P_{\mu}(\mathbf{Y}'|\boldsymbol{\lambda})\right)\right]\right| \\
&\leq \sum_{\mathbf{Y}'\in\mathcal{Y}} \mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[\left|P_{\mu^*}(\mathbf{Y}'|\boldsymbol{\lambda}) - P_{\mu}(\mathbf{Y}'|\boldsymbol{\lambda})\right|\right] \\
&\leq |\mathcal{Y}| \max_{\mathbf{Y}'} \mathbb{E}_{(X,\mathbf{Y})\sim\mathcal{D}}\left[\left|P_{\mu^*}(\mathbf{Y}'|\boldsymbol{\lambda}) - P_{\mu}(\mathbf{Y}'|\boldsymbol{\lambda})\right|\right],
\end{aligned}$$

where we have now bounded $|L_{\mu^*}(w') - L_{\mu}(w')|$ by the size of the structured output space $|\mathcal{Y}|$, and a term having to do with the difference between the probability distributions of μ and μ^*.

Now, we use the result from [16] (Lemma 19) which establishes that the log probabilities of discrete factor graphs with indicator features (such as our model $P_\mu(\boldsymbol{\lambda}, \mathbf{Y})$) are $(l_\infty, 2)$-Lipschitz with respect to their parameters, and the fact that for x, y s.t. $|x|, |y| \leq 1$, $|x - y| \leq |\log(x) - \log(y)|$, to get:

$$\begin{aligned}
\left|P_{\mu^*}(\mathbf{Y}'|\boldsymbol{\lambda}) - P_{\mu}(\mathbf{Y}'|\boldsymbol{\lambda})\right| &\leq \left|\log P_{\mu^*}(\mathbf{Y}'|\boldsymbol{\lambda}) - \log P_{\mu}(\mathbf{Y}'|\boldsymbol{\lambda})\right| \\
&\leq \left|\log P_{\mu^*}(\boldsymbol{\lambda}, \mathbf{Y}') - \log P_{\mu}(\boldsymbol{\lambda}, \mathbf{Y}')\right| + \left|\log P_{\mu^*}(\boldsymbol{\lambda}) - \log P_{\mu}(\boldsymbol{\lambda})\right| \\
&\leq 2\left|\left|\mu^* - \mu\right|\right|_\infty + 2\left|\left|\mu^* - \mu\right|\right|_\infty \\
&\leq 4\left|\left|\mu^* - \mu\right|\right|,
\end{aligned}$$

where we use the fact that the statement of Lemma 19 also holds for every marginal distribution as well. Therefore, we finally have:

$$L(\hat{w}) - L(w^*) \leq \gamma(n) + 4|\mathcal{Y}|\left|\left|\mu^* - \mu\right|\right|.$$

B.4 Proof of Theorem 2

Proof: First we briefly provide a roadmap of the proof of Theorem 2. We estimate $\tilde{\mu}$ with our procedure, and we seek a tail bound on $\|\tilde{\mu} - \mu\|$. The challenge here is that the observed matrix O we see is itself the mean of a series of samples $O_1, O_2, \ldots, O_n$. We bound (through a matrix concentration inequality) the error $\Delta_O = \tilde{O} - O$, and view Δ_O is a perturbation of O. Afterwards, we use a series of perturbation analyses to ultimately bound $\|\tilde{\mu} - \mu\|$; each of the perturbation results is in terms of Δ_O.

We begin with some notation. We write the following perturbations (note that all the terms written with Δ are additive, while the δ term is relative)

$$\begin{aligned}
\tilde{\mu} &= \mu + \Delta_\mu, \\
\tilde{O} &= O + \Delta_O, \\
\tilde{\ell} &= \ell + \Delta_\ell, \\
\tilde{z} &= (I + \text{diag}(\delta_z))z.
\end{aligned}$$

Now we start our perturbation analysis:

$$\begin{aligned}
\tilde{\mu} = \frac{1}{\sqrt{\tilde{c}}}\tilde{O}\tilde{z} &= \frac{1}{\sqrt{\tilde{c}}}(O + \Delta_O)(I + \text{diag}(\delta_z))z \\
&= \frac{1}{\sqrt{\tilde{c}}}\left(Oz + O\text{diag}(\delta_z)z + \Delta_O(I + \text{diag}(\delta_z))z\right).
\end{aligned}$$

Subtracting $\mu = \frac{1}{\sqrt{c}}Oz$, we get

$$\Delta_\mu = \left(\frac{1}{\sqrt{\tilde{c}}} - \frac{1}{\sqrt{c}}\right)Oz + \frac{1}{\sqrt{\tilde{c}}}\left(O\text{diag}(\delta_z)z + \Delta_O(I + \text{diag}(\delta_z))z\right). \tag{31}$$

The rest of the analysis requires us to bound the norms for each of these terms.

Left-most term. We have that

$$\left\|\left(\frac{1}{\sqrt{\tilde{c}}}-\frac{1}{\sqrt{c}}\right)Oz\right\| = \left|\frac{\sqrt{c}}{\sqrt{\tilde{c}}}-1\right|\left\|\frac{1}{\sqrt{c}}Oz\right\| = \left|\frac{\sqrt{c}}{\sqrt{\tilde{c}}}-1\right|\|\mu\| \leq \sqrt{m}\left|\frac{\sqrt{c}}{\sqrt{\tilde{c}}}-1\right| \leq \sqrt{m}|\tilde{c}-c|.$$

Here, we bounded $\|\mu\|$ by $\sqrt{m}$, since $\mu_i \leq 1$ for $1 \leq i \leq m$. In the last inequality, we used the fact that $c, \tilde{c} > 1$, so that $|\sqrt{c}/\sqrt{\tilde{c}} - 1| \leq |\sqrt{c} - \sqrt{\tilde{c}}| \leq |\tilde{c} - c|$.

Next we work on bounding $|\tilde{c} - c|$. We have

$$\begin{aligned}
|\tilde{c}-c| &= |\tilde{z}^T\tilde{O}\tilde{z} - z^TOz| \\
&= |z^T(I+\text{diag}(\delta_z))^T(O+\Delta_O)(I+\text{diag}(\delta_z))z - z^TOz| \\
&= |z^TO\text{diag}(\delta_z)z + z^T\Delta_O(I+\text{diag}(\delta_z))z + z^T\text{diag}(\delta_z)^T(O+\Delta_O)(I+\text{diag}(\delta_z))z| \\
&\leq \|z\|^2(\|O\|\|\delta_z\| + \|\Delta_O\|(1+\|\delta_z\|) + (\|\delta_z\|^2 + \|\delta_z\|)(\|O\| + \|\Delta_O\|)).
\end{aligned}$$

Thus,

$$\left\|\left(\frac{1}{\sqrt{\tilde{c}}}-\frac{1}{\sqrt{c}}\right)Oz\right\| \leq \sqrt{m}(\|z\|^2(\|O\|\|\delta_z\| + \|\Delta_O\|(1+\|\delta_z\|) + (\|\delta_z\|^2 + \|\delta_z\|)(\|O\| + \|\Delta_O\|)). \tag{32}$$

Bounding c. We will need a bound on c to bound z. We have that

$$c = (1 - \mu^TO^{-1}\mu)^{-1}.$$

Next, $\mu^TO^{-1}\mu \leq \lambda_{\min}^{-1}(O)\|\mu\|^2$, so that

$$1 - \mu^TO^{-1}\mu \geq 1 - \lambda_{\min}^{-1}(O)\|\mu\|^2.$$

Then,

$$c \leq \left(1 - \lambda_{\min}^{-1}(O)\|\mu\|^2\right)^{-1}.$$

Bounding z. We'll use our bound on c, since $z = \sqrt{c}O^{-1}\mu$.

$$\begin{aligned}
\|z\| &= \|\sqrt{c}O^{-1}\mu\| \\
&\leq \left(1 - \lambda_{\min}^{-1}(O)\|\mu\|^2\right)^{-\frac{1}{2}}\lambda_{\min}^{-1}(O)\|u\| \\
&= \frac{\lambda_{\min}^{-1}(O)\|u\|}{\left(1 - \lambda_{\min}^{-1}(O)\|\mu\|^2\right)^{\frac{1}{2}}} \\
&= \frac{\lambda_{\min}^{-1}(O)}{\left(\frac{1}{\|\mu\|^2} - \lambda_{\min}^{-1}(O)\right)^{\frac{1}{2}}} \\
&\leq \frac{\lambda_{\min}^{-1}(O)}{\left(m^{-1} - \lambda_{\min}^{-1}(O)\right)^{\frac{1}{2}}}.
\end{aligned}$$

In the last inequality, we used the fact that $\|u\|^2 \leq m$. Now we want to control $\|\Delta_\ell\|$.

Perturbation bound. We have the perturbation bound

$$\|\Delta_\ell\| \leq \|M_S^+\|\|\tilde{q}_S - q_S\|. \tag{33}$$

We need to work on the term $\|\tilde{q}_S - q_S\|$. To avoid overly heavy notation, we write $P = O^{-1}$, $\tilde{P} = \tilde{O}^{-1}$, and $\Delta_P = P - \tilde{P}$.

$$\begin{aligned}
\|\tilde{q}_S - q_S\|^2 &= \sum_{(i,j)\in S}\left(\log(\tilde{P}_{i,j}^2) - \log(P_{i,j}^2)\right)^2 \\
&= 4\sum_{(i,j)\in S}\left(\log(|\tilde{P}_{i,j}|) - \log(|P_{i,j}|)\right)^2 \\
&= 4\sum_{(i,j)\in S}(\log(|P_{i,j} + (\Delta_P)_{i,j}|) - \log(|P_{i,j}|))^2
\end{aligned}$$

$$
\begin{aligned}
&= 4 \sum_{(i,j)\in S} \left(\log(|P_{i,j} + (\Delta_P)_{i,j}|) - \log(|P_{i,j}|)\right)^2 \\
&\leq 4 \sum_{(i,j)\in S} \left[\log\left(1 - \left|\frac{(\Delta_P)_{i,j}}{P_{i,j}}\right|\right)\right]^2 \\
&\leq 8 \sum_{(i,j)\in S} \frac{|(\Delta_P)_{i,j}|}{|P_{i,j}|} \\
&\leq \frac{8}{P_{\min}^2} \sum_{(i,j)\in S} (\Delta_P)_{i,j}^2 \\
&\leq \frac{8\|\tilde{O}^{-1} - O^{-1}\|^2}{(O_{\min}^{-1})^2}.
\end{aligned}
$$

Here, the second inequality uses $(\log(1+x))^2 \leq x^2$, and the fourth inequality sums over squared values. Next, we use the perturbation bound $\|\tilde{O}^{-1} - O^{-1}\| \leq \|O^{-1}\|^2\|\Delta_O\|$, so that we have

$$
\|\tilde{q}_S - q_S\| \leq \frac{\sqrt{8}\|O^{-1}\|^2\|\Delta_O\|}{O_{\min}^{-1}}.
$$

Then, plugging this into (33), we get that

$$
\|\Delta_\ell\| \leq \frac{\sqrt{8}\|O^{-1}\|^2\|\Delta_O\|}{O_{\min}^{-1}} \sigma_{\max}(M_S^+). \tag{34}
$$

Bounding δ_z. Note also that $\|\Delta_\ell\|^2 = \sum_{i=1}^m (\log(\tilde{z}_i^2) - \log(z_i^2))$. We have that

$$
\begin{aligned}
\|\Delta_\ell\|^2 &= \sum_{i=1}^m \log\left(\frac{\tilde{z}_i^2}{z_i^2}\right) \\
&= 2\sum_{i=1}^m \log\left(\frac{|\tilde{z}_i|}{|z_i|}\right) \\
&= 2\sum_{i=1}^m \log(1 + |(\delta_z)_i|), \\
&\geq 2\sum_{i=1}^m (\delta_z)_i^2 \\
&= 2\|\delta_z\|^2,
\end{aligned}
$$

where in the fourth step, we used the bound $\log(1+a) \geq a^2$ for small a. Then, we have

$$
\|\delta_z\| \leq \frac{\sqrt{2}\|O^{-1}\|^2\|\Delta_O\|}{O_{\min}^{-1}} \sigma_{\max}(M_S^+). \tag{35}
$$

Putting it together. Using (31), we have that

$$
\begin{aligned}
\|\delta_\mu\| &= \left\| \left(\frac{1}{\sqrt{\tilde{c}}} - \frac{1}{\sqrt{c}}\right) Oz + \frac{1}{\sqrt{\tilde{c}}} \left(O\text{diag}(\delta_z)z + \Delta_O(I + \text{diag}(\delta_z))z\right) \right\| \\
&\leq \left\| \left(\frac{1}{\sqrt{\tilde{c}}} - \frac{1}{\sqrt{c}}\right) Oz \right\| + \left(\|O\text{diag}(\delta_z)\| + \|\Delta_O(I + \text{diag}(\delta_z))\|\right)\|z\| \\
&\leq \sqrt{m}(\|z\|^2(\|O\|\|\delta_z\| + \|\Delta_O\|(1 + \|\delta_z\|) + (\|\delta_z\|^2 + \|\delta_z\|)(\|O\| + \|\Delta_O\|)) \\
&\qquad + \|O\|\|\delta_z\|\|z\| + \|\Delta_O\|\|z\|(1 + \|\delta_Z\|) \\
&\leq \sqrt{m}(\|z\|^2(\|O\|\|\delta_z\| + \|\Delta_O\|(1 + \|\delta_z\|) + 2\|\delta_z\|(\|O\| + \|\Delta_O\|)) \\
&\qquad + \|O\|\|\delta_z\|\|z\| + \|\Delta_O\|\|z\|(1 + \|\delta_Z\|).
\end{aligned}
$$

In the third inequality, we relied on the fact that we can control $\|\delta_z\|$ (through $\|\Delta_O\|$) so that we can make it small enough and thus take $\|\delta_z\|^2 \leq \|\delta_z\|$. A little bit of rearrangement and algebra shows that

$$
\|\delta_\mu\| \leq (3\sqrt{m}\|z\| + 1)(\|z\|\|O\|\|\delta_z\| + \|z\|\|\delta_z\|\|\Delta_O\| + \|z\|\|\Delta_O\|).
$$

Now we can plug in our bounds from before. For convenience, we set $\|\Delta_O\| = t$. Recall that

$$a = (m^{-1} - \lambda_{\min}^{-1}(O))^{1/2}$$

and

$$b = \frac{\|O^{-1}\|^2}{O_{\min}^{-1}}.$$

Then, we have

$$\|\delta_\mu\| \leq (3\sqrt{m}a\lambda_{\min}^{-1}(O) + 1)\left(\sqrt{2}ab\kappa(O)\sigma_{\max}(M_S^+)t + \sqrt{2}ab\frac{\sigma_{\max}(M_S^+)}{\lambda_{\min}(O)}t^2 + a\lambda_{\min}^{-1}(O)t\right).$$

Again we can take t small so that $t^2 \leq t$. Simplifying further, we have

$$\|\delta_\mu\| \leq (3\sqrt{m}a\lambda_{\min}^{-1}(O) + 1)\left(\sqrt{2}ab\sigma_{\max}(M_S^+)\left[\kappa(O) + \lambda_{\min}^{-1}(O)\right] + a\lambda_{\min}^{-1}(O)\right)t.$$

Finally, since the $a\lambda_{\min}^{-1}(O)$ is smaller than the left-hand term inside the parentheses, we can write

$$\|\delta_\mu\| \leq (3\sqrt{m}a\lambda_{\min}^{-1}(O) + 1)\left(2\sqrt{2}ab\sigma_{\max}(M_S^+)\left[\kappa(O) + \lambda_{\min}^{-1}(O)\right]\right)t. \tag{36}$$

Concentration bound. We need to bound $t = \|\Delta_O\|$, the error when estimating O from observations $O_1, ..., O_n$ over n unlabeled data points. We apply the matrix Hoeffding inequality [34].

Let $S_k = \frac{1}{n}(O_k - O)$, and thus clearly $\mathbb{E}[S_k] = 0$. We seek a sequence of symmetric matrices A_k s.t. $S_k^2 \preceq A_k^2$. First, note that, for some vectors x, v,

$$x^T\left(||v||^2 I - vv^T\right)x = ||v||^2 ||x||^2 - \langle x, v\rangle^2 \geq 0$$

using Cauchy-Schwarz; therefore $||v||^2 I \succeq vv^T$, so that

$$m^2 I \succeq ||X_k||^4 I \succeq ||X_k||^2 X_k X_k^T = O_k^2.$$

Next, note that $(O_k + O)^2 \succeq 0$. Now, we use this to see that:

$$(nS_k)^2 = (O_k - O)^2 \preceq (O_k - O)^2 + (O_k + O)^2 = 2(O_k^2 + O^2) \preceq 2(m^2 I + O^2).$$

Therefore, let $A_k^2 = \frac{2}{n^2}(m^2 I + O^2)$, and note that $\left|\left|O^2\right|\right| \leq ||O||^2 \leq (m\,||O||_{max})^2 = m^2$. We then have

$$\sigma^2 = \left|\left|\sum_{k=1}^n A_k^2\right|\right| \leq \frac{2}{n}\left(m^2 + \left|\left|O^2\right|\right|\right) \leq \frac{4m^2}{n}.$$

And thus,

$$P\left(\left|\left|\tilde{O}_n - O\right|\right| \geq \gamma\right) \leq 2m\exp\left(-\frac{n\gamma^2}{32m^2}\right). \tag{37}$$

For notational convenience, we abbreviate $\tilde{O}_n = \tilde{O}$. Thus we have that

$$P(\|\Delta_O\| \geq \gamma) = P(t \geq \gamma) \leq 2m\exp\left(-\frac{n\gamma^2}{32m^2}\right).$$

Final steps We use the bound on t in (36) and the concentration bound above to write

$$\begin{aligned} P(\|\Delta_\mu\| \geq t') &\leq P(Vt \geq t') \\ &= P\left(t \geq \frac{t'}{V}\right) \\ &\leq 2m\exp\left(-\frac{nt'^2}{32V^2m^2}\right), \end{aligned}$$

where $V = (3\sqrt{m}a\lambda_{\min}^{-1}(O) + 1)\left(2\sqrt{2}ab\sigma_{\max}(M_S^+)\left[\kappa(O) + \frac{1}{\lambda_{\min}(O)}\right]\right)$.

We only have one more step:

$$\begin{aligned}
\mathbb{E}\left[||\tilde{\mu}-\mu||\right] &= \int_0^\infty P(\|\tilde{\mu}-\mu)\| \geq \gamma) d\gamma \\
&\leq \int_0^\infty 2m \exp\left(-\frac{n}{32V^2m^2}\gamma^2\right) d\gamma \\
&= \frac{2m\sqrt{\pi}}{2\sqrt{\frac{n}{32V^2m^2}}} \\
&= m^2\sqrt{\frac{32\pi}{n}}V.
\end{aligned}$$

Here, we used the fact that $\int_0^\infty \exp(-a\gamma^2)d\gamma = \frac{\sqrt{\pi}}{2\sqrt{a}}$. □

C Experimental Details

C.1 Data Balancing and Label Model Training Procedure

For each application, rebalancing was applied via direct subsampling to the training set in the manner that was found to most improve development set micro-averaged accuracy. Specifically, we rebalance with respect to the median class for OpenI (i.e. removing examples from majority class such that none had more than the original median class), the minimum class for TACRED, and perform no rebalancing for OntoNotes. For generative model training, we use stochastic gradient descent with a step size, step number, and ℓ_2 penalty listed in Table 3 below. These parameters were found via 10-trial coarse random search, with all values determined via maximum micro-averaged accuracy evaluated on the development set.

	OntoNotes	TACRED	OpenI
Label Model Training			
Step Size	5e-3	1e-2	5e-4
ℓ_2 Regularization	1e-4	4e-4	1e-3
Step Number	50	25	50
End Model Architecture			
Embedding Initialization	PubMed	FastText EN	Random
Embedding Size	100	300	200
LSTM Hidden Size	150	250	150
LSTM Layers	1	2	1
Intermediate Layer Dimensions	200, 50	200, 50, 25	200, 50
End Model Training			
Learning Rate	1e-2	1e-3	1e-3
ℓ_2 Regularization	1e-4	1e-4	1e-3
Epochs	20	30	50
Dropout	0.25	0.25	0.1

Table 3: Model architecture and training parameter details.

C.2 End Model Training Procedure

Before training over multiple iterations to attain averaged results for reporting, a 10-trial random search over learning rate and ℓ_2 regularization with the Adam optimizer was performed for each application based on micro-averaged development set accuracy. Learning rate was decayed by an order of magnitude if no increases in training loss improvement or development set accuracy were observed for 10 epochs, and the learning rate was frozen during the first 5 epochs. Models are reported using early stopping, wherein the best performing model on the development set is eventually used for evaluation on the held-out test set, and maximum epoch number is set for each application at a point beyond which minimal additional decrease in training loss was observed.

C.3 Dataset Statistics

We give additional detail in here (see Table 4) on the different datasets used for the experimental portion of this work. All data in the development and test sets is labeled with ground truth, while data in the training set is treated as unlabeled. Each dataset has a particular advantage in our study. The OntoNotes set, for instance, contains a particularly large number of relevant data points (over 63k), which enables us to investigate empirical performance scaling with the number of unlabeled data points. Further, the richness of the TACRED dataset allowed for the creation of an 8-class, 7-sub-task hierarchical classification problem, which demonstrates the utility of being able to supervise at each of the three levels of task granularity. Finally, the OpenI dataset represents a real-world, non-benchmark problem drawn from the domain of medical triage, and domain expert input was directly leveraged to create the relevant supervision sources. The fact that these domain expert weak supervision sources naturally occurred at multiple levels of granularity, and that the could be easily integrated to train an effective end model, demonstrates the utility of the `MeTaL` framework in practical settings.

	# Train	# Dev	# Test	Tree Depth	# Tasks	# Sources/Task
OntoNotes (NER)	62,547	350	345	2	3	11
TACRED (RE)	9,090	350	2174	3	7	9
OpenI (Doc)	2,630	200	378	2	3	19

Table 4: Dataset split sizes and sub-task structure for the three fine-grained classification tasks on which we evaluate `MeTaL`.

C.4 Task Accuracies

For clarity, we present in Table **??** the individual task accuracies of both the learned `MeTaL` model and MV for each experiment. These accuracies are computed from the output of evaluating each model on the test set with ties broken randomly.

C.5 Ablation Study: Unipolar Correction and Joint Modeling

We perform an additional ablation to demonstrate the relative gains of modeling unipolar supervision sources and jointly modeling accuracies across multiple tasks with respect to the data programming (DP) baseline [29]. Results of this investigation are presented in Table 6. We observe an average improvement of 2.8 points using the unipolar correction (DP-UI), and an additional 1.3 points from joint modeling within `MeTaL`, resulting in an aggregate gain of 4.1 accuracy points over the data programming baseline.

	OntoNotes	TACRED	OpenI
Task 1			
MV	93.3	74.2	83.9
MeTaL	91.9	80.5	84.1
Task 2			
MV	73.3	46.2	77.8
MeTaL	75.6	65.9	83.7
Task 3			
MV	71.4	74.9	61.7
MeTaL	74.1	74.8	61.7
Task 4			
MV	-	34.4	-
MeTaL	-	60.2	-
Task 5			
MV	-	36.2	-
MeTaL	-	40.2	-
Task 6			
MV	-	56.3	-
MeTaL	-	49.9	-
Task 6			
MV	-	36.8	-
MeTaL	-	56.3	-

Table 5: Label model task accuracies for each task for for both our approach and majority vote (MeTaL/MV)

	OntoNotes (NER)	TACRED (RE)	OpenI (Doc)	Average
DP [30]	78.4 $\pm$ 1.2	49.0 $\pm$ 2.7	75.8 $\pm$ 0.9	67.7
DP-UI	81.0 $\pm$ 1.2	54.2 $\pm$ 2.6	76.4 $\pm$ 0.5	70.5
MeTaL	**82.2** $\pm$ 0.8	**56.7** $\pm$ 2.1	**76.6** $\pm$ 0.4	**71.8**

Table 6: **Effect of Unipolar Correction.** We compare the micro accuracy (avg. over 10 trials) with 95% confidence intervals of a model trained using data programming (DP), data program with a unipolar correction (DP-UI), and our approach (MeTaL).