Regularized Data Programming with Bayesian Priors

Anonymous ACL submission

Abstract

The cost of manual data labeling can be a significant obstacle in supervised learning. Data programming (DP) offers a weakly supervised solution for training dataset creation, wherein the outputs of user-defined programmatic labeling functions (LFs) are reconciled through unsupervised learning. However, DP can fail to outperform an unweighted majority vote in some scenarios, including low-data contexts. This work introduces a Bayesian extension of classical DP that mitigates failures of unsupervised learning by augmenting the DP objective with regularization terms. Regularized learning is achieved through maximum a posteriori estimation in the Bayesian model. Majority vote is proposed as a proxy signal for estimating prior parameters. Results suggest that regularized DP improves performance relative to maximum likelihood and majority voting, confers greater interpretability, and bolsters performance in low-data regimes.

1 Introduction

Data programming (DP) is a paradigm for training dataset creation wherein weakly supervised label generation is encoded by user-defined labeling functions (LFs). The weak supervision signals offered by LFs allow for inexpensive yet noisy label generation, with LFs ranging from simple heuristics to wrappers for pre-trained models. Ratner et al. (2017a) cast DP as a generative model that reconciles cheap, noisy, and conflicting LF outputs by treating groundtruth labels as latent variables.

The utility of DP for research and industrial applications (Bach et al., 2019) has encouraged multiple adaptations of this paradigm, including adversarial DP (Arachie and Huang, 2019), DP for continuous LF outputs (Chatterjee et al., 2020), interactive weak supervision (Boecking et al., 2021), and semi-supervised DP (Maheshwari et al., 2021). However, the DP generative model can fail to produce accurate labels for some tasks, including contexts where unlabeled training data are scarce. The present work seeks to mitigate this failure mode via a Bayesian extension, as Bayesian priors serve as strong regularizers under limited data. This method regularizes unsupervised learning by specifying user beliefs, resulting in higher quality labels.

1.1 Motivating Example: Biomedical Information Extraction

Clinicians need efficient access to biomedical research findings to support evidence-based decision-making. The resource intensity of manually structuring and summarizing these data motivates the application of DP to biomedical information extraction. DP has facilitated extraction of protein–protein interactions from full-texts (Mallory et al., 2015), curation of genome-wide association study and chemical reaction databases (Kuleshov et al., 2019; Mallory et al., 2020), and creation of a MRI training dataset that outperformed a hand-labeled counterpart (Fries et al., 2019). The present work is motivated by the potential for regularized DP to support biomedical literature curation.

1.2 Contributions

The present work introduces a method for regularizing DP that mitigates failure modes of classical DP to produce higher quality labels in low-data regimes. The proposed Bayesian extension enables users to specify prior beliefs over LF accuracies and unknown groundtruth labels by adapting the DP objective from maximum likelihood estimation (MLE) to maximum a posteriori estimation (MAP). This approach incorporates signals from majority vote (MV) over the training data to facilitate automated prior selection and improve inference. A novel LF dataset that simulates curation of a transcriptomics systematic review is released as a DP benchmark. Data and source code are available on GitHub (https://github.com/regularized-dp/regularized-data-programming/).

2 Methodology

The DP framework generates synthetic labels for a dataset $\{x_i, y_i\}_{i=1}^n$ whose true labels $y_i \in \{-1, 1\}$ are unknown. Synthetic labels $\hat{y}_i$ are derived from noisy LFs $\{\lambda_j\}_{j=1}^m$, where each LF λ_j is characterized by its coverage β_j (the probability that the LF votes rather than abstains) and its accuracy α_j (the probability that the LF votes correctly). All notation is described in Table 1.

Variable	*Meaning*
$(x, y) \in \mathcal{X} \times \{-1, 1\}$	Observations and unknown labels.
m, n	Total LFs, total observations.
$i \in [1..n]$	Index over x, y, and $\hat{y}$.
$j \in [1..m]$	Index over λ, α, and β.
Λ	A $n \times m$ LF output matrix.
$\lambda_i : \mathcal{X} \mapsto \{-1, 0, 1\}^m$	Outputs of every LF for a given x_i.
$\lambda_j : \mathcal{X} \mapsto \{-1, 0, 1\}^n$	Outputs of a given LF for every x_i.
$\lambda_{ij} \in \{-1, 0, 1\}$	Output of LF λ_j for x_i; 0 = abstain.
$\hat{y}$	Predicted label vector given Λ.
$\hat{y}_i$	Predicted label for x_i given λ_i.
$\alpha \in \mathbb{R}^m$	Vector of all LF accuracies.
$\beta \in \mathbb{R}^m$	Vector of all LF coverages.
(u_j, v_j)	Beta distribution parameters.
p	Bernoulli distribution parameter.

Table 1: Variables referenced in model definition.

2.1 Model Definition

DP infers the unknown labels $\hat{y}$ from LF outputs by specifying a probabilistic model $\mathbf{P}(\lambda_{ij}, \hat{y}_i \mid \alpha_j, \beta_j)$, in which $\hat{y}_i$ is a latent variable and $\mathbf{P}(\lambda_{ij} | \hat{y}_i, \alpha_j, \beta_j) =$

$$\begin{cases} 1 - \beta_j & \text{if } \lambda_{ij} = 0 \\ \alpha_j \beta_j & \text{if } \lambda_{ij} = \hat{y}_i \\ (1 - \alpha_j)\beta_j & \text{if } \lambda_{ij} = -\hat{y}_i. \end{cases} \tag{1}$$

where $\lambda_{ij} = 0$ denotes abstention by λ_j.

Bayesian formulation We introduce priors over y_i and α_j as $\mathbf{Ber}(y_i; p)$ and $\mathbf{Beta}(\alpha_j; u_j, v_j)$, respectively. The choice of a beta prior is motivated by its conjugacy with the Bernoulli likelihood of the DP model. This yields a Bayesian latent variable model of the form

$$\prod_{i=1}^n \mathbf{P}(\lambda_i, \alpha, \beta) = \prod_{i=1}^n \big[\mathbf{P}(\lambda_i, \hat{y}_i = 1, \alpha, \beta) + \mathbf{P}(\lambda_i, \hat{y}_i = -1, \alpha, \beta)\big] \tag{2}$$

such that $\mathbf{P}(\lambda_i, \hat{y}_i, \alpha, \beta)$

$$= \mathbf{P}(\lambda_i | \hat{y}_i, \alpha, \beta)\mathbf{P}(\hat{y}_i)\mathbf{P}(\alpha) \tag{3}$$

$$= \prod_{j=1}^m \mathbf{P}(\lambda_{ij} | \hat{y}_i, \alpha_j, \beta_j)\mathbf{P}(\hat{y}_i)\mathbf{P}(\alpha_j). \tag{4}$$

2.2 Learning and Inference

MAP estimation In classical DP (Ratner et al., 2017a), model parameters are learned via MLE: $\arg\max_{\alpha,\beta} \mathbf{P}(\Lambda | \alpha, \beta)$. The proposed method instead computes MAP estimates

$$\hat{\alpha}, \hat{\beta} = \arg\max_{\alpha,\beta} \mathbf{P}(\Lambda, \alpha, \beta). \tag{5}$$

In practice, $\hat{\alpha}$ is estimated through stochastic gradient descent (SGD) and $\hat{\beta}$ is the observed abstention rate on the training set.

Inference Observations are assigned the most likely label under the Bayesian model:

$$\hat{y} = \arg\max_{y} \mathbf{P}(\Lambda, y, \hat{\alpha}, \hat{\beta}). \tag{6}$$

Abstention tie-breaking enables the model to assign $\hat{y}_i = 0$ when neither label is more probable under the model, allowing for the possibility of incomplete prediction coverage in $\hat{y}$.

2.3 Prior Selection and Model Regularization

Regularized learning is motivated by the observation that MV can outperform MLE when training data are scarce. While priors can be informed by user beliefs, we propose strategies for automatically selecting priors over y and α that exploit $\hat{y}_{mv}$, the labels predicted by MV over the training set.

Priors over y A single parameter p is applied to all Bernoulli priors over y. Wherever $\hat{y}_{mv}$ votes, we set $p \geq 0.5$ such that sufficiently strong priors (p approaching 1) will force the model to recapitulate every vote in $\hat{y}_{mv}$. While abstentions in $\hat{y}_{mv}$ are assigned an uninformative prior, the model can be forced to abstain on instances where MV abstains. This option exploits the observation that abstention forcing can improve performance on datasets where MV outperforms MLE, as demonstrated experimentally. The value of p is tuned via grid search on the validation set, along with the boolean flag for forced abstention.

Priors over α Priors over α_j are beta distributions with parameters (u_j, v_j) whose means approximate LF accuracies. Prior strength is dictated by the magnitude of parameters (u_j, v_j) and is selected through grid search on validation data. Beta distribution means are selected by computing the accuracy of each λ_j with respect to $\hat{y}_{mv}$, treating $\hat{y}_{mv}$ as a proxy for groundtruth. We refer to models employing this heuristic as MAP_{mv}. While this obviates human expertise and is cheaper than random

search, this heuristic only performs well for select datasets. As an experimental control, a theoretical upper bound on prior quality was obtained by setting distribution means to α^*, the LF empirical accuracies with respect to the training set. These simulated optimal priors provide a benchmark for automated prior selection, termed MAP_{α^*}.

3 Experimental Design

Experimental objectives are to demonstrate impacts of regularization on 1) the added value of unsupervised learning relative to MV, 2) performance in low-data contexts, and 3) the ability to infer LF accuracies. All experiments used abstention tie-breaking to assess variation in $\hat{y}$ coverage. Datasets were randomly split into training (80%) and testing (20%) sets, with 10% of training data held out as a validation set. Learned baselines underwent grid search hyperparameter tuning with evaluation on the validation split of each dataset. Detailed hyperparameter search outcomes are in Table A1.

3.1 Baseline Models

Maximum likelihood model This PyTorch module implements the objective defined in Ratner et al. (2017a). The MAP model is implemented identically, with the addition of priors over α and y.

Majority vote model Unweighted MV offers a naive baseline against which to compare trained models. This model assigns observations to the class selected by the majority of LFs.

Snorkel labeling model The Snorkel project (https://www.snorkel.org/) (Ratner et al., 2017b) is a DP framework with demonstrated utility at industrial scale (Bach et al., 2019). This baseline serves as an example of state-of-the-art DP.

CAGE This baseline offers a non-Bayesian DP regularization method for continuous and discrete LFs, with *quality guides* representing user estimates of LF accuracy (Chatterjee et al., 2020). The CAGE likelihood for discrete LFs is employed, which differs from Ratner et al. (2017a) by introducing k parameters per LF (where k is the total levels of y). This method requires all LFs to vote only in one direction (1 or -1), precluding its use for the transcriptomics dataset.

Support vector machine A regularized linear support vector machine (SVM) with a hinge loss optimized through SGD serves as a supervised baseline. SVMs were implemented in scikit-learn using bag-of-words features and training labels.

3.2 Datasets

TubeSpam This dataset (Alberto et al., 2015) (training $n = 1407$; validation $n = 157$; testing $n = 392$) is applied to 10 LFs defined for these data in the Snorkel documentation,[1]. This study employs the original TubeSpam dataset ($n = 1961$) rather than the truncated version used by Snorkel ($n = 1836$). Majority class is 51.28% of data.

Spouse This spousal relation information extraction dataset (training $n = 3858$; validation $n = 553$; testing $n = 1101$) is applied to 4 LFs defined for these data in the Snorkel documentation.[2] Majority class is 92.64% of data.

RNA This manually labeled dataset consists of PubMed titles obtained for a systematic review on time-course transcriptomics (training $n = 1656$; validation $n = 184$; testing $n = 460$). Titles are labeled according to relevance, with a positive label indicating that a given title is pertinent to the review and a negative label indicating irrelevance. Three LFs were written for this study (Table A2). Majority class is 62.17% of data.

4 Results and Discussion

Experimental results highlight three benefits of regularization: 1) priors buoy performance in low-data contexts; 2) MAP meets or exceeds MV and MLE on full data; and 3) model parameters approximate true LF accuracies more effectively than MLE. Reported scores are computed by excluding abstentions, per the convention of Snorkel. Priors over α were derived from MV unless otherwise stated.

TubeSpam	*MAP*	*MLE*	*MV*	*CAGE*	*SVM*
n = 500	92.3 (0.7)	91.8 (0.5)	-	75.3 (0.0)	94.4 (0.6)
n = 1407	92.6	91.5	91.4	75.3	95.1
Spouse	*MAP*	*MLE*	*MV*	*CAGE*	*SVM*
n = 500	48.9 (0.0)	48.7 (0.4)	-	44.6 (0.0)	12.2 (8.6)
n = 3858	48.9	48.9	48.9	44.6	24.6
RNA	*MAP*	*MLE*	*MV*	*CAGE*	*SVM*
n = 500	87.3 (0.0)	87.3 (0.0)	-	-	80.9 (3.3)
n = 1656	87.3	87.3	87.3	-	85.0

Table 2: F1 scores on test instances for which all models vote (75.3% for TubeSpam; 18.8% for Spouse; 77.2% for RNA) when trained on 500 and all training instances. Means (standard deviations) are over five replicates. See Tables A3 and A4 for additional metrics.

[1] https://github.com/snorkel-team/snorkel-tutorials/
[2] Ibid.

4.1 MAP Aids Performance on Limited Data

MAP was hypothesized to confer performance gains when training instances are limited. Training sets were randomly subsetted to simulate a range of training data availability ($n = \{1, 5, 10, 50, 100, 500, 1K, 2K, \text{full size}\}$). Performance on the full test set was averaged over five training replicates. MV priors were computed per subset. MAP provided superior performance on all metrics when training data were restricted, while SVM suffered most under limited data (Figs. 1, A2). MLE generally demonstrated higher variance on small training sets, though this instability decreased when evaluating on test instances for which all models vote (Table 2). As training set size increases, MLE approaches MAP performance. This is theoretically supported, as MAP and MLE are known to converge as training set size trends to infinity.

4.2 MAP Meets or Exceeds MLE on Full Data

A fundamental shortcoming of MLE for DP is its potential failure to outperform MV even on full data, as seen with TubeSpam (Tables 2, A3, A4). Behavior on TubeSpam can be explained by accuracy on instances where the model and MV agree versus disagree (Table A5). Poor accuracy on the latter showcases the added value of priors over y and forced abstention. As previously observed (Chatterjee et al., 2020), MLE experiences abrupt performance decay, while regularization facilitates a robust training process that is less sensitive to training epoch count (Fig. A3). MLE was more sensitive to hyperparameter values than MAP for TubeSpam and RNA. Conversely, MAP defers to the wisdom of both MV and MLE, ensuring that MAP meets or exceeds the performance of both on all tasks. Unlike MAP, Snorkel and CAGE did not outperform MV on any dataset (Tables 2, A3, A4).

4.3 MAP Learns Unknown LF Accuracies

MAP was hypothesized to provide greater fidelity of $\hat{\alpha}$ to true LF accuracies α^*. Here model quality is conceptualized as convergence of $\hat{\alpha}$ to α^*, measured by the ℓ^2-norm $||\alpha^* - \hat{\alpha}||_2$. A norm of zero indicates that α^* was learned exactly. We also measure this norm with respect to model priors to demonstrate the contribution of training. MAP_{mv}, MAP_{α^*}, and MAP_{rn} (random priors) simulated a range of prior quality. MAP_{mv} achieved greater convergence than MLE on two of three datasets, while MAP_{α^*} achieved greatest convergence on all datasets (Table A6). This suggests that MAP with well-selected priors learns LF accuracies better than MLE. In contrast, CAGE quality guides failed to provide a natural interpretation (Fig. A4).

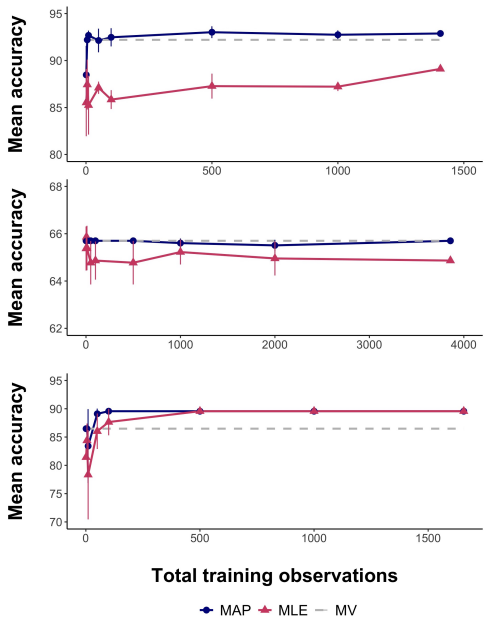

Figure 1: Accuracy on TubeSpam (top), Spouse (center), and RNA (bottom) for increasing training set size. Performance is evaluated on the full test set. Error bars depict standard deviation over 5 replicates.

5 Conclusion

The present work introduces a Bayesian extension of DP where MAP replaces MLE as the objective when learning the joint distribution governing a LF matrix. Experimental results suggest that MAP learns LF accuracies more effectively and improves performance in low-data contexts. MV is found to be an effective heuristic for estimating prior parameters. Incorporating MV as a signal at both training and inference time allowed MAP to meet or exceed both MV and MLE performance in all experiments. Results on RNA demonstrate a real-world application for DP in literature curation. Though automated labeling is not recommended under formal guidelines for systematic review (Page et al., 2021a,b), results suggest that judiciously constructed LFs may accelerate title filtering.

6 Limitations

Generalizability should be assessed on additional datasets. As MV generally offers lower $\hat{y}$ coverage than MLE, improved model performance via forced abstention comes at the price of reduced coverage.

7 Ethical Considerations

This work does not imply obvious ethical considerations, though downstream applications should evaluate label quality uncertainty with respect to user risk in cases where test labels are limited or unavailable.

References

Tulio C. Alberto, Johannes V. Lochter, and Tiago A. Almeida. 2015. TubeSpam: Comment Spam Filtering on YouTube. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 138–143, Miami, FL. IEEE.

Chidubem Arachie and Bert Huang. 2019. Adversarial Label Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3183–3190.

Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchhal, Chris Ré, and Rob Malkin. 2019. Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale. In *Proceedings of the 2019 International Conference on Management of Data*, pages 362–375, Amsterdam Netherlands. ACM.

Benedikt Boecking, Willie Neiswanger, Eric Xing, and Artur Dubrawski. 2021. Interactive Weak Supervision: Learning Useful Heuristics for Data Labeling. *arXiv:2012.06046 [cs, stat]*. ArXiv: 2012.06046.

Oishik Chatterjee, Ganesh Ramakrishnan, and Sunita Sarawagi. 2020. Data Programming Using Continuous and Quality-Guided Labeling Functions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4):3397–3404.

Jason A. Fries, Paroma Varma, Vincent S. Chen, Ke Xiao, Heliodoro Tejeda, Priyanka Saha, Jared Dunnmon, Henry Chubb, Shiraz Maskatia, Madalina Fiterau, Scott Delp, Euan Ashley, Christopher Ré, and James R. Priest. 2019. Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences. *Nature Communications*, 10(1):3111.

Volodymyr Kuleshov, Jialin Ding, Christopher Vo, Braden Hancock, Alexander Ratner, Yang Li, Christopher Ré, Serafim Batzoglou, and Michael Snyder. 2019. A machine-compiled database of genome-wide association studies. *Nature Communications*, 10(1):3341.

Ayush Maheshwari, Oishik Chatterjee, Krishnateja Killamsetty, Ganesh Ramakrishnan, and Rishabh Iyer. 2021. Semi-Supervised Data Programming with Subset Selection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4640–4651, Online. Association for Computational Linguistics.

Emily K. Mallory, Matthieu de Rochemonteix, Alex Ratner, Ambika Acharya, Chris Re, Roselie A. Bright, and Russ B. Altman. 2020. Extracting chemical reactions from text using Snorkel. *BMC Bioinformatics*, 21(1):217.

Emily K. Mallory, Ce Zhang, Christopher Ré, and Russ B. Altman. 2015. Large-scale extraction of gene interactions from full-text literature using DeepDive. *Bioinformatics*, page btv476.

Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and David Moher. 2021a. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *Journal of Clinical Epidemiology*, 134:178–189.

Matthew J Page, David Moher, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and Joanne E McKenzie. 2021b. PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. *BMJ*, page n160.

Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2017a. Data Programming: Creating Large Training Sets, Quickly. *arXiv:1605.07723 [cs, stat]*. ArXiv: 1605.07723.

Alexander J. Ratner, Stephen H. Bach, Henry R. Ehrenberg, and Chris Ré. 2017b. Snorkel: Fast Training Set Generation for Information Extraction. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1683–1686, Chicago Illinois USA. ACM.

A Appendix

MAP$_{mv}$		*Selected values*		
Hyperparameter	*Search values*	*TubeSpam*	*RNA*	*Spouse*
Prior strength scaling factor	$\{10, 100\}$	10	10	10
Learning rate	$\{0.001, 0.01\}$	0.01	0.01	0.01
Initialization value for all α_j	$\{0.8, 0.9, 1.0\}$	1.0	1.0	1.0
Parameter p for prior over y	[0.0, 0.1, ..., 0.9, 1.0]	0.5	0.5	0.5
Force abstention	{True, False}	True	False	True
MAP$_{\alpha^*}$ benchmark		*Selected values*		
Hyperparameter	*Search values*	*TubeSpam*	*RNA*	*Spouse*
Prior strength scaling factor	$\{10, 100\}$	10	10	10
Learning rate	$\{0.001, 0.01\}$	0.01	0.01	0.01
Initialization value for all α_j	$\{0.8, 0.9, 1.0\}$	1.0	1.0	1.0
Parameter p for prior over y	[0.0, 0.1, ..., 0.9, 1.0]	0.5	0.5	0.7
Force abstention	{True, False}	True	False	True
MLE		*Selected values*		
Hyperparameter	*Search values*	*TubeSpam*	*RNA*	*Spouse*
Learning rate	$\{0.001, 0.01\}$	0.001	0.001	0.01
Initialization value for all α_j	$\{0.8, 0.9, 1.0\}$	0.9	0.8	1.0
Snorkel		*Selected values*		
Hyperparameter	*Search values*	*TubeSpam*	*RNA*	*Spouse*
Training epochs	$\{50, 100, 250\}$	50	50	100
Learning rate	$\{0.001, 0.01, 0.1\}$	0.1	0.01	0.01
$\ell 2$ regularization strength	$\{0.0, 0.2, 0.4\}$	0.2	0.4	0.0
CAGE		*Selected values*		
Hyperparameter	*Search values*	*TubeSpam*	*RNA*	*Spouse*
Learning rate	$\{0.01, 0.001\}$	0.001	-	0.01
Guide value for all LFs	$[0.4, 0.5, ..., 0.8, 0.9]$	0.4	-	0.4
Initialization value for all θ_j	$\{0.8, 0.9, 1.0\}$	0.9	-	0.8
Support vector machine		*Selected values*		
Hyperparameter	*Search values*	*TubeSpam*	*RNA*	*Spouse*
$\ell 2$ regularization strength	$\{0.1, 0.01, 0.001, 0.0001\}$	0.001	0.0001	0.0001

Table A1: **Hyperparameter values combinatorially explored through grid search.** Performance was evaluated for the full training set. Selected value combinations were those that yielded the highest scores for the largest number of performance metrics on the validation split of each dataset (among accuracy, F1 score, precision, recall, and AUC ROC). MLE was found to be more sensitive to hyperparameter values than MAP for TubeSpam and RNA. Note that optimal hyperparameter combinations were rarely unique; in this case, model training time and simplicity were prioritized. All models used vanilla SGD except CAGE, which used the Adam optimizer as employed in the original paper. MAP and MLE models employed early stopping using the validation loss with a patience of 5 epochs. MAP$_{mv}$ was trained with priors derived from majority vote over the training data. MAP$_{\alpha^*}$ used simulated optimal priors derived from the empirical LF accuracies with respect to the training data, as computed by the Snorkel Python package. MAP$_{\alpha^*}$ serves as a benchmark for the evaluation of MAP$_{mv}$, providing an experimental control to demonstrate the utility of the majority vote heuristic for prior selection (see Tables A3, A4).

Labeling function	*Rule*	*Coverage*	*Conflicts*	*Accuracy*
`exclude_organism`	-1 if organism is irrelevant, else 0.	0.05	0.04	1.00
`include_terms`	1 if title contains necessary terms, else -1.	1.00	0.21	0.68
`exclude_terms`	-1 if title contains unwanted terms, else 0.	0.37	0.17	1.00

Table A2: **LFs used for transcriptomics literature curation, as designed for this study and applied to the RNA dataset.** A label of -1 denotes that a PubMed title is irrelevant; 1 denotes that the title should proceed to the next stage of systematic review; and 0 denotes abstention. Conflicts are defined as discordance between the output of a given LF and all other LFs. Coverage indicates the percent of observations for which a given LF outputs a label rather than abstaining. Accuracy is measured with respect to the training data by excluding abstained votes (the default behavior of Snorkel). Note that in real-world applications, training labels would be unavailable and LF empirical accuracies would not be calculable.

	TubeSpam						
	MAP_{α^*}	MAP_{mv}	*MLE*	*MV*	*SNO*	*CAGE*	*SVM*
F1	94.29	92.59	91.51	91.39	91.39	75.28	95.07
Accuracy	94.58	93.22	92.20	92.20	92.20	70.17	95.25
Precision	97.06	99.20	97.64	99.19	99.19	63.21	96.43
Recall	91.67	86.81	86.11	84.72	84.72	93.06	93.75
AUC ROC	94.51	93.07	92.06	92.03	92.03	70.70	95.22
Coverage $\hat{y}$	75.26	75.26	75.26	75.26	75.26	75.26	75.26
	Spouse						
	MAP_{α^*}	MAP_{mv}	*MLE*	*MV*	*SNO*	*CAGE*	*SVM*
F1	48.92	48.92	48.92	48.92	48.92	44.59	24.56
Accuracy	65.70	65.70	65.70	65.70	65.70	60.39	79.23
Precision	34.34	34.34	34.34	34.34	34.34	30.56	41.18
Recall	85.00	85.00	85.00	85.00	85.00	82.50	17.50
AUC ROC	73.04	73.04	73.04	73.04	73.04	68.79	55.76
Coverage $\hat{y}$	18.80	18.80	18.80	18.80	18.80	18.80	18.80
	RNA						
	MAP_{α^*}	MAP_{mv}	*MLE*	*MV*	*SNO*	*CAGE*	*SVM*
F1	87.30	87.30	87.30	87.30	87.30	-	85.04
Accuracy	86.48	86.48	86.48	86.48	86.48	-	85.63
Precision	80.88	80.88	80.88	80.88	80.88	-	86.83
Recall	94.83	94.83	94.83	94.83	94.83	-	83.33
AUC ROC	86.64	86.64	86.64	86.64	86.64	-	85.59
Coverage $\hat{y}$	77.17	77.17	77.17	77.17	77.17	-	77.17

Table A3: **Comparative performance evaluation on test set instances for which all models vote.** Models were trained on the full training set. As all models are evaluated on an identical subset of the test data, $\hat{y}$ coverage is identical. MV, MAP_{mv}, the MAP_{α^*} benchmark using simulated optimal priors, MLE, CAGE, Snorkel (SNO), and SVM performance scores are expressed as percentages. Dummy accuracy when predicting the majority class is 51.28% of data for TubeSpam, 92.64% for Spouse, and 62.17% for RNA. CAGE could not be run on RNA, as this method requires all LFs to output only a single label (positive or negative) when triggered to vote.

	TubeSpam						
	MAP_{α^*}	MAP_{mv}	*MLE*	*MV*	*SNO*	*CAGE*	*SVM*
F1	94.29	92.59	89.56	91.39	88.07	70.32	94.74
Accuracy	94.58	93.22	89.11	92.20	87.97	62.76	94.64
Precision	97.06	99.20	94.77	99.19	96.88	59.45	95.45
Recall	91.67	86.81	84.89	84.72	80.73	86.07	94.03
AUC ROC	94.51	93.07	89.58	92.03	88.77	62.14	94.66
Coverage $\hat{y}$	75.26	75.26	89.03	75.26	89.03	100.0	100.0
	Spouse						
	MAP_{α^*}	MAP_{mv}	*MLE*	*MV*	*SNO*	*CAGE*	*SVM*
F1	48.92	48.92	46.58	48.92	46.58	12.91	17.54
Accuracy	65.70	65.70	64.86	65.70	64.86	15.44	91.46
Precision	34.34	34.34	33.66	34.34	33.66	6.98	30.30
Recall	85.00	85.00	75.56	85.00	75.56	85.19	12.35
AUC ROC	73.04	73.04	68.85	73.04	68.85	47.54	55.05
Coverage $\hat{y}$	18.80	18.80	20.16	18.80	20.16	100.0	100.0
	RNA						
	MAP_{α^*}	MAP_{mv}	*MLE*	*MV*	*SNO*	*CAGE*	*SVM*
F1	87.30	87.30	87.30	87.30	82.50	-	81.23
Accuracy	89.56	89.56	89.56	86.48	84.78	-	85.43
Precision	80.88	80.88	80.88	80.88	73.01	-	79.23
Recall	94.83	94.83	94.83	94.83	94.83	-	83.33
AUC ROC	90.60	90.60	90.60	86.64	86.75	-	85.02
Coverage $\hat{y}$	100.0	100.0	100.0	77.17	100.0	-	100.0

Table A4: **Comparative performance evaluation on the full test set.** Models were trained on the full training set. Testing on all instances in the test set highlights differences in $\hat{y}$ coverage across models, as well as MAP performance gains from forced abstention. MV, MAP_{mv}, the MAP_{α^*} benchmark using simulated optimal priors, MLE, CAGE, Snorkel (SNO), and SVM performance scores are expressed as percentages. Dummy accuracy when predicting the majority class is 51.28% of data for TubeSpam, 92.64% for Spouse, and 62.17% for RNA. CAGE could not be run on RNA, as this method requires all LFs to output only a single label (positive or negative) when triggered to vote.

	TubeSpam		**RNA**	
	MV-concordant	*MV-discordant*	*MV-concordant*	*MV-discordant*
MAP model				
Accuracy (%)	93.15	64.91	86.48	100.00
MV abstentions (%)	12.84	94.74	0.0	100.00
MLE model				
Accuracy (%)	94.41	65.07	86.48	100.00
MV abstentions (%)	13.07	85.71	0.0	100.00

Table A5: **Disaggregating performance by concordance with majority vote (MV).** Performance is evaluated for the full training and test sets. Accuracy when predicting on instances for which a labeling model and MV agree (MV-concordant) versus disagree (MV-discordant) explain the benefit of forced abstention on TubeSpam. The majority of discordant predictions were MV abstentions for both datasets. MAP trained on RNA does not benefit from forced abstention, as the model achieves 100% accuracy on MV-discordant instances, all of which were instances for which MV abstained.

	TubeSpam		Spouse		RNA	
Model	$\|\|\alpha^* - prior\|\|_2$	$\|\|\alpha^* - \hat{\alpha}\|\|_2$	$\|\|\alpha^* - prior\|\|_2$	$\|\|\alpha^* - \hat{\alpha}\|\|_2$	$\|\|\alpha^* - prior\|\|_2$	$\|\|\alpha^* - \hat{\alpha}\|\|_2$
MAP$_{mv}$	0.506	0.468	0.782	0.757	0.307	0.103
MAP$_{rn}$	1.476	1.470	0.578	0.566	0.712	0.726
MAP$_{\alpha^*}$	0.000	**0.047**	0.000	**0.329**	0.000	**0.046**
MLE	-	0.486	-	0.512	-	0.467

Table A6: Distance between α^*, model priors, and $\hat{\alpha}$ for MAP with MV priors (MAP$_{mv}$), MAP with random priors (MAP$_{rn}$), MAP with empirical accuracy priors (MAP$_{\alpha^*}$), and MLE, as measured by ℓ^2-norms. Performance is evaluated for the full training and test sets.

$$\begin{bmatrix} Row & LF0 & LF1 & LF2 & LF3 & LF4 & LF5 & LF6 & LF7 & LF8 & LF9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 245 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 246 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 247 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 248 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 249 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} y \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ ... \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} \hat{y} \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ ... \\ 1 \\ -1 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

Figure A1: **Sample labeling function matrix with gold label (y) and predicted label vectors ($\hat{y}$).** A labeling function matrix may be sparse, with zeros denoting abstention. Zero values are permissible in $\hat{y}$ under the abstention tie-breaking policy, along with class assignments $\in \{-1, 1\}$. Note that y is unavailable at training time and may also be unavailable at test time.

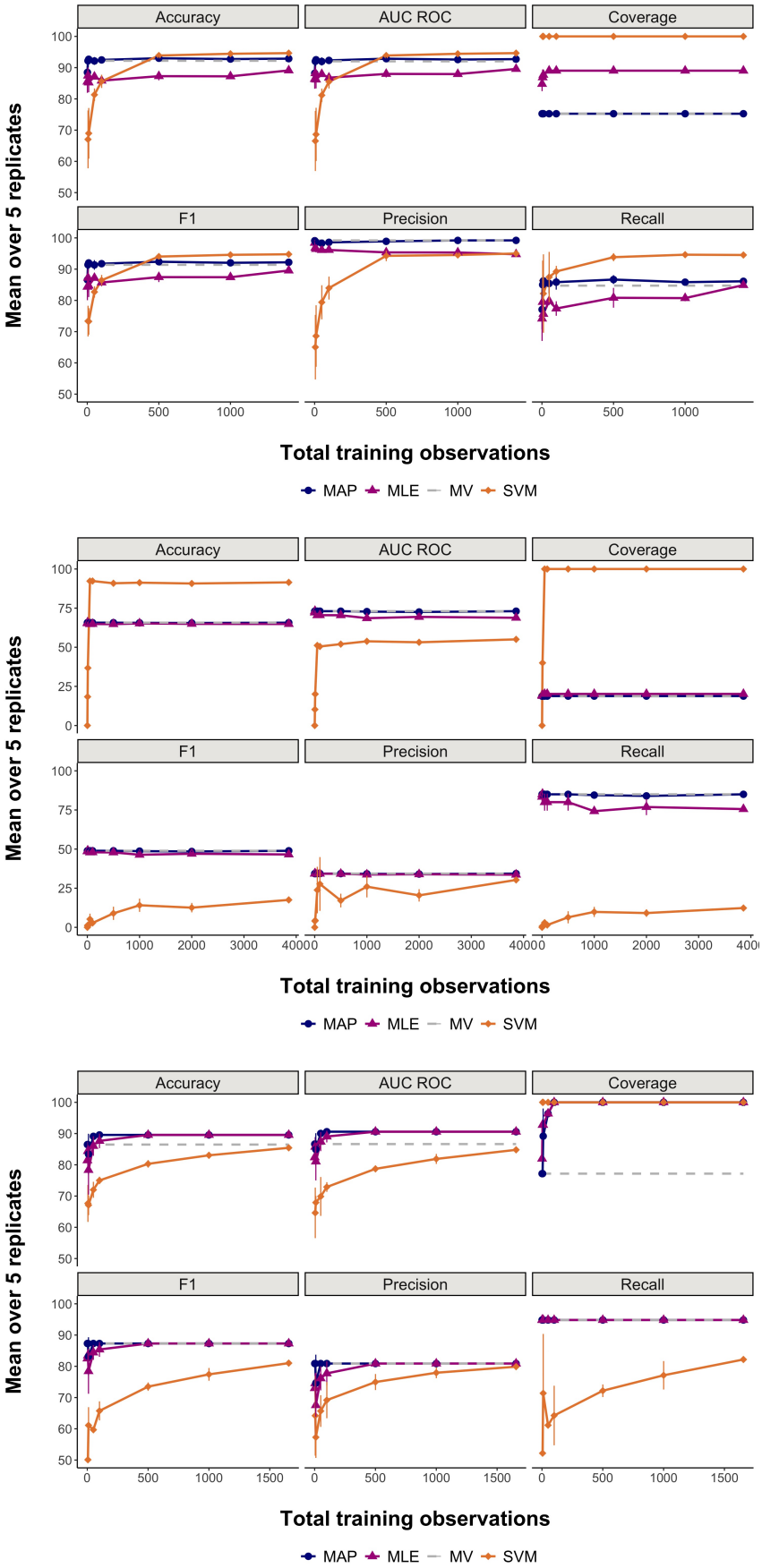

Figure A2: **Low-data regimes for TubeSpam (top), Spouse (center), and RNA (bottom).** Performance of MAP with MV priors, MLE, MV, and SVM as training set size increases. Performance is evaluated for the full test set. Means are across five random splits and error bars depict standard deviation.

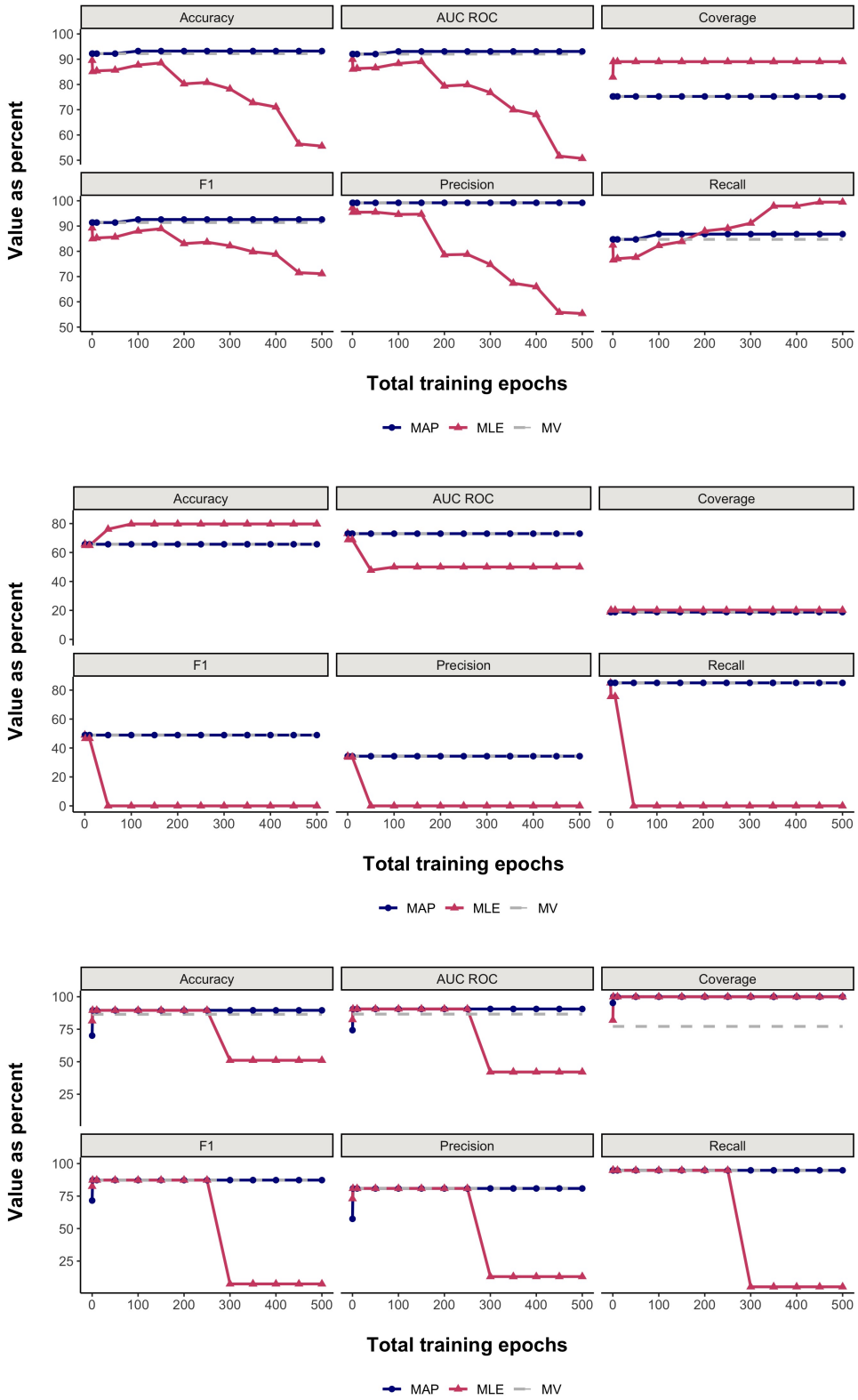

Figure A3: **Performance stability on TubeSpam (top), Spouse (center), and RNA (bottom).** MAP and MLE performance stability as training epochs increase. Performance is evaluated for the full training and test sets. Majority vote (MV) represents baseline performance. Similar trends were observed for regularization vs MLE by Chatterjee et al. 2020 in their comparison of CAGE and Snorkel.

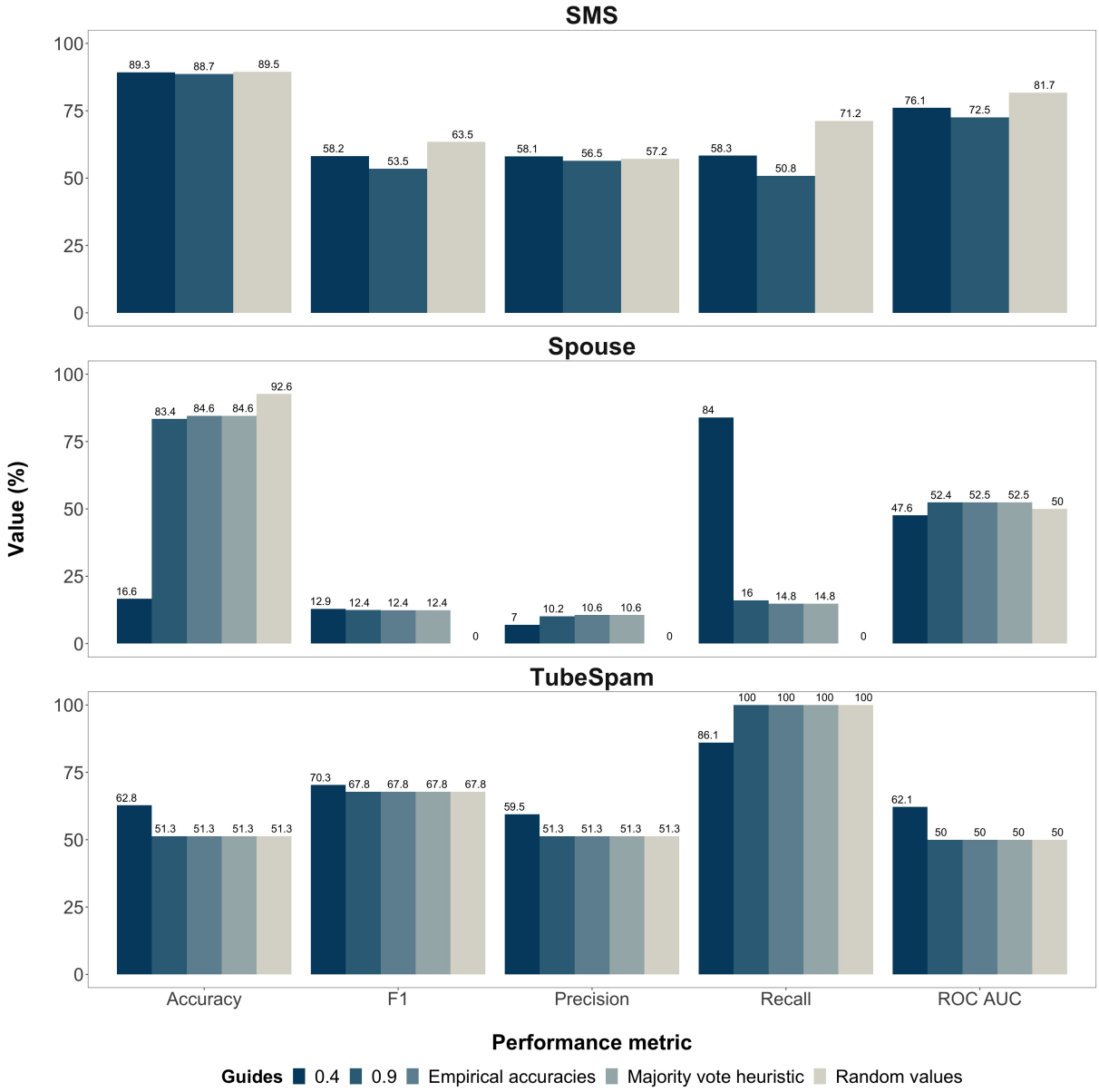

Figure A4: **CAGE performance with varying guide values.** Guides were set to **1)** the values used in the original publication (Chatterjee et al., 2020), which assigns a guide value of 0.9 to all LFs; **2)** a randomly selected low value uniformly assigned to all LFs (0.4); **3)** random values $\in (0...1)$ that vary per LF; **4)** the empirical accuracies of each respective LF on the training data; and **5)** the majority vote heuristic used to derive the Bayesian priors presented in this paper. For SMS, CAGE model hyperparameters and the SMS dataset are exactly as provided by the authors (https://github.com/oishik75/CAGE), with the exception of guide values. The continuous LF loss function was used for SMS, while the discrete LF loss was employed for TubeSpam and Spouse.

For the SMS dataset, both methods of random guide selection (0.4 and random values) outperformed the guides reported in Chatterjee et al. 2020 for all performance metrics. Empirical accuracies were unavailable for SMS training data. For TubeSpam, uniformly assigning guide values of 0.4 performed best, despite LF empirical accuracies ranging from 59.4% to 100%. Similarly, uniform guide values of 0.4 performed best for Spouse on the basis of F1 and recall (the metrics of interest, as 92.6% of true labels were negative) despite LF empirical accuracies ranging from 39.9% to 96.8%. Despite their intention to be estimates of LF accuracy, guide values derived from empirical accuracies and the majority vote heuristic performed equivalently to non-uniform random values for Tubespam. These results suggest that CAGE guides offer a less intuitive and less natural interpretation than the Bayesian priors introduced in this work.