Using Large Language Model Annotations for the Social Sciences: A General Framework of Using Predicted Variables in Downstream Analyses[*]

Naoki Egami[†] Musashi Hinck[‡] Brandon M. Stewart[§] Hanying Wei[¶]

First Version: May 13, 2024

This Version: June 19, 2024

Abstract

Social scientists use automated annotation methods, such as supervised machine learning and, more recently, large language models (LLMs), that can predict labels and generate text-based variables. While such predicted text-based variables are often analyzed as if they were observed without errors, we show that ignoring prediction errors in the automated annotation step leads to substantial bias and invalid confidence intervals in downstream analyses, even if the accuracy of the automated annotations is high, e.g., above 90%. We propose a framework of *design-based supervised learning* (DSL) that can provide valid statistical estimates, even when predicted variables contain non-random prediction errors. DSL employs a doubly robust procedure to combine predicted labels and a smaller number of expert annotations. DSL allows scholars to apply advances in LLMs to social science research while maintaining statistical validity. We illustrate its general applicability using two applications where the outcome and independent variables are text-based.

[*] The proposed methodology is implemented via our R package, `dsl` (http://dsl.software). This paper extends and generalizes the methods we proposed in Egami et al. (2023). We appreciate the excellent research assistance by Songpo Yang, TaeJun Seo, and Benedikt Ströbl. We would like to thank Mitchell Bosley, Christian Fong, Fabrizio Gilardi, Max Goplerud, Katie Keith, John Marshall, Molly Offer-Westort, Joe Ornstein, Nicholas Pangakis, Tom Robinson, Arthur Spirling, and Hannah Waight for their thoughtful comments on an early draft. We also appreciate comments from participants at the Political Methodology Summer meeting, the 2023 TADA conference, the 2023 Neurips meeting, and the Conference on Generative AI in Social Science at Yale.

[†] Corresponding Author. Assistant Professor, Department of Political Science, Columbia University. Email: naoki.egami@columbia.edu. URL: https://naokiegami.com

[‡] Postdoctoral Research Associate, Data-Driven Social Science Initiative, Princeton University. Email: mj2976@princeton.edu. URL: https://muhark.github.io/about

[§] Corresponding Author. Associate Professor, Department of Sociology and the Office of Population Research, Princeton University. Email: bms4@princeton.edu. URL: https://brandonstewart.org

[¶] Ph.D. student, Department of Political Science, Columbia University. Email: hw2893@columbia.edu. URL: https://polisci.columbia.edu/content/hanying-wei

1 Introduction

Over the last decade, social scientists have developed and applied a variety of text analysis and natural language processing methods to study a large collection of documents. In text-as-data applications, one of the most common tasks is text annotation (or text classification) to generate text-based variables for subsequent statistical analyses. For example, Pan and Chen (2018) annotate whether each online post accuses local Chinese officials of corruption so that they can later study whether and how much such online complaints are censored. Fowler et al. (2021) annotate the tone of political ads and then analyze how politicians strategically change the tone of political advertising online and offline.

In an ideal world without any budget and time constraints, researchers, as domain experts, might want to carefully annotate all the documents they use in their main statistical analyses. However, this is often infeasible for the scale of corpora today. To facilitate large-scale annotations, social scientists have used a variety of supervised machine learning (ML) methods to automate this text annotation step by training machines to mimic expert coding (e.g., Grimmer and Stewart 2013; Barberá et al. 2021). More recently, a growing number of papers propose using large language models (LLMs), such as ChatGPT, to automate text annotations by predicting text labels (e.g., Bommasani et al. 2021; Ornstein, Blasingame, and Truscott 2022; Gilardi, Alizadeh, and Kubli 2023; Linegar, Kocielnik, and Alvarez 2023; Ollion et al. 2023; Pangakis, Wolken, and Fasching 2023; Ziems et al. 2024). Given that researchers can adapt LLMs to perform a wide range of text annotation tasks by simply changing prompts, automated LLM annotations present exciting opportunities for the social sciences.

While text annotation is essential, it is only the first step. Social scientists are often primarily interested in using text labels predicted by automated methods as key variables in subsequent statistical analyses (Hopkins and King 2010; Egami et al. 2022; Grimmer, Roberts, and Stewart 2022). In the vast majority of current applications, researchers treat predicted text-based variables as if they were observed without errors; that is, they ignore *prediction errors* created by automated text annotation. The natural intuition behind this common practice is that when the prediction accuracy is high enough, the underlying automated text annotation model, whether it is an LLM or a supervised ML model, "learned" how to label texts, and prediction errors are small enough that analysts can ignore them. This problem has been previously raised but is seldom addressed in practice (Benoit, Laver, and Mikhaylov 2009; Wang, McCormick, and Leek 2020; Fong and Tyler 2021; Knox, Lucas, and Cho 2022).

We clarify that ignoring such prediction errors in the first step of text annotation, even if the errors are small, leads to substantial bias, invalid confidence intervals, and inaccurate p-values in downstream statistical analyses of text-based variables. Biases from prediction errors exist even when the prediction accuracy in the text classification step is extremely high, e.g., above 90% or even at 95%. This is because prediction errors are not random—prediction errors are correlated with observed and unobserved variables we include in downstream analyses. In practice, this means that substantive and statistical conclusions can easily flip if researchers choose slightly different automated text annotation methods, as we empirically show in Section 5.

In this paper, we make two contributions. First, we develop a general framework for using predicted variables in downstream statistical analyses without suffering from bias due to prediction errors. Unlike the existing approaches, the proposed approach, which we call *design-based supervised learning* (DSL), allows researchers to obtain statistically valid estimates and standard errors, even when automated text annotation methods have arbitrary non-random prediction errors. Second, we show how this general framework allows researchers to unlock the recent advances in LLMs without suffering from bias in downstream statistical analysis.

Preview of DSL

DSL combines large-scale (potentially biased) automated annotations and a smaller number of expert annotations using a bias-correction step built on doubly robust estimation (Robins,

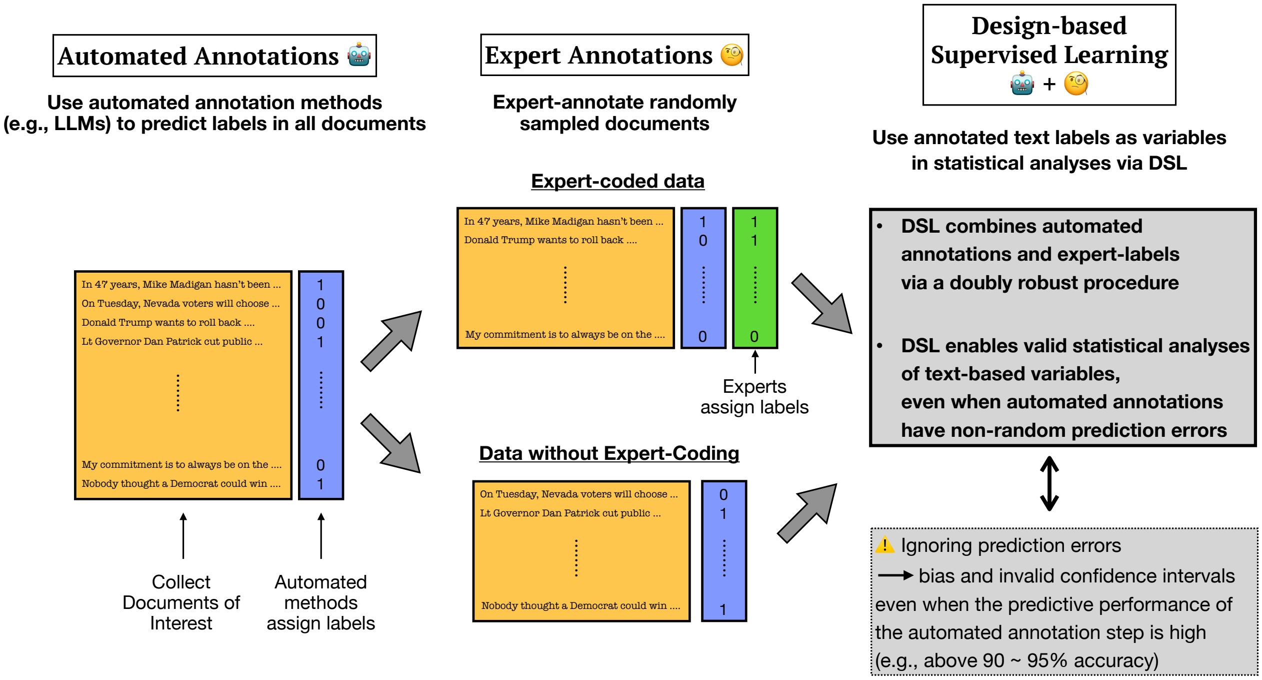

Automated Annotations 🤖
Use automated annotation methods
(e.g., LLMs) to predict labels in all documents
Expert Annotations 🧐
Expert-annotate randomly
sampled documents
Design-based
Supervised Learning
🤖 + 🧐
Use annotated text labels as variables
in statistical analyses via DSL
In 47 years, Mike Madigan hasn't been ...
On Tuesday, Nevada voters will choose ...
Donald Trump wants to roll back ....
Lt Governor Dan Patrick cut public ...
My commitment is to always be on the ....
Nobody thought a Democrat could win ....
1
0
0
1
0
1
Collect
Documents of
Interest
Automated
methods
assign labels
Expert-coded data
In 47 years, Mike Madigan hasn't been ...
Donald Trump wants to roll back ....
My commitment is to always be on the ....
1
0
0
1
1
0
Experts
assign labels
Data without Expert-Coding
On Tuesday, Nevada voters will choose ...
Lt Governor Dan Patrick cut public ...
Nobody thought a Democrat could win ....
0
1
1
• DSL combines automated annotations and expert-labels via a doubly robust procedure
• DSL enables valid statistical analyses of text-based variables, even when automated annotations have non-random prediction errors
⚠️ Ignoring prediction errors
→ bias and invalid confidence intervals
even when the predictive performance of
the automated annotation step is high
(e.g., above 90 ~ 95% accuracy)

Figure 1: **Overview of the Design-based Supervised Learning (DSL).**

Rotnitzky, and Zhao 1994; Chernozhukov et al. 2018). Figure 1 provides an overview of the method. While DSL provides statistically valid estimates regardless of the prediction accuracy of the underlying automated annotation method, DSL can reduce standard errors when the automated annotation method becomes more accurate. This pairs nicely with LLMs, which are rapidly improving over time: as LLMs improve, estimation with DSL becomes more efficient.

Importantly, DSL only requires one transparent assumption—that researchers control the process through which documents are sampled for expert annotations. One of the most common ways to guarantee the assumption is to randomly sample documents for expert coding. This assumption is straightforward to guarantee by research design in many social science applications, which gives the name, *design-based* supervised learning. We do not make any assumptions about prediction errors in the underlying automated annotation method, and as a result, the proposed method is applicable to any automated labeling procedure (including LLMs that have not yet been released).

When we use the term "expert annotations," we define it to be a procedure that defines the benchmark against which the quality of the automated text annotation is evaluated, as done in the established supervised machine learning literature for decades (Hopkins and King 2010; Grimmer and Stewart 2013). We call these labels "expert annotations" because we believe that the target procedure that the modal social scientist is trying to approximate is domain experts (e.g., the principal investigators) *carefully* labeling all documents by hand. However, our method does not require "human" experts to provide this target procedure. More generally, our procedure is applicable whenever researchers have an annotation procedure that they wish to implement for the entire sample (but they cannot do so due to costs) and a cheaper automated annotation procedure approximating this target.[1] For example, if users want to correct annotations by lower-quality smaller LLMs with more expensive annotations by larger LLMs as the benchmark, the same proposed methodology can be applied. Similarly, if the ideal procedure is to have a panel of experts vote on each classification, that can also be the source of the expert annotation. We provide concrete, practical recommendations for how to think about errors or mistakes that remain in "expert annotations" in Section 6. Researchers can combine DSL with widely used strategies for handling simple errors and uncertainties in expert annotations (e.g., Benoit, Laver, and Mikhaylov 2009; Hopkins and King 2010).

[1] Formally, the statistical properties of DSL are defined with respect to what we would have observed if all documents had been labeled using a given target procedure. For certain social science applications, researchers might be afraid that there is no procedure that they can use to validate labels (i.e., even the principal investigators cannot create a well-defined codebook and apply it to validate labels from predictive methods). Such problems are often under the domain of construct validity and operationalization, and they have to be primarily addressed by substantive theories, not by statistical methods.

Our proposed approach is a general-purpose method that works in a wide range of text-as-data applications. DSL can incorporate any automated text annotation method (including dictionaries, supervised ML methods, and any LLMs) and works for a variety of common downstream analyses scholars conduct with text-based variables: linear, logistic, multinomial-logistic, Poisson, and linear fixed-effects regression, as well as the estimation of category proportions and causal inference with texts.[2] While we focus on the use of LLMs in text-as-data applications because of the rapid rate of improvement and interest in the field, our proposed framework can also be used in any application where predictive methods are used to scale up measurements, e.g., analyses of images, videos, and audio, which we discuss in Section 6. We offer an easy-to-use R package, `dsl`, which can implement all the methods described in this paper with simple functions.

Related Literature

Our paper contributes to the growing literature on the use of predicted variables in statistical analyses. A number of papers develop methods for specific scenarios by making assumptions about the underlying data-generating process and prediction errors (e.g., Wang, McCormick, and Leek 2020; Fong and Tyler 2021; Zhang 2021; Knox, Lucas, and Cho 2022). In contrast to these papers, we only assume that researchers control the sampling process for expert annotations, and we do not make any assumption about the nature of prediction errors, which is particularly difficult to justify in applications of LLMs. Our paper is most closely related to recent methods that build on the doubly robust estimation (Robins, Rotnitzky, and Zhao 1994; Chernozhukov et al. 2018) to deal with predicted variables, such as the original proposal of DSL (Egami et al. 2023), prediction-powered inference (e.g., Angelopoulos et al. 2023), and

[2] In general, the proposed DSL framework can be applied to any statistical method that can be written as a convex optimization problem or a moment estimator.

model-assisted impact analysis (Mozer and Miratrix 2023). Methodologically, our paper extends these previous results in three ways. First, while these papers only cover cases of text-based outcome variables, we cover cases where any subset of the outcome and independent variables are text-based. This methodological generalization is fundamental because about 45% of applications use text-based variables as independent variables. Second, we develop a data-driven power analysis to help users determine the required number of expert annotations. Third, we derive DSL estimators for a much wider range of downstream analyses popular in the social sciences, including linear fixed effects regression and the instrumental variable method. We provide additional technical discussions in Appendix A.

Roadmap

In the next section, we provide an introduction to annotation using LLMs. In Section 3, we review how social scientists use text annotations in downstream analysis and clarify that the current practice of directly using predicted variables in downstream analyses leads to substantial bias and invalid confidence intervals. Section 4 outlines our solution, DSL, and provides intuition for how it works. Section 5 demonstrates our approach with two empirical applications, Fowler et al. (2021) and Pan and Chen (2018). Before concluding, we offer practical guidance in Section 6 to help researchers apply these techniques to their own work—answering questions such as how to determine the required number of expert annotations.

2 Automated Text Annotation

One of the most fundamental steps in many text-as-data research projects is to annotate documents. Over the last decade, scholars have used automated annotation methods to facilitate this time-consuming step by training machines to mimic expert annotation. In this section, we discuss how researchers can use the recent advances in LLMs for a wide range of text annotation tasks. We then clarify the potential risks of using LLM annotations, which motivates our main methodological contributions in Section 3 and Section 4.

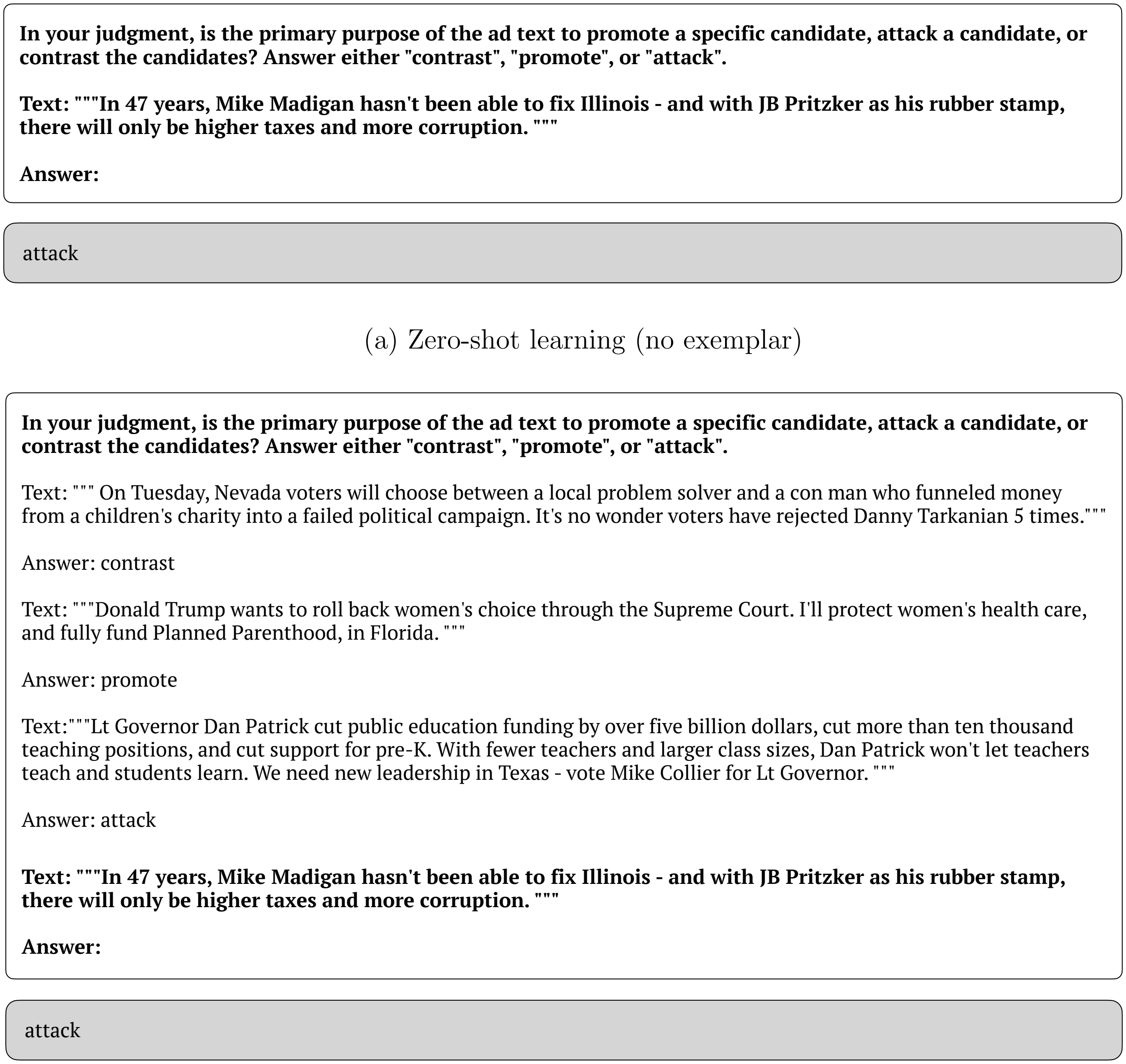
2.1 Large Language Models as Text Classifier

An increasing number of social scientists use LLMs as automated text classifiers: researchers simply describe the annotation task in natural language instructions, and the LLM generates text labels by predicting the most appropriate text to follow such a request. For example, scholars have used LLMs to annotate sentiments, ideology, topics, hate speech, attitudes toward immigrants, and support for a war, among others. See a wide range of examples we summarize in Appendix F.

2.1.1 How to Use LLMs as Text Classifiers

To illustrate this exciting potential, we use Fowler et al. (2021) as an example. The text annotation task here is to code the tones of ads into three categories ("Attack," "Contrast," and "Promote"). In the codebook developed in the well-known Wesleyan Media Project and used by Fowler et al. (2021), the tone of ads is defined as an answer to the following question. "In your judgment, is the primary purpose of the ad text to promote a specific candidate, attack a candidate, or contrast the candidates?" Instead of providing the codebook to trained expert coders, we can supply the same codebook to LLMs (see Figure 2-(a)) by first describing the codebook, then supplying Text to be classified, and finally prompting the LLM to Answer. In this example, when we use GPT 4, it understands the codebook and annotates a given document correctly as "Attack" (a response from the LLM is in a gray box).

More generally, text classification by LLMs can be performed in two steps. First, researchers need to choose which LLM to use. For example, famous commercial models like GPT 4 and GPT 3.5 are currently popular and famous, but researchers can also use open-source LLMs like Llama 2. We note that our discussion is general and applicable to any LLMs, including those developed in the future. In the second step, researchers decide on a *prompt*, i.e., a codebook



In your judgment, is the primary purpose of the ad text to promote a specific candidate, attack a candidate, or contrast the candidates? Answer either "contrast", "promote", or "attack".

Text: """In 47 years, Mike Madigan hasn't been able to fix Illinois - and with JB Pritzker as his rubber stamp, there will only be higher taxes and more corruption. """

Answer:

attack

(a) Zero-shot learning (no exemplar)

In your judgment, is the primary purpose of the ad text to promote a specific candidate, attack a candidate, or contrast the candidates? Answer either "contrast", "promote", or "attack".

Text: """ On Tuesday, Nevada voters will choose between a local problem solver and a con man who funneled money from a children's charity into a failed political campaign. It's no wonder voters have rejected Danny Tarkanian 5 times."""

Answer: contrast

Text: """Donald Trump wants to roll back women's choice through the Supreme Court. I'll protect women's health care, and fully fund Planned Parenthood, in Florida. """

Answer: promote

Text:"""Lt Governor Dan Patrick cut public education funding by over five billion dollars, cut more than ten thousand teaching positions, and cut support for pre-K. With fewer teachers and larger class sizes, Dan Patrick won't let teachers teach and students learn. We need new leadership in Texas - vote Mike Collier for Lt Governor. """

Answer: attack

Text: """In 47 years, Mike Madigan hasn't been able to fix Illinois - and with JB Pritzker as his rubber stamp, there will only be higher taxes and more corruption. """

Answer:

attack

(b) Few-shot learning (exemplars in non-bold face)

Figure 2: **How to Use LLMs as Text Classifiers.**
Note: In (a) zero-shot learning, the basic prompt consists of a codebook (the first two lines), a text to be classified (the next two lines), and an answer box (the last line). A response from an LLM is represented in a gray box. In (b) few-shot learning, while keeping the basic components (parts in a bold face), users can add examples (parts in a non-bold face) in the middle.

and an instruction about a given annotation task. Many papers show that recent LLMs can perform a wide range of text annotation tasks by simply changing this prompt (e.g., Bommasani et al. 2021; Ornstein, Blasingame, and Truscott 2022; Gilardi, Alizadeh, and Kubli 2023; Ziems et al. 2024). See Appendix F for examples of different types of prompts used in a wide range of social science annotation tasks. Researchers can also provide some examples (several pairs of texts and labels), also known as few-shot learning or in-context learning, to improve the prediction accuracy.[3] In Figure 2-(b), while we keep the codebook, a text to be classified, and an answer box (parts in boldface), we added three pairs of texts and answers as examples in the middle (parts in a non-bold face). The biggest benefit of using LLM annotations is that researchers can finish this automated text annotation step within a day for most applications, which is even faster than the classical supervised ML approach.

2.1.2 Empirical Illustration of LLM Annotation

While the idea of using LLMs for text classification sounds promising, does it work in practice? In this section, we use two empirical applications of ours and the literature review to empirically illustrate the performance in a wide range of settings, which clarifies the promise and challenges.

Our first application is based on Fowler et al. (2021). In particular, we use the 13040 ads that are expert-coded by the original authors and examine the prediction accuracy of LLMs classifying the tone of ads. The second application is based on Pan and Chen (2018). We use their expert-coded 1412 citizen complaints to evaluate how well LLMs can classify whether each complaint accuses of wrongdoing by prefecture-level officials in China.

Panels (a) and (b) in Figure 3 report F1 scores[4] for six versions of LLMs: GPT 4, GPT

[3] How to choose examples is an active area of research in NLP. In practice, we recommend including illustrative examples that researchers would include when creating a codebook and training human coders.

[4] F1 score is a harmonic mean of the recall and precision, i.e., F1 = $2/(\text{recall}^{-1} + \text{precision}^{-1})$, and it is the most standard measure of prediction performance when categories

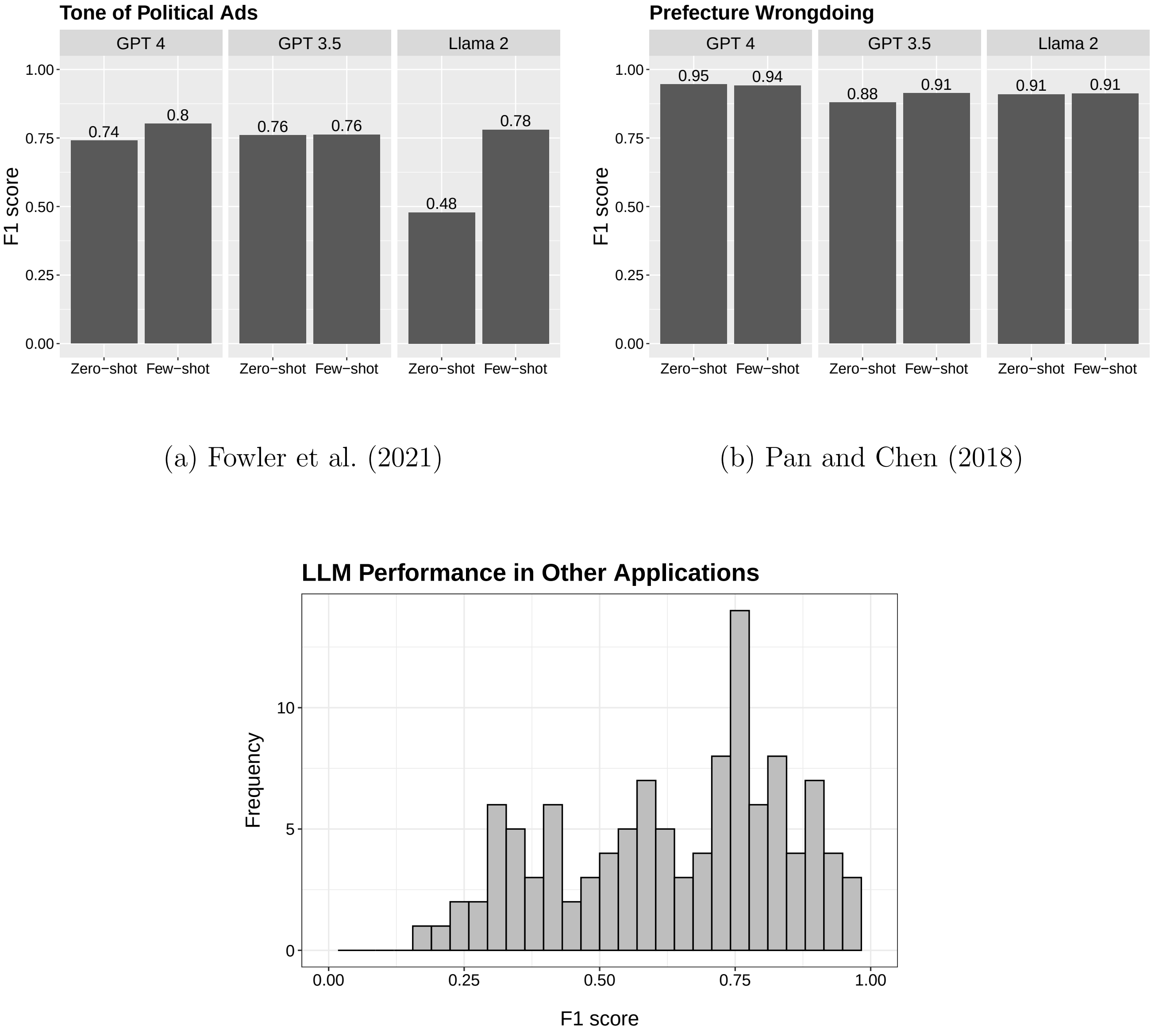
3.5, and Llama 2 with zero-shot and few-shot learning. We provide the exact implementation details in Appendix G and H. Several points are worth noting. First, most of the LLMs can achieve F1 scores of about 75 $\sim$ 90%. This is promising and surprising given that these LLMs were *not* trained for these text annotation tasks, and LLMs were only given the codebook and a couple of examples (in the case of few-shot learning).

Second, the prediction performance varies across models and applications. In these two applications, F1 scores range from 48% to 95%. To further illustrate this wide variation in prediction performance, we also analyze a diverse set of empirical validation studies. In particular, based on a review paper by Ollion et al. (2023), we collected eight recent papers that examine the performance of LLM annotations in the social sciences, and we analyzed 113 text annotations tasks in total (see more details in Appendix F.2.2). We find that F-1 scores range from as low as 20% to more than 95%, and many tasks show about 70 $\sim$ 80% (see Panel (c) in Figure 3). This huge variation in prediction accuracy is a common feature of LLM annotations in the social sciences, and it is one of the key potential challenges of using LLMs, which we turn to next.

2.2 Potential Risks of LLM Annotation

As with any new technology, we have to carefully understand the potential risks as well as its promises. Even though LLMs have huge potential in many different text annotation tasks, they are, of course, not perfect and make *prediction errors.* While prediction errors can arise in any prediction method, including the classical supervised ML method, prediction errors in LLM classification are particularly difficult to understand for many reasons.

are imbalanced. Classification accuracy is another popular measure, but it can be artificially high when categories are imbalanced. For readers who are more familiar with accuracy, we also report the figures based on accuracy in Appendix G and H, finding qualitatively similar results.



(a) Fowler et al. (2021)

(b) Pan and Chen (2018)

(c) 113 annotation tasks in 8 papers

Figure 3: **Prediction Performance of LLMs as Text Classifiers.**

First, as we saw in Section 2.1.2, the amount and direction of prediction errors in LLM classification can substantially vary depending on tasks, prompts, LLM models, and other unknown parameters in models. Most importantly, these variations in prediction errors are unknown and unpredictable to users. Recent LLMs are large-scale black box models that exhibit incredible language capacities in many different tasks for which LLMs were *not* trained

(Bommasani et al. 2021). We are constantly surprised by how well LLMs can perform many different tasks, but this also means that we do not fully understand when and why they might fail.

Second, social scientists often study complex, nuanced concepts expressed in documents, and thus, text annotation tasks in the social sciences are inherently difficult. Indeed, for many applications, even domain experts need several rounds of pilot coding and extensive discussions to create, polish, and finalize a codebook to define how to label texts. Given the inherent difficulty of the task, prediction errors are inevitable.

Third, many recent LLMs lack the basic scientific requirement of transparency and replicability. In particular, many recent successful LLMs, e.g., GPTs, are proprietary methods, and as a result, users and research communities, in general, do not know the training data or exact training procedures that LLMs use (Spirling 2023). Without access to the training data and training procedures, it is nearly impossible for users to understand the prediction errors that LLMs make.

Finally, a large number of papers have shown that LLMs also inherit unknown social, political, and racial biases contained in the unknown large-scale training data (see, e.g., Bender et al. 2021). Prediction errors in LLMs can come not only from technical reasons but also from deeper reasons related to ethics and fairness, which further complicates the understanding of prediction errors.

In sum, it is extremely difficult or nearly impossible to fully understand how prediction errors occur in LLM classification. In Section 3, we clarify that such prediction errors in LLM classification, and more generally in any automated text classification approach, can significantly bias downstream text analyses, if the errors are ignored when using predicted text labels.

3 Predicted Text Labels as Variables in Downstream Analyses

While document-level text classification is essential, text annotation is rarely the end goal of social science research. It is only the first step. Social scientists are often interested in using predicted text labels as variables in subsequent statistical analyses. Even though researchers often analyze predicted text-based variables as if they were observed without any error, this section clarifies that ignoring such *prediction errors*[5] in the first step of text annotation, even if the errors are small, leads to substantial bias, invalid confidence intervals, and wrong p-values in downstream statistical analyses of text-based variables.

3.1 Setup and Quantity of Interest

We begin by defining statistical analyses we conduct after text annotation. Here, we focus on the most common regression analyses, and we discuss other common analyses, such as causal inference with texts, in Section 6.

Suppose researchers are interested in analyzing N documents. For each document i, we define Y_i as the outcome of interest and $\mathbf{X}_i$ as independent variables. Using general notation, we can define the quantity of interest as coefficients β of a generalized linear model.

$$\mathbb{E}(Y_i \mid \mathbf{X}_i) = f(\mathbf{X}_i^\top \beta) \tag{1}$$

[5] In this paper, we define prediction errors to be the discrepancy between the text labels predicted by an automated text annotation method and the expert annotations. As emphasized in Section 1, expert annotation is defined as a procedure that defines the benchmark against which the quality of the automated text annotation is evaluated, and DSL does not require that human experts provide the benchmark.

where $f(\cdot)$ is an inverse of a canonical link function for the generalized linear model. This general setup incorporates a wide range of common statistical analyses, such as linear, logistic, multinomial logistic, Poisson, and linear fixed-effects regression, as well as the estimation of category proportions over time or across groups.[6] Importantly, we only view coefficients β as a low-dimensional summary, and thus, this paper does not assume the underlying data-generating process follows a specified parametric model (Lundberg, Johnson, and Stewart 2021).

Researchers might also be interested in estimating the first differences or other quantities that are functions of coefficients rather than coefficients themselves. Our proposed methods can be applied not only to coefficients but also to any function of coefficients. We provide such examples in Section 5.2.

3.2 Current Practice: Directly Using Predicted Labels as Variables

In text analyses, a subset of the outcome Y and independent variables $\mathbf{X}$ are based on some forms of text labels, and creating such text-based variables requires text annotation. When using automated text annotation methods to predict text labels, regardless of the exact choice, statistical analyses take the following steps in general. First, researchers check the accuracy of prediction against expert-coded data, e.g., using cross-validation. If the accuracy is "low,"[7] researchers retrain the model until it gets better (e.g., using different LLMs or ML models

[6] Our literature review of the ten political science journals finds that our setup covers common statistical models used in more than 91% of applications using text annotations: Linear regression (49% of applications), Logistic regression (21%), Category proportions over time or across groups (Subgroup means) (19%), and Poisson regression (2%).

[7] Scholars use different criteria for deciding how much is "low" and "high enough," but many scholars use 80 $\sim$ 90% as rough thresholds. In our literature review, the final prediction models researchers chose have the accuracy of 89.8% and the F1 score of 84.6%, on average.

and adding more informative predictors). Then, once the accuracy becomes "high enough," they now use predicted text labels directly in downstream analyses as if those variables were directly observed and not predicted. The idea is that when the prediction accuracy is high, the prediction model sufficiently mimics expert coding, and thus, prediction errors are small enough that they do not affect downstream analyses significantly.

More concretely, most researchers use one of the following two ways to use predicted variables in downstream text analyses. The first approach, which we call LLM-only estimation, is to use LLMs to predict text labels for every document (see Section 2 for different ways to improve LLM-prediction).

LLM-Only Estimation

Step 1: Predict text labels using LLMs for each document.

Step 2: Sample a subset of documents for expert coding.

Step 3: Check the prediction accuracy using the expert-coded data. Repeat Step 1 until the prediction accuracy is high.

Step 4: Use LLM-predicted variables in downstream text analyses.

The second and more classical approach is to use the supervised machine learning model (e.g., random forest, lasso, and so on) to predict text labels. The main steps are essentially the same as those in the LLM-only estimation, and the only difference is the way in which researchers produce predictions (via LLMs or the supervised ML method estimated with the expert-coded data).

Classical Supervised Learning Estimation

Step 1: Sample a subset of documents for expert coding.

Step 2: Train a supervised machine learning model with the expert-coded data.

Step 3: Check the prediction accuracy using the expert-coded data via cross-validation.

Repeat Step 2 until the prediction accuracy is high.

Step 4: Use ML-predicted variables in downstream text analyses.

3.3 The Methodological Challenges of the Current Practice

Ignoring prediction errors in the text annotation step, even if the errors are small, leads to bias, invalid confidence intervals, and wrong p-values in the subsequent statistical analyses of text-based variables. Biases from prediction errors exist even when the prediction accuracy in the text classification step is extremely high, e.g., above 90% or even at 95%. This is because prediction errors are *not completely random*—prediction errors are correlated with observed and unobserved variables we include in downstream analyses. Even small prediction errors can bias downstream analyses in any direction by any amount. Because exactly the same problem applies to the LLM-only estimation and the classical supervised learning estimation, we do not distinguish them, and we discuss prediction errors in general.

To concretely illustrate the problem, we focus on a simple case where the outcome variable Y requires text annotation, and researchers want to regress Y on independent variables $\mathbf{X}$ to estimate coefficients β defined as,

$$\mathbb{E}(Y_i \mid \mathbf{X}_i) \ = \ \mathbf{X}_i^\top \beta. \tag{2}$$

Researchers can easily obtain the ordinary squares estimates of β when the outcome variable of interest Y is observed for every document. However, when Y requires text annotation and Y itself is not observed for each document, researchers instead regress the predicted outcome variable $\widehat{Y}$ on independent variables $\mathbf{X}$. This linear regression with the predicted outcome variable will lead to unbiased coefficient estimation when prediction error, $e_i = \widehat{Y}_i - Y_i$, is zero on average across all different combinations of $\mathbf{X}$.

$$\mathbb{E}(e_i \mid \mathbf{X}_i) \ = \ 0. \tag{3}$$

Even though this expression might seem similar to the standard exogeneity assumption, it turns out that this condition implies much stronger assumptions. Formally, researchers can ignore prediction errors only when prediction errors are completely random, i.e., prediction errors are not affected by the independent variable, the outcome variable, or any unobserved confounder. Unfortunately, this condition is untenable in almost all social science applications. While we focused on one setting where Y is text-based, similar stringent conditions are required when other types of variables (e.g., independent variables) are text-based. We offer additional discussions and the general bias formula in Appendix B.

The literature has explored several approaches to address this problem. First, researchers might consider incorporating bootstrap to capture the uncertainty of the text-prediction step, with the hope of addressing prediction errors. Unfortunately, the central problem of prediction errors is the bias correlated with observed and unobserved variables in downstream analyses and how fast the bias goes to zero asymptotically. Thus, simply adding bootstrap to the current practice cannot eliminate this problem. Second, researchers might make a stringent modeling assumption to explicitly model $\mathbb{E}(e_i \mid \mathbf{X}_i)$ (i.e., how prediction errors vary with $\mathbf{X}$) (e.g., Wang, McCormick, and Leek 2020). This type of method works only when the model for prediction errors is correct and the outcome variable is text-based, while they cannot produce valid confidence intervals or p-values even under the correct model of prediction errors. In contrast, our proposed approach will not make any modeling assumptions about how prediction errors occur.

3.4 Simulation Study

We now use a simulation study to illustrate the problem of ignoring prediction errors. While our method is general, we consider logistic regression with the text-based outcome as a common example. We vary the prediction accuracy of the underlying automated annotation methods—from as low as 50% to as high as 95%—and evaluate how ignoring prediction errors affects

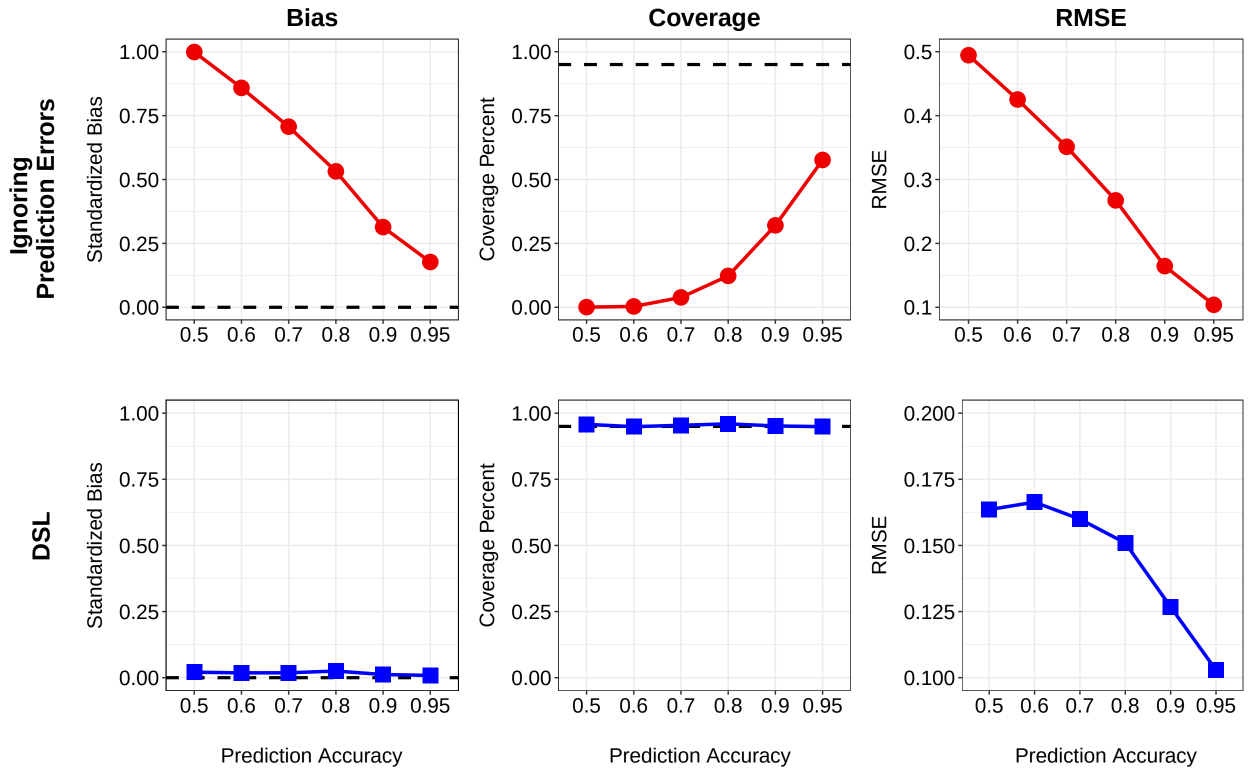


Figure 4: **Ignoring Prediction Errors Lead to Bias and Invalid Confidence Intervals.** *Note*: The first, second, and third columns represent bias, coverage rates of 95% confidence intervals, and root mean squared errors (RMSE), respectively. X-axis shows varying prediction accuracy of the underlying automated text annotation method.

downstream regression analyses. We detail the data generation process in Appendix E.

The first column in Figure 4 shows the average bias across coefficients standardized by the true coefficients. When prediction errors are ignored (the first row), bias decreases as the accuracy of the underlying prediction method goes up. However, bias can be as large as 30% and 18% of the true coefficients even when the underlying prediction accuracy is 90% and 95%. The second column in Figure 4 shows the coverage rate of the 95% confidence intervals (the probability of reported confidence intervals covering the true coefficients), and the coverage rates should be at least 95% if a given estimation method is statistically valid. Unfortunately, when prediction errors are ignored, the coverage rate of 95% confidence intervals is as low as 32% and 58%, even when the underlying prediction accuracy is 90% and 95%. Bias and coverage rates are, of course, much worse when the accuracy of the prediction method is about $80 \sim 90\%$

or lower, as we see in most applications. These results demonstrate that researchers cannot ignore prediction errors even when the underlying prediction method has excellent prediction accuracy.

The second row of Figure 4 previews the results of the proposed DSL. As we show in the next section, DSL is theoretically guaranteed to have asymptotically unbiased estimates and valid confidence intervals, regardless of the accuracy of the underlying prediction method (see the first and second columns in the second row). When the underlying prediction method becomes more accurate, DSL also gets more accurate and has smaller standard errors, which is shown by the reduction in root mean squared error (RMSE) (see the third column in Figure 4). When the accuracy of the underlying prediction method goes up from 50% to 95%, RMSE reduces from 0.16 to 0.10, which is equivalent to a 37.5% reduction in standard errors.

4 Design-based Supervised Learning

We propose a general method, which we call *design-based supervised learning* (DSL), to use predicted variables in downstream analyses without introducing bias from prediction errors. This general framework allows researchers to use LLM annotations or text labels predicted by ML methods in downstream text analyses while maintaining statistical validity.

4.1 Overview

We first provide an overview of the proposed DSL method. While the proposed framework can accommodate various versions of implementations, we first focus on the basic version and then discuss other extensions later.

Design-based Supervised Learning Estimator (DSL)

Step 1: Predict text labels using LLMs for each document.

Step 2: Sample a subset of documents for expert coding.

Step 3: Train an ML model to improve LLM-prediction with the expert-coded data.

Step 4: Combine expert-coded labels and predicted variables in the DSL regression.

Importantly, most steps (Steps 1–3) are similar to existing approaches and thus are already familiar to applied researchers. In the first step, like the LLM-only estimation, we predict text labels using LLMs. In the second step, like the existing approaches, we sample a subset of documents for expert-based coding. In the third step, researchers can use expert-coded documents as the training data and train a supervised machine learning model where we predict the expert-coded labels with predictors that include LLM annotations generated in Step 1 and any other variables that are predictive (e.g., term-document matrices).[8] This step is similar to Step 2 in the classical supervised learning estimation, and the only difference is that users can also incorporate LLM annotations as predictors for the expert-coded labels. The fourth step is an essential defining feature of DSL. We combine expert-based coding and predicted variables in a tailored fashion, which we describe in detail in the next sections.

4.2 Assumption: Design-based Sampling

Before we describe the details of the DSL regression estimator, we clarify the central assumption behind DSL. In particular, we require that researchers know the process through which documents are sampled for expert coding. Formally, we make the following assumption by defining π_i to be the probability of sampling document i for expert coding.

Assumption 1 (Design-based Sampling for Expert Coding)

The probability of sampling documents for expert coding π_i is known to researchers, and π_i is larger than zero for every document.

Assumption 1 holds when the researchers can choose which documents to be coded by

[8] Researchers can also skip this third step, and doing so is equivalent to using the identity function to predict expert-coded labels with LLM labels.

experts. For example, if the researchers have 10000 documents and sample 100 of them to expert-annotate at random, $\pi_i = \frac{100}{10000} = .01$ for all documents. Here, the sampling probability for each document is decided by the researchers and is greater than zero. We also allow more complex stratified or block sampling schemes (i.e., change the sampling probability of documents based on document-level observed covariates) and can cover any case where the sampling probability π_i depends on the LLM annotation, document-level covariates, independent variables, or the outcome variable, as long as π_i is known. This generality is important because researchers might want to over-sample documents that are difficult to annotate. For example, in Fowler et al. (2021), if researchers a priori expect that longer political ads are more difficult to annotate, they can change the sampling probability based on the length of the ads. In Section 5.1.3, we discuss how to determine the required number of expert annotations in each application.

Importantly, Assumption 1 does rule out some applications, and two are worth noting: (1) Researchers use external coding (rather than their own expert coding) to measure text-based variables of interest, and it is unknown why only a subset of documents were coded. (2) Another scenario occurs when researchers need to analyze documents in real-time as soon as they obtain text data, e.g., making polling predictions based on social media posts on election day. In Appendix B.5, we discuss them in detail with examples and explain how researchers can adjust how to apply DSL in these scenarios.

While we do not cover all text-as-data scenarios, our approach covers the vast majority of social science research applications where researchers need to annotate a corpus of documents that are available in total before analyzing data. Even more importantly, Assumption 1 can be guaranteed by research design alone. Our assumption is transparent and easy to justify. This is the reason why our method is named *design-based* supervised learning.

4.2.1 Assumptions We Do Not Make

Understanding assumptions we do not make is as important as understanding the assumptions we do make. In particular, we make *no* assumptions about prediction errors and allow for arbitrary prediction errors. Researchers do not need to assume how prediction errors arise in LLMs or ML prediction. We do not need to assume LLM annotations are unbiased and accurate, or the fitted supervised ML model is correctly specified, unbiased, and accurate. In practice, this means that researchers can use predictions from LLMs or supervised ML without worrying about their prediction errors or inherent biases.

This is in sharp contrast to existing alternatives. Both the LLM-only estimation and the classical supervised learning approach have to assume prediction errors are completely random. This assumption is often severely violated in practice, and most importantly, researchers cannot guarantee this assumption by research design.

4.3 DSL Regression

We now examine how the proposed DSL estimator can incorporate predicted variables without introducing bias under Assumption 1. To provide intuition, we start with a simple case and generalize it step by step.

4.3.1 Building Intuition with Estimation of Category Proportion

Suppose researchers are interested in estimating the proportion of documents belonging to a particular category, e.g., the proportion of political ads attacking opponents. Define $Y_i \in \{0, 1\}$ to denote whether a given political ad attacks opponents. When using the LLM-only estimation or the classical supervised ML methods, users would first predict whether each ad attacks opponents $\widehat{Y}_i$ and then average it over ads to estimate the proportion of attacking ads.

In contrast, DSL uses the following design-adjusted outcome.

$$\widetilde{Y}_i \;=\; \underbrace{\widehat{Y}_i}_{\substack{\text{Predicted}\\ \text{Outcome}}} \;-\; \underbrace{\frac{R_i}{\pi_i}(\widehat{Y}_i - Y_i),}_{\text{Bias-Correction Term}} \tag{4}$$

where Y_i is the outcome of interest coded by experts, R_i is a binary variable taking 1 if document i is expert-coded and 0 otherwise, and π_i (defined in Section 4.2) is the probability of sampling document i for expert coding.[9] This estimator has deep theoretical connections to doubly robust estimation in the causal inference literature (Robins, Rotnitzky, and Zhao 1994; Chernozhukov et al. 2018), and the bias-correction term is similar to the one in the augmented inverse probability weighting estimator.

In the most simple case of random sampling with equal probabilities ($\pi = n/N$ where n is the number of expert-coded documents and N is the total number of documents), the DSL estimator becomes simple.

$$\frac{1}{N}\sum_{i=1}^{N}\widetilde{Y}_i \;=\; \underbrace{\frac{1}{N}\sum_{i=1}^{N}\widehat{Y}_i}_{\substack{\text{Mean of}\\ \text{Predicted Outcomes}}} \;-\; \left(\underbrace{\frac{1}{n}\sum_{i:R_i=1}\widehat{Y}_i}_{\substack{\text{Mean of}\\ \text{Predicted Outcomes}\\ \text{in Labeled Data}}} \;-\; \underbrace{\frac{1}{n}\sum_{i:R_i=1}Y_i}_{\substack{\text{Mean of}\\ \text{Observed Outcomes}\\ \text{in Labeled Data}}}\right) \tag{5}$$

The main idea is to use the expert-coded data to estimate bias from prediction errors (the difference between the second and third terms on the right-hand side), which we subtract from

[9] The design-adjusted outcome is equal to $\widehat{Y}_i$ when $R_i = 0$ and is equal to $\widehat{Y}_i - (\widehat{Y}_i - Y_i)/\pi_i$ when $R_i = 1$. It might be counter-intuitive to change the outcome for documents $R_i = 1$ when the outcome of interest Y_i is observed. Importantly, the goal is not to correct the prediction error at each document level, but at the level of quantities of interest (average in this case, as we show in equation (5) below). In fact, we can only estimate the prediction error from documents with $R_i = 1$, which is used to correct outcomes for documents with $R_i = 0$ on average. More generally, it is usually impossible but also unnecessary to bias-correct outcomes at each document level (Hopkins and King 2010).

the conventional estimator that relies only on predicted labels (the first term on the right-hand side). For example, suppose users have $N = 10000$ ads and randomly sampled $n = 100$ ads for expert annotation. The first term on the right-hand side estimates the proportion of attacking ads by averaging the predicted labels in all $N = 10000$ documents (suppose it is 20%). Because this first term suffers from bias due to prediction errors, the second and third terms on the right-hand side estimate the bias to be subtracted. In particular, the second term estimates the proportion of attacking ads by averaging the predicted labels in $n = 100$ expert-coded documents (suppose it is 18%), and the third term estimates the proportion of attacking ads by averaging the expert-coded labels in $n = 100$ expert-coded documents (suppose it is 10%). Because the expert-coded data are randomly sampled, we can estimate the bias by taking the difference between the second and third terms, $18 - 10 = 8\%$, which we subtract from the first term (i.e., the original naive estimator that only uses predicted labels). In this simple example, the DSL estimate is $20 - (18 - 10) = 12\%$.

4.3.2 DSL Linear Regression

While the previous section focuses on the estimation of category proportions under simple random sampling, the DSL framework can be applied to linear regression as well (under any user-specified sampling strategy). Specifically, the DSL linear regression simply needs to regress the design-adjusted outcome $\widetilde{Y}_i$ (equation (4)) on independent variables $\mathbf{X}_i$. Formally, the DSL regression estimator can be written as

$$\widehat{\beta}_{DSL} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \widetilde{\mathbf{Y}} \tag{6}$$

where $\widetilde{\mathbf{Y}} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_N)$ and ith row of matrix $\mathbf{X}$ is $\mathbf{X}_i$. Under Assumption 1, the DSL estimator is consistent and asymptotically normal when we use cross-fitting (Chernozhukov et al. 2018) to generate predictions.[10] The corresponding confidence intervals can be constructed with the usual standard error formula. Valid statistical inference is possible because the design-adjusted outcomes correct the prediction error on average across all different combinations of $\mathbf{X}$.

$$\mathbb{E}(\widetilde{Y}_i - Y_i \mid \mathbf{X}_i) \;=\; 0. \tag{7}$$

We provide proof of these theoretical properties in Appendix B.

Importantly, the DSL estimator corrects bias only under Assumption 1 without making any assumption about prediction errors in $\widehat{Y}_i$. In practice, this means that researchers can use any LLMs and supervised ML methods to construct the predicted outcomes, even if LLMs and ML methods contain arbitrary prediction errors. While the DSL regression allows for any prediction error, it becomes more accurate (i.e., standard errors are smaller, and confidence intervals are narrower) when the prediction errors are smaller. Therefore, researchers can exploit the recent advances in LLMs and supervised ML methods without sacrificing valid statistical inference, while reducing standard errors as the prediction step becomes more accurate. We illustrated these desirable properties in simulation studies (Section 3.4) and will show more results in empirical applications (Section 5).

4.3.3 Generalization of DSL

Finally, we emphasize that the same general idea applies to a large class of generalized linear models we introduced in Section 3.1 (e.g., logistic, multinomial-logistic, Poisson, and linear fixed-effects regression) and to general cases where any subset of the outcome variable and independent variables are text-based. The only but crucial difference is that we have to bias-correct not the outcome variable itself (as we did in equation (4)) but the underlying moment function. In general, define $m(Y_i, X_i; \beta)$ to be the subgradient of the convex optimization

[10] This step corresponds to Step 3 in the DSL workflow (see Section 4.1). More technical details are in Appendix B.

problem defining a generalized linear model. Then, the moment function for the DSL estimator can be written as

$$m(\widehat{Y}_i, \widehat{X}_i; \beta) - \frac{R_i}{\pi_i}\Big(m(\widehat{Y}_i, \widehat{X}_i; \beta) - m(Y_i, X_i; \beta)\Big) \tag{8}$$

where $\widehat{Y}_i$ and $\widehat{X}_i$ are predictions for the outcome Y_i and independent variables X_i. We provide technical details in Appendix B.

5 Empirical Applications

We now use empirical applications to illustrate how to apply DSL in a wide range of settings. The first application, based on Fowler et al. (2021), considers settings where the outcome variable is text-based, while the second, based on Pan and Chen (2018), examines cases where the independent variables are text-based.

5.1 Text as Outcome: Fowler et al. (2021)

Fowler et al. (2021) examine how and whether the tone of political ads varies across Facebook and television. To test this question, after annotating the tone of ads, the original authors run a linear fixed effects model that regresses the tone of ads on the main independent variable indicating whether a given ad is from Facebook or television, while including candidate-fixed-effects.[11]

In this section, we conduct empirical validation using the expert-coded political ads from Fowler et al. (2021). In particular, we use 13040 expert-coded political ads as the target population of documents ($N = 13040$). But we pretend that we can only sample $n = 1000$ documents for expert coding (less than 8% of the original number of expert coding) and use

[11] We follow the original paper's definition, and the tone of ads is computed by the weighted average of the tone of ads in a given pair of a candidate and a platform where weights are proportional to the expenditure of a given ad.

automated text annotation methods to predict the tone of ads for all the documents. We then assess how well DSL and other methods, which are based on 1000 expert annotations with 13040 predicted labels, can recover the benchmark estimates, which use the entire 13040 expert annotations. By doing so, we can illustrate the use of DSL step by step, while testing how DSL and other methods perform when the underlying automated text annotation methods have non-random prediction errors.

5.1.1 Setup

DSL requires four simple steps. First, we generate LLM annotations for the entire population of documents. As we discussed in Section 2, we here consider six versions: GPT 4, GPT 3.5, and Llama 2 with zero-shot and few-shot learning. In the second step, we randomly sample 1000 documents for expert coding (we will discuss how to determine the number of expert coding in Section 5.1.3).[12] In the third step, using the expert-coded data, we further improve LLM predictions by cross-fitting the generalized random forest (Athey, Tibshirani, and Wager 2019) to predict the expert-coded labels with LLM annotations produced in the first step. Finally, we combine expert-coded labels and predicted labels in the DSL linear fixed-effects regression, where we regress the design-adjusted outcome on the same independent variables used in the original paper, that is, the Facebook dummy variable and candidate-fixed-effects. The main quantity of interest in the original paper is the coefficient of the Facebook dummy variable. Our companion R package `dsl` can implement the third and fourth steps with one function, while taking LLM annotations (Step 1) and expert coding (Step 2) as inputs from users.

We compare DSL against the classical supervised learning approach and the LLM-only estimation. For the classical supervised learning approach, we examine five widely used supervised

[12] In this empirical validation, we rely on expert coding from the original authors, so we simply reveal expert coding for sampled documents.

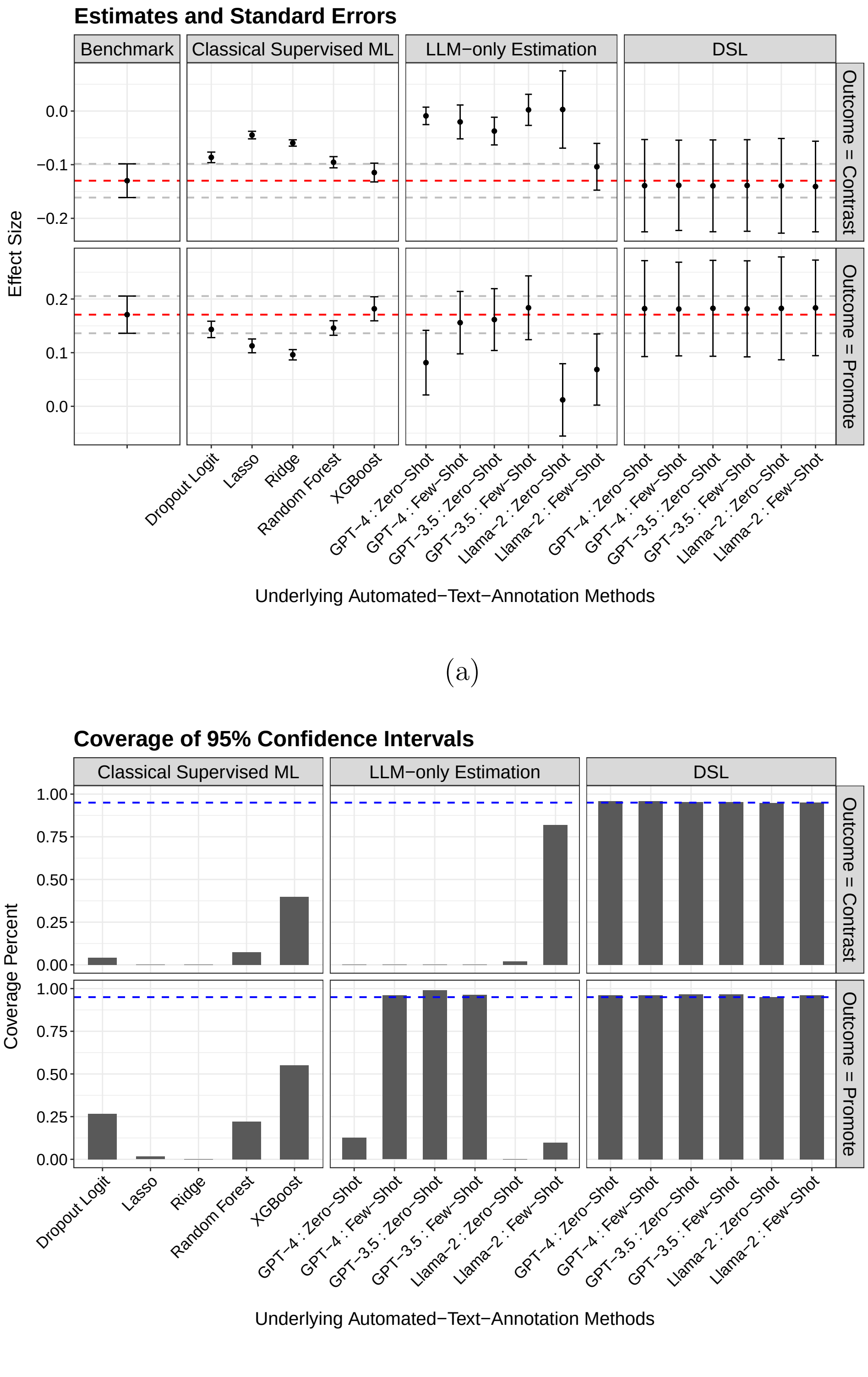
ML methods: drop-out regularized logistic regression (used in the original paper), lasso, ridge, random forest, and XGBoost. We use a set of predictors used in the original paper (more than 7000 variables) that are constructed by processing the ad's text, images, video, and audio. For the LLM-only estimation, we consider the same six versions of LLM annotations. We expect that these existing approaches can provide unbiased estimates with valid confidence intervals only when prediction errors are completely random, while DSL provides valid statistical guarantees even with arbitrary prediction errors.

5.1.2 Results

The results are reported in Figure 5. Due to the space constraints, we focus on two outcomes, "Contrast" and "Promote," in the main text as these two outcomes have the highest and lowest LLM performances, while reporting the results on "Attack" in Appendix G.2. In Figure 5-(a), the leftmost column reports the benchmark estimates based on the entire sample of 13040 expert-coded documents. The remaining columns show point estimates and standard errors of different methods, and the X-axis shows the underlying automated text annotation method. Figure 5-(b) reports coverage rates of the 95% confidence intervals.[13] If a method can produce valid confidence intervals, coverage rates of its 95% confidence interval should be at least 95%.

We now discuss each method in order. First, we look at the LLM-only estimation. Figure 5-(a) shows that point estimates have large variations depending on the underlying LLM method used for automated text annotations. This is because the LLM-only estimation ignores differential prediction errors that each LLM method makes, and as a result, estimates of the quantity of interest are biased. Some methods have small biases for one outcome (e.g., Few-shot learning with Llama 2 when the outcome is "Contrast"), but no method has small biases across

[13] We compute this as the probability of confidence intervals covering the benchmark estimate over 500 repeated sampling of the population of documents and expert coding.



(a)

(b)

Figure 5: **Comparisons of DSL and Existing Approaches using Fowler et al. (2021).** *Note*: In Panel (a), red dotted lines represent point estimates of the "Benchmark" estimates, and gray dotted lines represent their 95% confidence intervals. To show the average performance across random sampling of expert coding, we report the average point estimates and standard errors across 500 repeated sampling. In Panel (b), blue dotted lines represent 95%.

both outcomes. Crucially, in the real-world application where researchers cannot observe the "Benchmark" estimate (unless they expert-code every single document), it is impossible for users to decide which estimates to report and trust. Indeed, depending on which LLMs users choose, they could reach statistically and substantively different results. For example, when researchers use GPT 4 with Few-shot learning, they might conclude the effect on "Contrast" is substantively and statistically indistinguishable from zero, whereas they would find a statistically significant, negative effect when they use Llama 2 with Few-shot learning. Figure 5-(b) shows that confidence intervals based on the LLM-only estimation are, in general, invalid (i.e., cannot cover the benchmark estimates with 95%) due to large biases. In sum, this large variation in estimates is the fundamental problem of ignoring prediction errors: researchers can get statistically and substantively different estimates depending on the choice of LLMs, and there is no way to decide which estimate is the most credible. More generally, the LLM-only estimation has no statistical guarantees in the presence of non-random prediction errors, i.e., some methods had good point estimates for one outcome in this application, but that was a statistical coincidence.

Next, we look at the classical supervised ML method, which has the same problem of ignoring prediction errors. Just like the LLM-only estimation, point estimates have large variations depending on the underlying ML method used for automated text annotation. Importantly, this variation exists even though each supervised ML method has roughly the same prediction performance. Interestingly, estimates from XGBoost have small biases for both outcomes. However, getting a good point estimate is not sufficient in social science analyses, and it is crucial to report a valid uncertainty measure. As we discussed in Section 3, unfortunately, the classical supervised ML method underestimates standard errors, and as a result, it has invalid confidence intervals. Figure 5-(b) shows that, even for "XGBoost," which has good point estimates, the 95% confidence intervals only cover the true effect for about 50%, which in practice means that reported standard errors are severely underestimated and reported p-values are wrong. As discussed in Section 3, these problems cannot be solved by simply adding bootstrap.[14]

Finally, we discuss how the proposed DSL overcomes the shortcomings of the existing methods. Several points are worth emphasizing. First, unlike the existing methods, point estimates of DSL are stable regardless of the underlying automated text annotation methods users choose, and they all have small biases. This property is fundamental in empirical research because researchers do not need to worry that statistical and substantive conclusions might change if they happen to use different LLMs. Therefore, researchers can justify the use of LLMs without assuming that predictions from LLMs are unbiased or accurate. Here, we focus on prediction based on LLMs, but the DSL regression can also be used to correct biases when the automated text annotation is done with the classical supervised ML method. Second, as we see in Figure 5-(b), DSL gives valid standard errors and confidence intervals (i.e., reported confidence intervals have a coverage rate of 95%), unlike the existing methods that significantly underestimate the true uncertainty. Taken together, DSL provides stable, unbiased point estimates and valid confidence intervals regardless of which LLMs researchers use to automate annotations. This is because DSL explicitly takes into account prediction errors through the design-based sampling of expert coding.

Researchers might wonder about the wider confidence intervals of DSL relative to other methods. First, DSL estimators rightly have larger standard errors because they properly take into account prediction errors. In contrast, by ignoring prediction errors, the confidence intervals of the existing methods are invalid and underestimate the true uncertainties. Indeed,

[14] Researchers might wonder about extremely small confidence intervals for the supervised ML method. This is due to both regularization bias and the failure to incorporate prediction uncertainty, which both lead to smaller invalid standard errors.

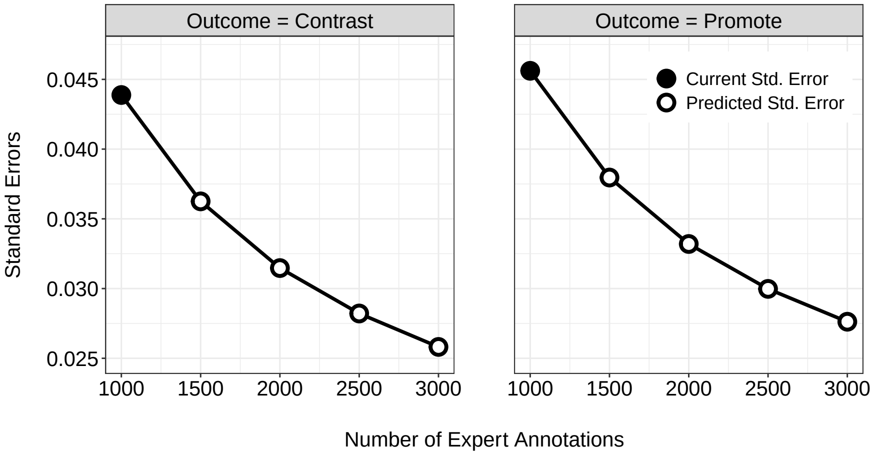


Figure 6: **Power Analysis to Determine the Required Number of Expert Annotations.** *Note*: Each panel reports the current standard errors (1000 expert annotated samples) and predicted standard errors for different numbers of expert annotations. The left and right panels consider DSL analyses when the outcome is "Contrast" and "Promote," respectively.

falsely narrow confidence intervals around biased estimates are exactly what we should avoid: we do not want to be falsely confident about wrong estimates. Second, in practice, researchers can conduct a power analysis to decide the number of expert-coded documents necessary for reducing standard errors of DSL to a certain level, which we discuss next.

5.1.3 Power Analysis

The number of documents experts need to annotate depends on applications. To help researchers in each specific application, we develop a data-driven power analysis: after annotating a small number of documents, we can predict how many more documents researchers need to annotate in order to achieve a user-specified size of standard error.[15] Figure 6 predicts how standard errors reduce as the number of expert annotations increases. For example, as in traditional power analysis, suppose researchers expected a coefficient of the Facebook dummy variable to be -0.08 when the outcome is "Contrast." To detect this effect size with suffi-

[15] Our `R` package `dsl` implements this power analysis with one function.

cient statistical power, scholars ordinarily need standard errors smaller than 0.04. From this figure, researchers can predict that randomly sampling 500 additional documents for expert annotations will reduce the current standard errors from 0.044 to about 0.036.

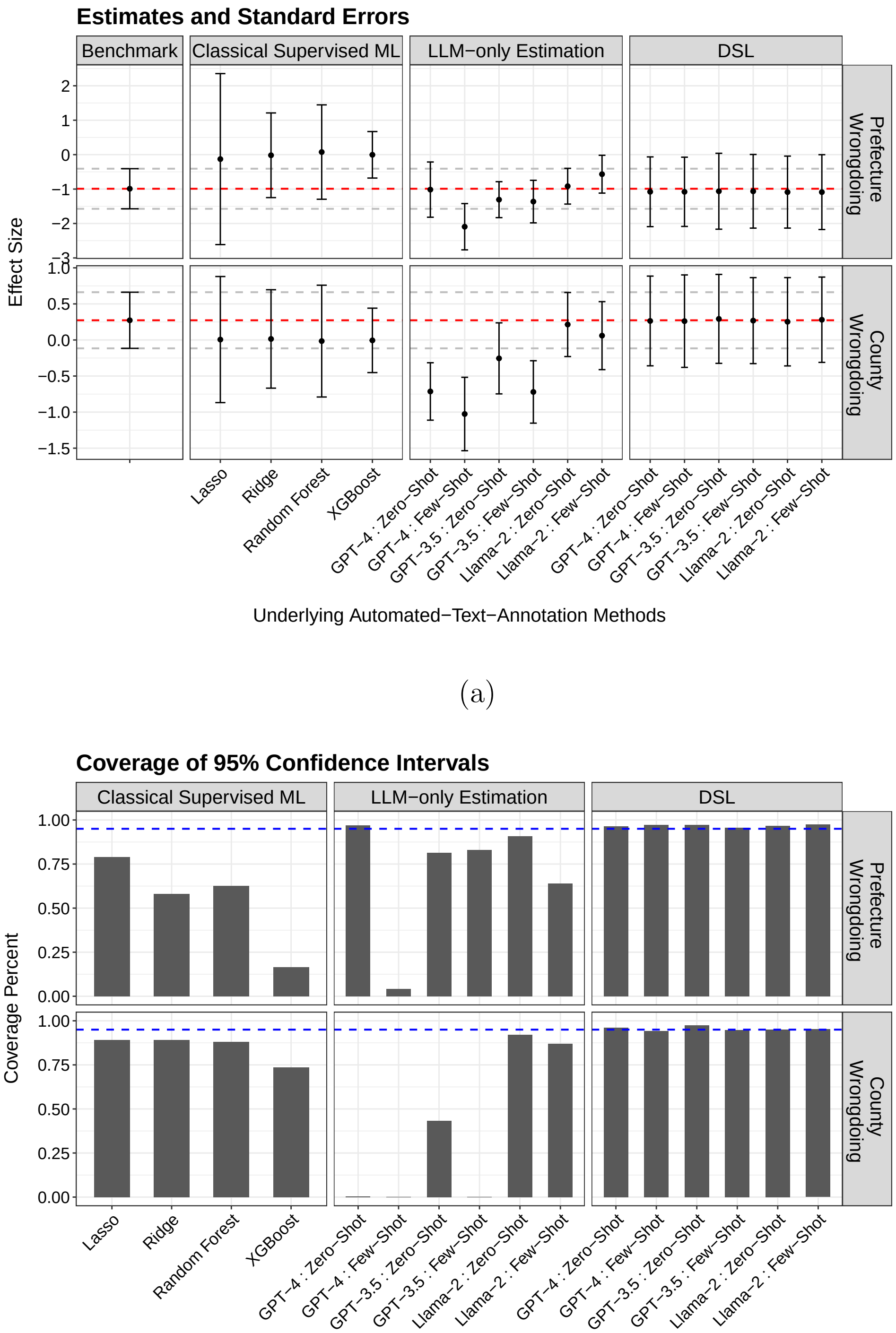
5.2 Text as Independent Variables: Pan and Chen (2018)

We now use Pan and Chen (2018) to consider settings where the independent variables are text-based. This application illustrates the general applicability of our proposed approach: unlike the previous application, documents of interest are written in Chinese, and the original authors use logistic regression as the downstream statistical model.

The key research question in this study asks whether Chinese officials systematically conceal complaints of corruption from upper-level authorities. To test this question, after annotating whether each citizen complaint accuses of prefecture-level or county-level wrongdoing (*Prefecture Wrongdoing* and *County Wrongdoing*), the original authors run a logistic regression that regresses the upward reporting (i.e., whether a given complaint is reported upward to provincial-level officials) on the aforementioned two independent variables (*Prefecture Wrongdoing* and *County Wrongdoing*) and other control variables.

In this section, we again check the performance of DSL and existing methods against the benchmark estimate based on the entire 1412 expert-coded complaints. We pretend that we can only sample $n = 500$ documents for expert coding and use automated text annotation methods to predict *Prefecture Wrongdoing* and *County Wrongdoing* for all the documents. We then assess how well DSL and other methods can recover the benchmark estimates. While the implementation of each method is similar to the one we illustrate in the previous application, we provide all the details in Appendix H.2.

Figure 7 shows estimated coefficients of *Prefecture Wrongdoing* and *County Wrongdoing* as



(a)

(b)

Figure 7: **Comparisons of DSL and Existing Approaches using Pan and Chen (2018).** *Note*: In Panel (a), red dotted lines represent point estimates of the "Benchmark" estimates, and gray dotted lines represent their 95% confidence intervals. To show the average performance across random sampling of expert coding, we report the average point estimates and standard errors across 500 repeated sampling. In Panel (b), blue dotted lines represent 95%.

well as the coverage rates of their 95% confidence intervals.[16] As in the previous application, estimates from the LLM-only estimation are biased, and importantly, substantive and statistical conclusions can flip depending on which LLM users choose for automated text annotation. These variations exist even though the prediction accuracy of different LLMs is roughly similar (see Appendix H). We emphasize that some methods (e.g., Llama 2) happened to have small biases and reasonable coverages in this application, but this is simply a statistical coincidence without any theoretical guarantee. In the real-world application where researchers cannot see the "Benchmark" estimate, it is impossible for users to decide which estimate is the most credible. As in the previous application, estimates from the classical supervised ML approach are heavily biased and have invalid confidence intervals.

In contrast to these existing approaches, DSL is theoretically guaranteed to be asymptotically unbiased and have valid confidence intervals, as we can clearly see in Figure 7. In practice, this means that researchers can get valid statistical estimates regardless of the choice of the underlying automated text annotation methods.

6 Practical Guide

In this section, we provide practical recommendations regarding the most frequently asked questions. We offer additional practical guides in Appendix C, including reporting standards, how to choose LLMs, how to use more complex sampling strategies (e.g., active learning), and what to do if the performance of LLM annotations is excellent or poor.

6.1 Errors in Expert Annotations

In this paper, expert annotation is defined as a procedure that acts as the benchmark against which the quality of the automated text annotation is evaluated. In practice, expert annota-

[16] Appendix H.2 reports results based on the first differences, which reveal the same findings.

tions can also contain errors and mistakes.[17] Completely random errors in expert annotations do not affect the validity of downstream analyses with DSL. DSL standard errors already take into account uncertainties due to random errors in expert annotations because DSL uses heteroskedasticity-robust standard errors. However, in some applications, users might worry that errors in expert annotations can be systematic.

Importantly, concerns about such errors in expert annotations are far from new, and they equally apply to almost all existing text-as-data methods, including any supervised machine learning methods and any unsupervised learning methods that are validated by expert reading of documents (Grimmer and Stewart 2013). Thus, there already exist various strategies and recommendations for handling errors and uncertainties in expert annotations (e.g., Benoit, Laver, and Mikhaylov 2009; Hopkins and King 2010; Mikhaylov, Laver, and Benoit 2012), and researchers can straightforwardly apply them to DSL as well.

Building on this literature, we have three practical recommendations. First, we recommend checking expert annotations with multiple rounds. When LLMs are expected to perform reasonably well, the principal investigators themselves or independent coders can double-check expert coding for all the documents that LLMs and experts disagree on. When there are mul-

[17] We distinguish two scenarios. First, as discussed in the introduction, for some applications, researchers might be concerned that there is no procedure that they can use to validate labels (i.e., even the principal investigators cannot create a codebook and apply it to validate labels by automated methods). This problem is a matter of construct validity and operationalization, and it is best addressed by substantive theories, not by statistical adjustment. Second, even when there is a procedure that the principal investigators wish to implement (e.g., the PIs themselves read every document carefully), experts can make errors or mistakes in applying a given codebook even when they focus on a small number of documents. This second problem is what the following recommendations and existing methods aim to address.

tiple expert coders, the principal investigators can re-annotate any text that has at least one disagreement among experts. These expensive, high-quality annotations are not possible for a large number of documents, but they are extremely valuable even if they are done for a small number of documents. Automated text annotation methods like LLMs can provide the quantity. DSL allows researchers to combine these two complementing annotation methods: high-quality, expensive expert annotations and lower-quality, large-scale automated annotations.

Second, when researchers have multiple expert-coders, they can also empirically evaluate the robustness of statistical estimates to errors in expert annotations. In particular, users can treat annotations from one expert coder as if they were the only expert annotations and check whether and how much DSL estimates change depending on which expert annotations are used. When the intercoder reliability between experts is low and disagreements are systematic, DSL estimates will vary substantially across annotations from different expert coders. In this case, researchers have to re-assess a codebook and disambiguate any systematic disagreement they have in expert annotations. Users can compute the misclassification matrix (e.g., Mikhaylov, Laver, and Benoit 2012) to guide disambiguation steps. When the intercoder reliability between experts is high and disagreements are non-systematic, DSL estimates will be similar across annotations from different expert coders.

Finally, when users can justify modeling assumptions about errors, they can also apply simulation-based methods, such as SIMEX (see, e.g., Hopkins and King 2010; Mikhaylov, Laver, and Benoit 2012), to correct errors in expert annotations. We explain how to combine DSL and SIMEX using Fowler et al. (2021) as an example in Appendix D.

6.2 Limitations

Automated text annotation, in particular, the use of LLM annotations, is a rapidly advancing technology. While we propose a generic method that can incorporate any automated text annotation methods with any prediction error, it might sometimes be possible to derive an application-specific automated text annotation method that can statistically guarantee completely random prediction errors or can model prediction errors. When researchers can justify additional assumptions about how prediction errors arise in the automated text annotation step, they can potentially get smaller standard errors than DSL. DSL is a general-purpose method that is most useful when researchers want to avoid stringent assumptions about prediction errors in automated text annotation methods.

6.3 Wide Applicability

While we so far focused on regression analyses in text-as-data applications, which are the most common downstream analyses, the DSL framework can be used for a broader range of statistical analyses in the social sciences. Our general framework is applicable to any application where researchers use predictive methods to scale up measurements. In Appendix C.5, we discuss how researchers can also apply DSL to (a) estimation of category proportions over time or across groups (e.g., Hopkins and King 2010), (b) causal inference with texts (e.g., Egami et al. 2022), and (c) analyses of a range of unstructured data, e.g., images, audios, videos (e.g., Knox and Lucas 2021; Torres and Cantú 2022).

7 Concluding Remarks

In this paper, we propose a general framework for using recent advances in automated text annotation methods (e.g., LLMs) and, more generally, generative artificial intelligence (AI) in the social sciences. The proposed framework guarantees statistical validity of downstream analyses, without suffering from bias due to unknown non-random prediction errors in AI models.

Due to the recent rapid advances in AI, we can happily expect that new AI models will be developed every month or even faster. This also means that we will continue to have a suite of models that have high predictive performance but lack scientific and theoretical understanding about their prediction errors and various biases (political, racial, gender, social, and so on). However, the existing approaches (i.e., ignoring prediction errors) exactly need to justify how prediction errors arise. Therefore, currently, researchers have to pretend that new AI models have completely random prediction errors, or they have to miss those recent advances. DSL overcomes this tradeoff by incorporating a small number of high-quality, expensive expert annotations. DSL allows users to incorporate any future and current AI models because we do not make any assumptions about prediction errors. With DSL, researchers can always apply state-of-the-art AI models to their social science studies, without worrying that prediction errors and biases in such AI models might invalidate their scientific and statistical conclusions. We hope that this paper provides a foundation for future work considering this exciting intersection of the social sciences, machine learning, and AI.

References

Angelopoulos, Anastasios N., Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. 2023. "Prediction-powered inference." *Science* 382 (6671): 669–674. https://doi.org/10.1126/science.adi6000. https://www.science.org/doi/abs/10.1126/science.adi6000.

Athey, Susan, Julie Tibshirani, and Stefan Wager. 2019. "Generalized Random Forests." *The Annals of Statistics* 47 (2): 1148–1178. https://doi.org/10.1214/18-AOS1709. https://doi.org/10.1214/18-AOS1709.

Barberá, Pablo, Amber E Boydstun, Suzanna Linn, Ryan McMahon, and Jonathan Nagler. 2021. "Automated Text Classification of News Articles: A Practical Guide." *Political Analysis* 29 (1): 19–42.

Bender, Emily M, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency,* 610–623.

Benoit, Kenneth, Michael Laver, and Slava Mikhaylov. 2009. "Treating Words as Data with Rrror: Uncertainty in Text Statements of Policy Positions." *American Journal of Political Science* 53 (2): 495–513.

Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. "On the Opportunities and Risks of Foundation Models." *arXiv preprint arXiv:2108.07258.*

Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. 2018. "Double/Debiased Machine Learning for Treatment and Structural Parameters." *Econometrics Journal* 21:C1–C68.

Egami, Naoki, Christian J. Fong, Justin Grimmer, Margaret E. Roberts, and Brandon M. Stewart. 2022. "How to make causal inferences using texts." *Science Advances* 8 (42): eabg2652. https://doi.org/10.1126/sciadv.abg2652. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.abg2652. https://www.science.org/doi/abs/10.1126/sciadv.abg2652.

Egami, Naoki, Musashi Hinck, Brandon Stewart, and Hanying Wei. 2023. "Using Imperfect Surrogates for Downstream Inference: Design-based Supervised Learning for Social Science Applications of Large Language Models." *Advances in Neural Information Processing Systems* 36.

Fong, Christian, and Matthew Tyler. 2021. "Machine learning predictions as regression covariates." *Political Analysis* 29 (4): 467–484.

Fowler, Erika Franklin, Michael M Franz, Gregory J Martin, Zachary Peskowitz, and Travis N Ridout. 2021. “Political Advertising Online and Offline.” *American Political Science Review* 115 (1): 130–149.

Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli. 2023. “ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks.” *arXiv preprint arXiv:2303.15056.*

Grimmer, Justin, Margaret E Roberts, and Brandon M Stewart. 2022. *Text as data: A new framework for machine learning and the social sciences.* Princeton University Press.

Grimmer, Justin, and Brandon M Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political analysis* 21 (3): 267–297.

Hopkins, Daniel J, and Gary King. 2010. “A Method of Automated Nonparametric Content Analysis for Social Science.” *American Journal of Political Science* 54 (1): 229–247.

Knox, Dean, and Christopher Lucas. 2021. “A Dynamic Model of Speech for the Social Sciences.” *American Political Science Review* 115 (2): 649–666.

Knox, Dean, Christopher Lucas, and Wendy K Tam Cho. 2022. “Testing Causal Theories with Learned Proxies.” *Annual Review of Political Science* 25:419–441.

Linegar, Mitchell, Rafal Kocielnik, and R Michael Alvarez. 2023. “Large Language Models and Political Science.” *Frontiers in Political Science* 5:1257092.

Lundberg, Ian, Rebecca Johnson, and Brandon M Stewart. 2021. “What is Your Estimand? Defining the Target Quantity Connects Statistical Evidence to Theory.” *American Sociological Review* 86 (3): 532–565.

Mikhaylov, Slava, Michael Laver, and Kenneth R Benoit. 2012. "Coder Reliability and Misclassification in the Human Coding of Party Manifestos." *Political Analysis* 20 (1): 78–91.

Mozer, Reagan, and Luke Miratrix. 2023. "Decreasing the Human Coding Burden in Randomized Trials with Text-based Outcomes via Model-Assisted Impact Analysis." *arXiv preprint arXiv:2309.13666.*

Ollion, Etienne, Rubing Shen, Ana Macanovic, and Arnault Chatelain. 2023. "Chatgpt for Text Annotation? Mind the Hype!" *SocArXiv. October* 4.

Ornstein, Joseph T, Elise N Blasingame, and Jake S Truscott. 2022. *How to Train Your Stochastic Parrot: Large Language Models for Political Texts.* Technical report. Working Paper.

Pan, Jennifer, and Kaiping Chen. 2018. "Concealing Corruption: How Chinese Officials Distort Upward Reporting of Online Grievances." *American Political Science Review* 112 (3): 602–620.

Pangakis, Nicholas, Samuel Wolken, and Neil Fasching. 2023. "Automated Annotation with Generative AI Requires Validation." *arXiv preprint arXiv:2306.00176.*

Robins, James M, Andrea Rotnitzky, and Lue Ping Zhao. 1994. "Estimation of Regression Coefficients When Some Regressors Are Not Always Observed." *Journal of the American Statistical Association* 89 (427): 846–866.

Spirling, Arthur. 2023. "Why Open-Source Generative AI Models Are An Ethical Way Forward For Science." *Nature* 616 (7957): 413–413.

Torres, Michelle, and Francisco Cantú. 2022. "Learning to See: Convolutional Neural Networks for the Analysis of Social Science Data." *Political Analysis* 30 (1): 113–131.

Wang, Siruo, Tyler H McCormick, and Jeffrey T Leek. 2020. "Methods for Correcting Inference based on Outcomes Predicted by Machine Learning." *Proceedings of the National Academy of Sciences* 117 (48): 30266–30275.

Zhang, Han. 2021. "How Using Machine Learning Classification as a Variable in Regression Leads to Attenuation Bias and What to Do About It." SocArXiv.

Ziems, Caleb, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. "Can large language models transform computational social science?" *Computational Linguistics* 50 (1): 237–291.

Online Supplementary Appendix:

Using Large Language Model Annotations for the Social Sciences:
A General Framework of Using Predicted Variables
in Downstream Analyses

Contents

A Connection to Literature

This paper builds on several lines of work. First, this paper is motivated by the rapid and fundamental development of LLMs and, more generally, generative artificial intelligence. Over the last couple of years, many papers have shown the incredible potential of LLMs for social science research in a wide range of problems (e.g., Argyle et al. 2023; Linegar, Kocielnik, and Alvarez 2023; Palmer and Spirling 2023; Wu et al. 2023). Among them, one of the most promising and popular use cases is text annotations by LLMs: to name a few papers, Bommasani et al. (2021), Ornstein, Blasingame, and Truscott (2022), Gilardi, Alizadeh, and Kubli (2023), Ollion et al. (2023), Pangakis, Wolken, and Fasching (2023), and Ziems et al. (2024), and this list is growing rapidly. Each paper discusses and evaluates the promise and risks of using LLM annotations in different types of social science applications. All of these papers currently only focus on assessing predictive performance, and no paper discusses how such predicted text labels can be properly used in downstream statistical analyses, which is the central focus of our paper.

This paper draws upon the large literature on double/debiased machine learning and doubly-robust estimation for missing data and causal inference (Robins, Rotnitzky, and Zhao 1994; Chernozhukov et al. 2018; Kennedy 2022). In particular, our doubly robust procedure builds on foundational results on semiparametric inference with missing data (Robins and Rotnitzky 1995; Tsiatis 2006; Rotnitzky and Vansteelandt 2014; Davidian 2022) and the growing literature on doubly robust estimators for surrogate outcomes (Kallus and Mao 2020) and semi-supervised learning (Chakrabortty and Cai 2018; Chakrabortty, Dai, and Tchetgen Tchetgen 2022). Like these papers, we exploit the influence function to derive debiased estimators.

Our paper contributes to the growing literature on the use of predicted variables in statistical analyses. A number of papers develop methods for specific scenarios by making assumptions about the underlying data generating process. For example, Wang, McCormick, and Leek (2020) take into account the predicted outcome by modeling prediction errors, Fong and Tyler (2021) address the predicted independent variables under exclusion restriction, Zhang (2021) relies on a conditional independence assumption about prediction errors, and Knox, Lucas, and Cho (2022) use signed causal diagrams to compute bounds. In contrast to these papers, we only assume that researchers control the sampling process for expert annotations, and we do not make any assumption about the nature of prediction errors, which is particularly difficult to justify in applications of LLMs.

Our paper is most closely related to recent methods that build on the doubly robust estimation to deal with predicted variables. In particular, our paper extends and generalizes methods proposed in Egami et al. (2023). In particular, they only cover cases where the outcome variable requires text annotation and only discuss several models for downstream analyses. By deriving a more general result, we cover cases where any subset of the outcome and independent variables are text-based and accommodate a much wider range of downstream analyses. This methodological generalization is fundamental because about 45% of applications use text-based variables as independent variables, which is not covered in Egami et al. (2023). In addition, we make practical contributions by providing detailed guides on LLM annotations (e.g., how to use LLM annotations in DSL) and expert annotations (e.g., how to determine the required number of expert annotations, and how to handle errors in expert annotations) using two empirical applications.

Theoretically, our paper is also closely related to two recent papers that similarly build on the literature on doubly robust methods. Prediction-powered inference (Angelopoulos et al. 2023) provides a similar framework to ours, but they have primarily focused on settings where the outcome variable is predicted while providing both asymptotic and non-asymptotic confidence intervals. Mozer and Miratrix (2023) focus on settings where the predicted outcome variable is used within randomized experiments. Methodologically, our paper extends these previous results in three ways. First, while these papers only cover cases of text-based outcome variables, we cover cases where any subset of the outcome and independent variables are text-based. This methodological generalization is fundamental because about 45% of applications use text-based variables as independent variables. Second, we develop a data-driven power analysis to help users determine the required number of expert annotations. Third, we derive DSL estimators for a much wider range of downstream analyses popular in the social sciences, including linear fixed effects regression and the instrumental variable method. In addition, we make practical contributions by providing new statistical software and clarifying detailed guides using two empirical applications. Katsumata and Yamauchi (2023) also develop a framework for using predicted variables while building on a different framework of control variates (Chen and Chen 2000).

B Theories of DSL

B.1 Notation and Assumption

Suppose researchers are interested in analyzing N documents. For each document i, we define D_i to be a vector of relevant variables we include in the downstream analyses. For regression problems, $D = (Y, X)$ where Y is the outcome variable and X is a vector of the independent variables. For the mean estimation problem, $D = Y$. For observational causal inference under conditional ignorability, $D = (Y, T, X)$ where Y is the outcome variable, T is the treatment, and X is a vector of observed covariates. For the instrumental variable method, $D = (Y, T, Z, X)$ where Y is the outcome of interest, T is the treatment, Z is the instrument, and X is a vector of observed covariates.

In text-as-data applications, we often cannot observe all relevant variables D for the entire population of documents. We decompose D into two parts $D = (D^{obs}, D^{mis})$ where D^{obs} represents variables that are observed for the entire population of documents and D^{mis} represents variables that are observed only for a subset of documents that are expert-coded. For example, when the outcome variable Y requires text annotation but $X = (X_1, \ldots, X_4)$ are observed for every document, $D^{mis} = Y$ and $D^{obs} = (X_1, X_2, X_3, X_4)$. When Y and X_1 are text-based but the remaining independent variables are observed for every document, $D^{mis} = (Y, X_1)$ and $D^{obs} = (X_2, X_3, X_4)$. This general setup allows for settings where any subset of relevant variables is text-based.

We use Q_i to denote a vector of optional document-level variables that help predict D_i^{mis}. When researchers use LLM annotations as the automated text annotation for D_i^{mis}, those LLM annotations are included in Q_i. When researchers use the classical supervised machine learning method to predict D_i^{mis}, a vector of word frequencies or word embedding is included in Q_i.

Finally, we use $R_i \in \{0, 1\}$ to denote whether document i is sampled for expert annotations. For documents with $R_i = 1$, we observe values for D_i^{mis}, but for documents with $R_i = 0$, values for D_i^{mis} are missing.

The key assumption behind DSL (Assumption 1) is that the probability of sampling documents for expert-coding π_i is decided by researchers, and π_i is larger than zero for every document. Without loss of generality, we can write $\pi_i = \pi(D_i^{obs}, Q_i)$. This notation only assumes that R_i depends on a subset of (D_i^{obs}, Q_i), so it also accommodates more common settings like random sampling where π_i does not depend on any variable or stratified sampling where π_i only depends on a small number of observed variables. Under this design-based sampling, the sampling probability π is known from the research design (i.e., not need to estimate the sampling probability), and we have

$$R_i \perp\!\!\!\perp D_i^{mis} \mid D_i^{obs}, Q_i. \tag{OA.1}$$

B.2 General Results

We first provide proof for a general DSL estimator based on convex objective functions.

Suppose researchers are interested in estimands that can be characterized as the solution to the following convex optimization problem.

$$\beta^* \; := \; \operatorname*{argmin}_{\beta \in \mathbb{R}^d} \mathbb{E}\left\{\ell(D; \beta)\right\} \tag{OA.2}$$

where $\ell(D; \beta)$ is the convex loss function and D represents all the relevant variables in the downstream statistical analyses. This general setup incorporates a wide range of common regression models (see Appendix B.3), such as linear regression with a continuous outcome, logistic regression with a binary outcome, multinomial logistic regression with a categorical outcome, and Poisson regression with a count outcome, as well as linear fixed-effects regression popular in causal inference.

Under mild regularity conditions, convexity allows us to express β^* as the solution to the following estimation equation.

$$\mathbb{E}\left\{m(D;\beta)\right\} \;=\; 0 \tag{OA.3}$$

where $m(D;\beta) \in \mathbb{R}^d$ is a subgradient of the loss function $\ell(D;\beta)$ with respect to β.

If researchers can observe all relevant variables D, they can directly solve the estimation equation to obtain a consistent and asymptotically normal estimator.

$$\widehat{\beta}_{\text{oracle}} \coloneqq \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} \left\| \frac{1}{N}\sum_{i=1}^{N} m(D_i;\beta) \right\|_2^2. \tag{OA.4}$$

However, when some relevant variables are not observed for the entire population of interest, this estimator is infeasible.

DSL can estimate β even when some variables D_i^{mis} are observed only for a subset of expert-coded documents. In general, the moment function for DSL is defined as

$$m_{\text{DSL}}(D_i, Q_i, R_i; \beta, \pi, \widehat{g}) \coloneqq m(D_i^{obs}, \widehat{D}_i^{mis}; \beta) - \frac{R_i}{\pi(D_i^{obs}, Q_i)}\Big(m(D_i^{obs}, \widehat{D}_i^{mis}; \beta) - m(D_i^{obs}, D_i^{mis}; \beta)\Big) \tag{OA.5}$$

where $\widehat{D}_i^{mis} = \widehat{g}(D_i^{obs}, Q_i)$ and $\widehat{g}(\cdot)$ is a estimated supervised machine learning model to predict D_i^{mis} with covariates (D_i^{obs}, Q_i). Using this moment function, the proposed DSL estimator is defined as,

$$\widehat{\beta}_{\text{DSL}} \coloneqq \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} \left\| \frac{1}{N}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} m_{\text{DSL}}(D_i, Q_i, R_i; \beta, \pi, \widehat{g}_k) \right\|_2^2, \tag{OA.6}$$

where we employ a K-fold cross-fitting procedure (Chernozhukov et al. 2018). We first partition the observation indices $i = 1, \ldots, n$ into K groups $\mathcal{L}_k$ where $k = 1, \ldots, K$. We then learn the supervised machine learning model $\widehat{g}_k$ by predicting D_i^{mis} using (D_i^{obs}, Q_i) using expert-coded documents *not* in $\mathcal{L}_k$.

Proposition 1 *Under Assumption 1 and the standard regularity conditions stated below, the cross-fitted DSL estimator (equation* (OA.6)*)* $\widehat{\beta}_{DSL}$ *is consistent and asymptotically normal as sample size* N *goes to infinity.*

$$\sqrt{N}(\widehat{\beta}_{\text{DSL}} - \beta^*) \xrightarrow{d} \mathcal{N}(0, V). \tag{OA.7}$$

where

$$V \;=\; S_V \mathbb{E}(m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g}) m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g})^\top) S_V,$$

$$S_V \;=\; \mathbb{E}\left(\frac{\partial m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g})}{\partial \beta}\right)^{-1}$$

Here we define $\overline{g}$ to be the probability limit of the estimated supervised machine learning function $\widehat{g}_k$ in the sense that for each k, $||\widehat{g}_k - \overline{g}||_2 = o_p(1)$ and $\mathbb{E}_k(||m(L; \beta^*, \widehat{g}_k) - m(L; \beta^*, \overline{g})||_2^2) = o_p(1)$. This probability limit does not need to be equal to the true conditional expectation g^*. Thus, we do not assume the correct specification of the estimated supervised machine learning function.

Proof. In this proof, for the notational simplicity, we use $L_i = (D_i, Q_i, R_i)$ and omit π from the notation of the moment function. That is, we use $m_{\text{DSL}}(L_i; \beta, g)$ to denote the DSL moment function. We also use $\widehat{\beta}$ to denote $\widehat{\beta}_{\text{DSL}}$.

Using the mean value theorem, we can first expand the moment equation around β^*.

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} m_{\text{DSL}}(L_i; \widehat{\beta}, \widehat{g}_k) \;=\; \frac{1}{N}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} m_{\text{DSL}}(L_i; \beta^*, \widehat{g}_k) + (\widehat{\beta} - \beta^*)\frac{1}{N}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} \frac{\partial m_{\text{DSL}}(L_i; \widetilde{\beta}, \widehat{g}_k)}{\partial \beta}$$

where $\widetilde{\beta}$ is a mean value, located between $\widehat{\beta}$ and β^*. For the convex objective function, the first order condition implies that we also have

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} m_{\text{DSL}}(L_i; \widehat{\beta}, \widehat{g}_k) \;=\; 0.$$

Therefore, combining two equations, we have

$$\sqrt{N}(\widehat{\beta} - \beta^*) \;=\; \underbrace{\left(-\frac{1}{N}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} \frac{\partial m_{\text{DSL}}(L_i; \widetilde{\beta}, \widehat{g}_k)}{\partial \beta}\right)^{-1}}_{(a)} \times \underbrace{\frac{1}{\sqrt{N}}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} m_{\text{DSL}}(L_i; \beta^*, \widehat{g}_k)}_{(b)}$$

We will consider terms (a) and (b) in order.

Term (b). We begin with the main term (b), which can be decomposed into three terms.

$$\frac{1}{\sqrt{N}}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} m_{\text{DSL}}(L_i; \beta^*, \widehat{g}_k) \;=\; H_1 + H_2 + H_3$$

where

$$H_1 \;:=\; \frac{1}{\sqrt{N}}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} (m_{\text{DSL}}(L_i; \beta^*, \widehat{g}_k) - \mathbb{E}_k(m_{\text{DSL}}(L_i; \beta^*, \widehat{g}_k))) - (m_{\text{DSL}}(L_i; \beta^*, \overline{g}) - \mathbb{E}_k(m_{\text{DSL}}(L_i; \beta^*, \overline{g})))$$

$$H_2 \;:=\; \frac{1}{\sqrt{N}}\sum_{k=1}^{K}\sum_{i \in \mathcal{L}_k} (m_{\text{DSL}}(L_i; \beta^*, \overline{g}) - \mathbb{E}_k(m_{\text{DSL}}(L_i; \beta^*, \overline{g})))$$

$$H_3 \;\; := \;\; \frac{1}{\sqrt{N}} \sum_{k=1}^{K} N_k \times \mathbb{E}_k(m_{\mathrm{DSL}}(L_i; \beta^*, \widehat{g}_k)).$$

Here we use $\mathbb{E}_k$ to denote the expectation over $\mathcal{L}_k$ conditional on $\mathcal{L}_{-k}$.

H_1 is known as the empirical process term. Given that we use cross-fitting and $\mathbb{E}_k(||m_{\mathrm{DSL}}(L_i; \beta^*, \widehat{g}_k) - m_{\mathrm{DSL}}(L_i; \beta^*, \overline{g})||_2^2) = o_p(1)$, we obtain $H_1 = o_p(1)$ by Lemma 2 of Kennedy, Balakrishnan, and G'Sell (2020).

Next, to examine H_2, we first show that $\mathbb{E}(m_{\mathrm{DSL}}(L_i; \beta^*, \widetilde{g})) = 0$ for any arbitrary fixed function $\widetilde{g}$.

$$\begin{aligned}
& \mathbb{E}(m_{\mathrm{DSL}}(L_i; \beta^*, \widetilde{g})) \\
= \;\; & \mathbb{E}\left(m(D_i^{obs}, \widetilde{g}(D_i^{obs}, Q_i); \beta^*) - \frac{R_i}{\pi(D_i^{obs}, Q_i)}\left(m(D_i^{obs}, \widetilde{g}(D_i^{obs}, Q_i); \beta^*) - m(D_i^{obs}, D_i^{mis}; \beta^*)\right)\right) \\
= \;\; & \mathbb{E}\left(\left(\frac{R_i}{\pi(D_i^{obs}, Q_i)} m(D_i^{obs}, D_i^{mis}; \beta^*) \middle| D_i^{obs}, Q_i\right)\right) \\
& + \mathbb{E}\left(\left(\left(1 - \frac{R_i}{\pi(D_i^{obs}, Q_i)}\right) m(D_i^{obs}, \widetilde{g}(D_i^{obs}, Q_i); \beta^*) \middle| D_i^{obs}, Q_i\right)\right) \\
= \;\; & \mathbb{E}\left(\frac{\mathbb{E}(R_i \mid D_i^{obs}, Q_i)}{\pi(D_i^{obs}, Q_i)} \mathbb{E}\left(m(D_i^{obs}, D_i^{mis}; \beta^*) \mid D_i^{obs}, Q_i\right)\right) \\
& + \mathbb{E}\left(\left(1 - \frac{\mathbb{E}(R_i \mid D_i^{obs}, Q_i)}{\pi(D_i^{obs}, Q_i)}\right) m(D_i^{obs}, \widetilde{g}(D_i^{obs}, Q_i); \beta^*)\right) \\
= \;\; & \mathbb{E}\left(m(D_i^{obs}, D_i^{mis}; \beta^*)\right) \\
= \;\; & 0.
\end{aligned}$$

where the first equality comes from the definition of the DSL moment function, and the second equality comes from the rearrangement of the terms and the law of total expectation. The third equality comes from Assumption 1, i.e., $R_i \perp\!\!\!\perp D_i^{mis} \mid D_i^{obs}, Q_i$, which implies $\mathbb{E}(R_i m(D_i^{obs}, D_i^{mis}; \beta^*) \mid D_i^{obs}, Q_i) = \mathbb{E}(R_i \mid D_i^{obs}, Q_i)\mathbb{E}(m(D_i^{obs}, D_i^{mis}; \beta^*) \mid D_i^{obs}, Q_i)$. The fourth equality comes from the equality that $\mathbb{E}(R_i \mid D_i^{obs}, Q_i) = \Pr(R_i = 1 \mid D_i^{obs}, Q_i) = \pi(D_i^{obs}, Q_i)$ because R_i is a binary variable. Finally, due to convexity of the objective function, $\mathbb{E}(m(D_i, \beta^*)) = 0$.

We also have

$$\begin{aligned}
H_2 \;\; &:= \;\; \frac{1}{\sqrt{N}} \sum_{k=1}^{K} \sum_{i \in \mathcal{L}_k} (m_{\mathrm{DSL}}(L_i; \beta^*, \overline{g}) - \mathbb{E}_k(m_{\mathrm{DSL}}(L_i; \beta^*, \overline{g}))) \\
&= \;\; \frac{1}{\sqrt{N}} \sum_{k=1}^{K} \sum_{i \in \mathcal{L}_k} (m_{\mathrm{DSL}}(L_i; \beta^*, \overline{g}) - \mathbb{E}(m_{\mathrm{DSL}}(L_i; \beta^*, \overline{g}))) \\
&= \;\; \frac{1}{\sqrt{N}} \sum_{i=1}^{N} (m_{\mathrm{DSL}}(L_i; \beta^*, \overline{g}) - \mathbb{E}(m_{\mathrm{DSL}}(L_i; \beta^*, \overline{g})))
\end{aligned}$$

because $\mathbb{E}_k(m_{\text{DSL}}(L_i;\beta^*,\overline{g})) = \mathbb{E}(m_{\text{DSL}}(L_i;\beta^*,\overline{g}))$. Finally, we can use the central limit theorem to show that

$$H_2 \xrightarrow{d} \mathcal{N}(0, \mathbb{E}(m_{\text{DSL}}(L_i;\beta^*,\overline{g})m_{\text{DSL}}(L_i;\beta^*,\overline{g})^\top)) \tag{OA.8}$$

where $\text{Var}(m_{\text{DSL}}(L_i;\beta^*,\overline{g})) = \mathbb{E}(m_{\text{DSL}}(L_i;\beta^*,\overline{g})m_{\text{DSL}}(L_i;\beta^*,\overline{g})^\top)$ as $\mathbb{E}(m_{\text{DSL}}(L_i;\beta^*,\overline{g})) = 0$.

As for H_3, using the similar proof for $\mathbb{E}(m_{\text{DSL}}(L_i;\beta^*,\widetilde{g})) = 0$, we have $\mathbb{E}_k(m_{\text{DSL}}(L_i;\beta^*,\widehat{g}_k)) = 0$ because $\widehat{g}_k$ is a fixed function conditional on $\mathcal{L}_{-k}$. Therefore, $H_3 = 0$.

Taken together, for the main term (b), we obtain

$$\frac{1}{\sqrt{N}}\sum_{k=1}^{K}\sum_{i\in\mathcal{L}_k} m_{\text{DSL}}(L_i;\beta^*,\widehat{g}_k) \;\; \xrightarrow{d} \;\; \mathcal{N}(0, \mathbb{E}(m_{\text{DSL}}(L_i;\beta^*,\overline{g})m_{\text{DSL}}(L_i;\beta^*,\overline{g})^\top)).$$

Term (a). We now consider the term (a), and we need to show that

$$\left(\frac{1}{N}\sum_{k=1}^{K}\sum_{i\in\mathcal{L}_k}\frac{\partial m_{\text{DSL}}(L_i;\widetilde{\beta},\widehat{g}_k)}{\partial\beta}\right)^{-1} \xrightarrow{p} \mathbb{E}\left(\frac{\partial m_{\text{DSL}}(L_i;\beta^*,\overline{g})}{\partial\beta}\right)^{-1} \tag{OA.9}$$

We require the standard regularity conditions that assume the smoothness of the derivative of the moment, which holds true for the most common method of moment estimators we consider here.

Assumption 5 from Chernozhukov et al. (2022). $\mathbb{E}\left(\partial m_{DSL}(L_i;\beta^*,\overline{g})/\partial\beta\right)$ *exists and there is a neighborhood* $\mathcal{N}_\beta$ *of* β^* *such that: (i) for each* k, $||\widehat{g}_k - \overline{g}||_2 = o_p(1)$; *(ii) for all* $||g-\overline{g}||_2$ *small enough,* $m_{DSL}(L;\beta,g)$ *is differentiable in* β *on* $\mathcal{N}_\beta$ *with probability approaching one, and there are* $C > 0$ *and* $\delta(D;g)$ *such that, for* $\beta\in\mathcal{N}_\beta$ *and* $||g-\overline{g}||_2$ *small enough,*

$$\left|\left|\frac{\partial m_{\text{DSL}}(L;\beta,g)}{\partial\beta} - \frac{\partial m_{\text{DSL}}(L;\beta^*,g)}{\partial\beta}\right|\right|_2 \leq \delta(L,g)||\beta-\beta^*||_2^{1/C}; \quad \mathbb{E}(\delta(L,g)) < C.$$

(iii) For each k *and* p *and* q, $\mathbb{E}(\partial m_{DSL}(L;\beta^*,\widehat{g}_k)_p/\partial\beta_q - \partial m_{DSL}(L;\beta^*,\overline{g})_p/\partial\beta_q) = o_p(1)$.

These regularity conditions are standard (Newey and McFadden 1994). Among this regularity condition, the main requirement is that, for each k, $||\widehat{g}_k - \overline{g}||_2 = o_p(1)$. However, we define $\overline{g}$ to be the probability limit of $\widehat{g}_k$, and thus, this automatically holds. Therefore, under this assumption and $\widehat{\beta} - \beta^* = o_p(1)$, we obtain equation (OA.9).

Finally, we combining terms (a) and (b), we have

$$\sqrt{N}(\widehat{\beta}_{\text{DSL}} - \beta^*) \xrightarrow{d} \mathcal{N}(0, V). \tag{OA.10}$$

where

$$V = \mathbb{E}\left(\frac{\partial m_{\text{DSL}}(L_i;\beta^*,\overline{g})}{\partial\beta}\right)^{-1}\mathbb{E}(m_{\text{DSL}}(L_i;\beta^*,\overline{g})m_{\text{DSL}}(L_i;\beta^*,\overline{g})^\top)\mathbb{E}\left(\frac{\partial m_{\text{DSL}}(L_i;\beta^*,\overline{g})}{\partial\beta}\right)^{-1},$$

which completes the proof. □

B.3 Examples

The general theoretical results developed in Appendix B.2 accommodate a wide range of regression problems. To derive a new DSL estimator, researchers just need to derive a corresponding moment function (i.e., a subgradient of a given convex objective function) for each estimator.

For linear regression, the moment function is defined as,

$$\frac{1}{N}\sum_{i=1}^{N} m(D_i;\beta) \coloneqq \frac{1}{N}\sum_{i=1}^{N}(Y_i - X_i^\top \beta)X_i.$$

For logistic regression, the moment function is

$$\frac{1}{N}\sum_{i=1}^{N} m(D_i;\beta) \coloneqq \frac{1}{N}\sum_{i=1}^{N}(Y_i - \mathrm{expit}(X_i^\top \beta))X_i$$

where $\mathrm{expit}(\cdot)$ is the inverse of the logit function.

For multinomial logistic regression, the moment function is

$$\frac{1}{N}\sum_{i=1}^{N} m(D_i;\{\beta\}_{k=1}^{J}) \coloneqq \left\{\frac{1}{N}\sum_{i=1}^{N}(Y_{ik} - \rho_{ik})X_i\right\}_{k=1}^{J-1}$$

where $Y_{ik} \coloneqq \mathbf{1}\{Y_i = k\}$,

$$\rho_{ik} \coloneqq \frac{\exp(X_i^\top \beta_k)}{1+\sum_{k=1}^{J-1}\exp(X_i^\top \beta_k)},$$

and $\beta_J = 0$.

For Poisson regression, the moment function is

$$\frac{1}{N}\sum_{i=1}^{N} m(D_i;\beta) \coloneqq \frac{1}{N}\sum_{i=1}^{N}(Y_i - \exp(X_i^\top \beta))X_i.$$

The DSL framework can also accommodate a variety of observational causal inference methods. For the two-stage least squares estimator used in the instrumental variable design, the moment function is

$$\frac{1}{N}\sum_{i=1}^{N} m(D_i;\beta) \coloneqq \frac{1}{N}\sum_{i=1}^{N}(Y_i - \beta_0 - \beta_1 T_i - X_i^\top \beta_{2:K})(1, Z_i, X_i), \tag{OA.11}$$

where T_i is the treatment, Z_i is the instrument, and X_i is $(K-1)$-dimensional pre-treatment covariates.

For the local linear regression estimator used in the regression discontinuity design, the moment function for an estimator based on observations above the cutpoint is

$$\frac{1}{N}\sum_{i=1}^{N} m_1(D_i;\beta_1)$$

$$:= \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{c < X_i < c+h\} K\left(\frac{X_i - c}{h}\right)(Y_i - \beta_{10} - \beta_{11}(X_i - c))(1, X_i - c),$$

where X_i is the forcing (running) variable, c is the cutpoint, h is the bandwidth, and $K(\cdot)$ is a user-specified kernel function. Similarly, the moment function for an estimator based on observations below the cutpoint is

$$\begin{aligned} & \frac{1}{N}\sum_{i=1}^{N} m_0(D_i; \beta_0) \\ := & \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{c - h < X_i < c\} K\left(\frac{X_i - c}{h}\right)(Y_i - \beta_{00} - \beta_{01}(X_i - c))(1, X_i - c). \end{aligned}$$

The regression discontinuity design estimator is $\widehat{\beta}_{10} - \widehat{\beta}_{00}$.

For linear two-way fixed-effects regression used in the difference-in-differences design, the moment function is

$$\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} m(D_{it}; \beta, \alpha, \gamma) := \begin{pmatrix} \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(Y_{it} - \alpha_i - \gamma_t - X_{it}^\top \beta) X_{it} \\ \left\{\frac{1}{NT}\sum_{t=1}^{T}(Y_{it} - \alpha_i - \gamma_t - X_{it}^\top \beta) X_{it}\right\}_{i=1}^{N} \\ \left\{\frac{1}{NT}\sum_{i=1}^{N}(Y_{it} - \alpha_i - \gamma_t - X_{it}^\top \beta) X_{it}\right\}_{t=1}^{T} \end{pmatrix},$$

where $\{\alpha_i\}_{i=1}^N$ is the individual-fixed-effect and $\{\gamma_t\}_{t=1}^T$ is the time-fixed-effect.

B.4 Power Analysis

Increasing the number of expert annotations is equivalent to increasing the sampling probability π for each document.

From the general results we derived in Appendix B.2, we know the asymptotic variance of the DSL estimator takes the following form.

$$\begin{aligned} V &= S_V \mathbb{E}(m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g}) m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g})^\top) S_V, \\ S_V &= \mathbb{E}\left(\frac{\partial m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g})}{\partial \beta}\right)^{-1} \end{aligned}$$

Based on the definition of the DSL moment, we also have

$$\mathbb{E}\left(\frac{\partial m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g})}{\partial \beta}\right)^{-1} = \mathbb{E}\left(\frac{\partial m(D_i; \beta^*)}{\partial \beta}\right)^{-1}. \qquad \text{(OA.12)}$$

Therefore, the asymptotic variance of the DSL estimator can be re-written as follows.

$$V = \mathbb{E}\left(\frac{\partial m(D; \beta^*)}{\partial \beta}\right)^{-1} \mathbb{E}(m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g}) m_{\text{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g})^\top) \mathbb{E}\left(\frac{\partial m(D; \beta^*)}{\partial \beta}\right)^{-1}.$$

Importantly, the "sandwich" part of the variance $\mathbb{E}\left(\frac{\partial m(D;\beta^*)}{\partial \beta}\right)^{-1}$ only depends on the original moment function and is not dependent on the sampling probability π.

Therefore, increasing the number of expert annotations contributes only to the "meat" part of the variance. We can further decompose the "meat" part of the variance as follows.

$$\begin{aligned}
& \mathbb{E}(m_{\mathrm{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g}) m_{\mathrm{DSL}}(D_i, Q_i, R_i; \beta^*, \pi, \overline{g})^\top) \\
= \ & \mathbb{E}\left(\frac{1}{\pi(D_i^{obs}, Q_i)}\left(m(D_i;\beta^*) - m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*)\right)\left(m(D_i;\beta^*) - m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*)\right)^\top\right) \\
& + \mathbb{E}\left(m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*) m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*)^\top\right) \\
& + \mathbb{E}\left(m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*)\left(m(D_i;\beta^*) - m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*)\right)^\top\right) \\
& + \mathbb{E}\left(\left(m(D_i;\beta^*) - m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*)\right) m(D_i^{obs}, \widehat{D}_i^{mis};\beta^*)^\top\right)
\end{aligned}$$

From this decomposition, we can predict the standard errors under different sampling probabilities by plugging-in different sampling probabilities into the first term. We can consistently estimate this variance under different sampling probabilities.

B.5 Examples of Violations of Assumptions

DSL covers the vast majority of social science research applications where researchers need to annotate a corpus of documents that are available in total before analyzing data. However, it is also important to understand specific examples where Assumption 1 is violated.

Importantly, Assumption 1 does rule out some applications, and two are worth noting: (1) Researchers use external coding (rather than their own expert-coding) to measure text-based variables of interest, and it is unknown why only a subset of documents were coded. For example, Hager and Hilbig (2020) analyze speech documents published by the German government. For 47% of all documents, the topic of the speech is assigned by the German government, but the rest of the documents are published without an explicit topic assignment. In this case, researchers do not decide which document to be sampled for the expert-labeling, and thus, the assumption is violated. However, if researchers can create a codebook, randomly sample documents without an explicit topic assignment, and then label such documents, Assumption 1 holds. With this additional sampling of expert coding, researchers can use DSL by redefining the sampling probability as follows. For documents that was coded by the government, their sampling probability is 1, and for documents that were not originally coded by the government, their sampling probability is n/N_0 where n is the number of expert-coding and N_0 is the number of documents that were not originally coded by the government.

(2) Another scenario occurs when researchers need to analyze documents in real-time as soon as they obtain text data, e.g., making polling predictions based on social media posts on election day. In such cases, it might be inevitable to use expert-coded documents from the past, but in this example, social media posts on election day have the probability of being sampled for expert-coding is zero, and thus, the assumption is violated. However, if researchers have time to sample a subset of social media posts on election day for expert-coding, Assumption 1 holds because now every document they analyze has the probability of being labeled greater than zero. Therefore, when researchers need to collect documents over time, researchers can guarantee Assumption 1 by making sure to sample documents for expert-coding from each time period they analyze.

B.6 Problem of Ignoring Prediction Errors

B.6.1 Example from Section 3.3

We now further investigate the following expression introduced in Section 3.3.

$$\mathbb{E}(e_i \mid \mathbf{X}_i) \ = \ 0. \tag{OA.13}$$

Even though this expression might seem similar to the standard exogeneity assumption, it turns out that this condition implies much stronger assumptions (see a diagram in Figure OA-1). First, prediction errors cannot be affected by any independent variable $\mathbf{X}$ included in downstream analyses. For example, in Fowler et al. (2021) where the original authors included candidate-fixed effects and a platform on which a given political ad is run as $\mathbf{X}$, this condition requires that prediction errors be the same across all candidates and platforms. This requirement might be the most natural one—differential error rates across $\mathbf{X}$ lead to biased estimates of effects of $\mathbf{X}$. Second, prediction errors cannot be affected by the outcome of interest Y, either. For example, in Fowler et al. (2021) where Y is the tone of ads that has three categories (Attack, Contrast, Promote), this condition requires that prediction errors be the same across all categories. This condition is particularly unlikely to hold in most applications of text analyses in the social sciences because most prediction approaches (LLMs or supervised ML methods) tend to have higher prediction errors for rare categories.[1] Finally, prediction errors cannot be affected by unobserved confounders U, either. This is the case even when researchers only estimate coefficients β for descriptive analyses and do not make explicit causal claims. Unfortunately, it is not sufficient to check relationships between prediction errors

[1] We find differential prediction errors across categories in LLM annotations for Fowler et al. (2021), which we report in Appendix G.

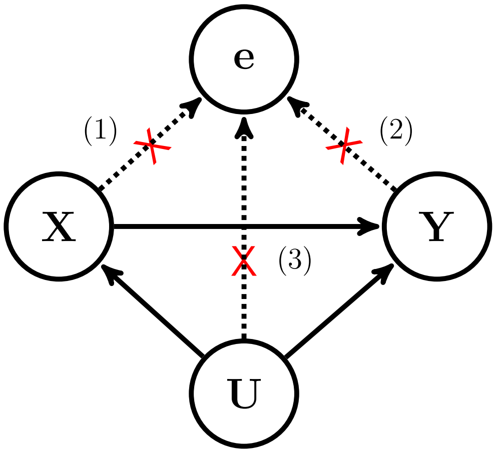


Figure OA-1: **Conditions Required to Ignore Prediction Errors.**
Note: Prediction errors e can be ignored only when e is uncorrelated with independent variables $\mathbf{X}$. This implies that (1) prediction errors e are not affected by independent variables $\mathbf{X}$, (2) prediction errors e are not affected by the outcome variable Y (to block the path $X \rightarrow Y \rightarrow e$), and (3) prediction errors e are not affected by unobserved confounders U (to block the path $X \leftarrow U \rightarrow e$).

and the main variables in the downstream analyses (the outcome and the independent variables). To ignore prediction errors, researchers also have to justify that prediction errors are unrelated to any unmeasured confounder, which is extremely difficult in most applications given that researchers often have limited information about unmeasured confounders.

Therefore, researchers can ignore prediction errors only when prediction errors are completely random, i.e., prediction errors are not affected by the independent variable, the outcome variable, or any unobserved confounder. Unfortunately, this condition is untenable in almost all social science applications. While we focused on one setting where Y is text-based, similar stringent conditions are required when other types of variables (e.g., independent variables) are text-based.

B.6.2 General Bias Formula

While we derived a condition required to ignore prediction errors above, we primarily focused on cases when Y_i is text-based. Here, using the general framework developed in Appendix B, we consider general conditions that apply to any convex optimization problem and cases where any subset of outcome and independent variables are text-based.

In general, to ignore prediction errors, researchers need to assume

$$\mathbb{E}\left(m(D_i^{obs}, \widehat{D}_i^{mis}; \beta) - m(D_i^{obs}, D_i^{mis}; \beta)\right) = 0, \tag{OA.14}$$

which means that the moment function that plugs in predicted variables is unbiased. In a special case when outcome Y_i is text based, this reduces to the bias condition derived in the main paper.

$$
\begin{aligned}
& \mathbb{E}\left(m(D_i^{obs}, \widehat{D}_i^{mis}; \beta) - m(D_i^{obs}, D_i^{mis}; \beta)\right) \\
= \; & \mathbb{E}\left(m(\widehat{Y}_i, X_i; \beta) - m(Y_i, X_i; \beta)\right) \\
= \; & \mathbb{E}\left(X_i(\widehat{Y}_i - X_i^\top \beta) - X_i(Y_i - X_i^\top \beta)\right) \\
= \; & \mathbb{E}\left(X_i(\widehat{Y}_i - Y_i)\right)
\end{aligned}
$$

which is equal to zero when $\mathbb{E}(\widehat{Y}_i - Y_i \mid X_i) = 0$.

C Practical Guide

In this section, we summarize our practical recommendations in each step of DSL.

C.1 Step 1: Predict Text Labels with LLMs

In the first step, researchers use LLMs to predict text labels for the entire population of documents. See our examples of prompts in Section 2 and Appendix F.2.

C.1.1 Which LLMs should we use?

Researchers can often start with zero-shot learning (i.e., no exemplar) using the state-of-the-art LLM. We also recommend implementing at least one open-source LLM like Llama-2. Researchers can also consider few-shot learning (i.e., adding exemplars). Note that specifics of LLM implementations are likely to evolve quickly over time given the speed of the LLM development.

More importantly, researchers do not need to choose one specific LLM in the proposed DSL framework. If researchers have multiple high-performing LLM annotations, they can incorporate all of them in DSL estimators. This is because DSL only uses LLM annotations as predictors for expert-coded labels, and we do not make any assumptions about errors in LLM annotations.

C.1.2 What if LLM annotations are Very Good?

If LLM annotations have extremely high predictive performance, DSL is going to have small standard errors because bias-correction terms are small (the second part of equation (4) is close to zero), while maintaining statistical validity. However, as long as LLM annotations are not perfect, even if they have excellent performance, the LLM-only estimation is not statistically valid (as seen in our applications and simulations). So, DSL is preferred to the LLM-only estimation even when LLM annotations have high predictive performance.[2]

[2] In a hypothetical scenario when LLM annotations have no error at all, DSL is going to be the same as the LLM-only estimation.

C.1.3 What if LLM annotations are Very Bad?

If LLM annotations have extremely low predictive performance, DSL is going to have large standard errors because bias-correction terms are large, even though it will still maintain statistical validity. In this case, it is recommended to retrain the automated text annotation step (e.g., using different LLMs or training the classical supervised machine learning method), which is the same advice as in the classical supervised learning literature. Using low-quality LLM annotations will not invalidate the statistical properties of DSL, but having higher performing automated text annotation will reduce standard errors and the required number of expert annotations.

C.1.4 Should we spend time on improving LLMs or increasing the number of expert annotations if we want to reduce standard errors?

In general, we recommend researchers should spend time on increasing the number of expert annotations as long as the predictive performance of the underlying automated text annotation method is reasonably high (around $80 \sim 90\%$, as in our examples in Section 2). When the predictive performance becomes moderately high, it is often difficult to further improve the predictive accuracy or F1 scores by more than 5 percentage points. Standard errors of DSL do not often reduce much even if the predictive accuracy of the underlying text annotation methods improves by 1 or 2 percentage points. In contrast, increasing the number of expert annotations is theoretically guaranteed to reduce standard errors of the downstream analyses. Researchers can also use a power analysis to explicitly predict how much standard errors will decrease by adding a certain number of expert annotations (see Section 5.1.3).

C.2 Step 2: Sample Documents for Expert Annotation

In the second step, we sample a subset of documents for expert annotations. See Section 6 of the main texts for recommendations about how to handle errors in expert annotations.

C.2.1 Required Number of Expert Annotations

How many documents experts need to annotate depends on applications. To answer this question in each specific application, we develop a data-driven power analysis: after annotating a small number of documents, users can predict how many more documents they need to annotate in order to achieve a user-specified size of standard error. For example, as in traditional power analysis, suppose researchers expect the main treatment effect to be about 4 percentage points and would like to design their study such that standard errors are about 2 percentage points in order to detect the expected effect size with a conventional threshold for statistical significance. In this case, for instance, after annotating 250 political ads, our data-driven power analysis can predict how many additional expert-annotated documents are required to make standard errors below 2 percentage points. See our empirical application in Section 5.1.3.

C.2.2 Construct Validity and Prediction Errors

In this paper, we developed a method to account for prediction errors, which is the discrepancy between expert annotations and automated text annotations. An equally important problem is the construct validity, which is a question about a mapping between a theoretical concept of interest and expert annotations. The proposed method can only account for prediction errors, and this does *not* replace careful, theoretical considerations about how operationalization in a user-specified codebook relates to the main theoretical concept. Rather, our method *augments* theoretical thinking about the construct validity by allowing researchers to focus on expert annotations and removing any additional error from the use of automated text annotation.

C.2.3 How to Sample Documents for Expert Annotations

DSL can allow for any sampling method for expert annotations as long as it is controlled by researchers, including the most common random sampling and any sampling method that depends on document-level observed covariates. In practice, researchers can often start with random sampling with equal probabilities. If researchers wish to over-sample documents that are difficult to annotate, one possible approach is to increase the sampling probability for documents that LLMs are more uncertain about (e.g., Li et al. 2023). Active learning is also a promising area of research to improve text sampling in the classical supervised learning settings (Bosley et al. 2022).

C.2.4 Active Learning

Active learning is a technique to adaptively choose documents for expert-coding. This technique is important when researchers use the classical supervised learning method (Bosley et al. 2022). However, the use of active learning for valid statistical inference is known to be challenging. The existing approaches assume away any prediction errors and uncertainties in the first step of the supervised learning step and also ignore the fact that expert-coded data are adaptively collected, which also makes the standard statistical inference invalid. Fortunately, when researchers use pre-trained LLMs (e.g., GPTs) as the automated text annotation methods, researchers can easily apply the idea of active learning within the DSL framework. We follow the idea developed in Li et al. (2023). In particular, researchers can first use multiple pre-trained LLMs—they can come from different LLMs models or different prompts with the same model— and then obtain the inter-LLM agreement for each document. This agreement score captures the difficulty of labeling a particular document. Researchers can use this agreement score to change the sampling probability for expert coding. For documents whose label multiple LLMs disagree on, we can increase its probability, and for documents whose label multiple LLMs agree on, we can decrease its probability. The key is that many of the current successful LLMs are pre-trained and we do not need to fit them to the data. Therefore, this LLM agreement score can be analyzed simply as one of the observed document-level variable. Therefore, without any additional complex statistical theory, researchers can systematically change the sampling probability for expert coding based the LLM agreement score.

C.3 Step 3: Train an ML model to further improve LLM-prediction

In this third step, we fit a supervised ML model via cross-fitting where we predict the expert-coded labels with predictors that include LLM annotations generated in Step 1 and any other variables that are predictive (e.g., term-document matrices). This step helps to calibrate LLM annotations using the expert-coded labels (similar to fine-tuning in LLMs). Our companion `R` package `dsl` can implement this third and the next fourth steps with one function.

DSL estimator does not assume the correct specification of the underlying prediction model. Thus, DSL estimates are consistent and have valid confidence intervals regardless of the choice of ML models.[3] This step is practically important because if researchers have multiple high-performing LLMs, they can include all of them as predictors in this step.

C.4 Step 4: Fitting DSL Regression

The final step combines the expert annotations and automated annotations within the DSL framework. Researchers can apply the DSL framework to a large class of generalized linear models commonly used in social science applications (e.g., linear, logistic, multinomial-logistic, Poisson, and linear fixed-effects regression). Our framework also accommodates settings where any subset of the outcome variable and independent variables are text-based.

C.5 Wide Applicability

While we so far focused on regression analyses in text-as-data applications, which are the most common downstream analyses, the DSL framework can be used for a broader range of statistical analyses in the social sciences. Our general framework is applicable to any application where researchers use predictive methods to scale up measurements.

[3] In our companion `R` package `dsl`, we use random forest as the default method, while users can always confirm that their results are indeed stable when changing ML models, as our theoretical results imply.

C.5.1 Estimation of Category Proportions over Time or across Groups

Many scholars are interested in estimating the proportion of all documents in each user-specified category (e.g., Hopkins and King 2010; Keith and O'Connor 2018; Card and Smith 2018; Jerzak, King, and Strezhnev 2023). For example, we might study how the proportion of censored documents changes over time or how the proportion of social media posts containing hate speech differs across groups, such as Democrats and Republicans.

These questions can be analyzed within the DSL framework, too. Specifically, researchers can regress a text category on time or groups. For example, using whether a document is censored as the outcome and time indicators as the independent variable in the DSL linear regression, researchers can estimate how the proportion of censored documents changes over time. When users include time-fixed effects, this estimation method is non-parametric and is equivalent to estimating the proportion of censored documents in each time period separately.

C.5.2 Causal Inference with Texts

An increasing number of scholars make causal inference with textual data (Fong and Grimmer 2021; Egami et al. 2022; Feder et al. 2022; Mozer and Miratrix 2023). DSL can be used in causal inference applications where the outcome, treatment, or confounders are text-based. In randomized experiments, researchers can use the DSL regression to perform the difference-in-means or covariate-adjusted linear regression for estimating the average treatment effect.[4] In observational studies, under corresponding causal identification assumptions, researchers can apply the DSL two-stage-least squares for the instrumental variable design, the DSL local linear regression for the regression discontinuity design, and the DSL two-way fixed effects estimator for the difference-in-differences design.

C.5.3 Statistical Analyses of Unstructured Data, e.g., Images, Audios, Videos

Social scientists have begun to utilize a wider range of new data sources, such as image, audio, and video (e.g., Knox and Lucas 2021; Torres and Cantú 2022; Tarr, Hwang, and Imai 2023). Like the text-as-data literature, researchers often use medium-specific automated annotation methods (e.g., convolutional neural networks, and recent foundation models) before analyzing such annotated data in the main downstream statistical analyses. However, as in the automated text annotation,

[4] Previous studies (e.g., Fong and Grimmer 2021; Egami et al. 2022) have clarified the challenges of inferring a codebook and causal estimates from the same data, especially when using unsupervised learning approaches. In contrast, this paper relies on a supervised learning framework where a codebook is given by researchers rather than estimated by a model.

they inevitably contain non-random prediction errors. DSL can be used directly to handle such prediction errors in image, audio, and video annotation tasks as well.

D Errors in Expert Annotations

In this paper, we define expert annotation to be a procedure that acts as the benchmark against which the quality of the automated text annotation is evaluated. We do not require "human" experts to provide the benchmark. We only assume that there is a procedure that researchers want to use as the benchmark.[5] Completely random errors in expert annotations do not affect the validity of downstream analyses with DSL. However, in some applications, users might worry that errors in expert annotations can be systematic.

As we emphasized in the main paper, concerns of such errors in expert annotations are far from new, and they equally apply to almost all existing text-as-data methods, including any supervised machine learning methods and any unsupervised learning methods that are validated by expert reading of documents (Grimmer and Stewart 2013). The DSL estimator is not more sensitive to errors in expert annotations than other methods. Importantly, the problem of prediction errors, which we focus on in this paper, is independent of the problem of errors in expert annotations. Thus, researchers can combine any method to deal with errors in expert annotations with the DSL estimator. There already exist various strategies and recommendations for handling errors and uncertainties in expert annotations (e.g., Benoit, Laver, and Mikhaylov 2009; Hopkins and King 2010; Mikhaylov, Laver, and Benoit 2012), which we discuss more in this section.

We have already discussed our two main recommendations in the main paper: (1) checking expert annotations with multiple rounds and (2) empirically evaluating the robustness of statistical estimates to errors in expert annotations by using multiple coders. Please see Section 6 of the main paper. If researchers believe that these two strategies are not sufficient, they can consider a model-based strategy to deal with measurement errors. We discuss one popular method below, but it is important to note that this method (and all the other similar model-based measurement error methods) works only when the assumption about how measurement errors arise is correct, so researchers should use it only when they believe the required assumption is warranted.

D.1 SIMEX

The simulation-extrapolation method, known as SIMEX, is a popular, model-based strategy to handle measurement errors (e.g., Küchenhoff, Mwalili, and Lesaffre 2006; Hopkins and King 2010).

[5] If researchers believe that there is no procedure that they can use as the benchmark, it is no longer the "supervised" learning problem.

The main idea of this method is to exploit the relationship between the size of misclassification and bias in estimated coefficients. In particular, given the misclassification matrix, the method simulates data with higher misclassification and extrapolates back to the case of no misclassification. This method requires assumptions about how measurement errors occur in expert annotations (explained below). When such assumptions are plausible in a given application, researchers can combine the DSL estimator to account for prediction errors and the SIMEX method to account for additional errors in expert annotations.

Here, we briefly introduce the SIMEX for comprehensiveness, but for a more detailed introduction, we refer readers to Küchenhoff, Mwalili, and Lesaffre (2006) and Hopkins and King (2010). For the sake of simplicity, following Küchenhoff, Mwalili, and Lesaffre (2006), we focus on a binary label, which can be used as the outcome or independent variable, but the same general idea can be used for a multi-class label and a continuous value as well. Suppose D is the correctly measured variable and $D^\dagger$ is an observed variable with measurement errors. In general, the misclassification matrix M captures the patterns of misclassification.

$$M_{jk} = \Pr(D^\dagger = j \mid D = k)$$

where M is a 2×2 matrix when D is a binary label, and more generally, a $q \times q$ matrix when the number of categories in D is q. M_{jk} captures the probability of observing j when the correctly measured variable is k. For example, M_{11} captures the probability of observing 1 when the correct label is also 1, and M_{01} captures the probability of observing 0 when the correct label is 1.

For SIMEX, we define the function ($\lambda \geq 0$),

$$\lambda \rightarrow \beta(M^\lambda) \tag{OA.15}$$

where M^λ is the misclassification matrix with λ times the measurement error in the original data and $\beta(\cdot)$ is a quantity of interest, which is a function of the misclassification matrix. This function captures how much the quantity of interest β changes as we increase the amount of measurement errors in expert annotations. An estimate based on the observed data is $\widehat{\beta}(M^1)$, and the SIMEX tries to obtain $\widehat{\beta}(M^0)$.

The SIMEX method aims to estimate this function by simulating data with higher measurement errors. Specifically, we can construct data sets with M^2 by applying the misclassification matrix M to observed variables $D^\dagger$ and generate $D^{\dagger\dagger}$, we can construct data sets with M^3 by applying the misclassification matrix M to $D^{\dagger\dagger}$ and generate $D^{\dagger\dagger\dagger}$, and so on. By simulating data sets with higher measurement errors and applying an estimation method of interest (in our case, DSL), we can observe how estimates $\widehat{\beta}$ change as a function of λ where we can vary $\lambda \in \{1, 2, 3, \ldots, \}$.

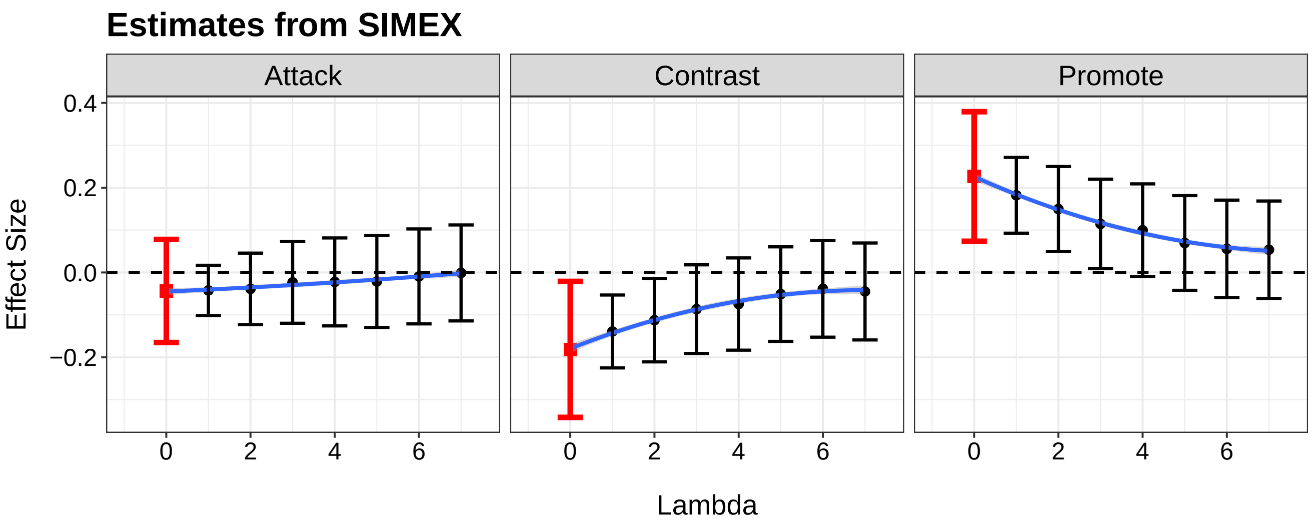


Figure OA-2: **SIMEX estimates.**

The SIMEX method typically uses the second-order polynomial or spline methods to estimate the relationship between λ and $\beta(M^{\lambda})$. Then, the final extrapolation step in the SIMEX estimator is to plug-in $\lambda = 0$. Importantly, this estimator works only when the misclassification matrix is correctly specified and the functional approximation (equation (OA.15)) is correctly specified.

To illustrate how researchers can incorporate this into DSL, we apply the SIMEX method to our empirical application based on Fowler et al. (2021). Figure OA-2 shows the results where the red square dot and its interval represent a point estimate and the 95% confidence intervals for the estimator combining the DSL and SIMEX. Estimates under $\lambda = 1$ are the DSL estimates without correction of expert annotation errors. Estimates under $\lambda = 2$ are the DSL estimates when data sets contain twice the amount of measurement error in the original data, and so on. Blue smooth lines represent the estimated functional approximation for equation (OA.15). Importantly, all black estimates ($\lambda \geq 1$) are only used as intermediate steps to obtain the SIMEX estimate ($\lambda = 0$), so we focus our interpretation on the estimates with $\lambda = 0$. Standard errors are calculated based on bootstrap as recommended in Küchenhoff, Mwalili, and Lesaffre (2006). In this illustration, we use the misclassification rate ($M_{11} = 0.9, M_{01} = 0.1, M_{10} = 0.1, M_{00} = 0.9$) based on the intercoder reliability measure reported in Fowler et al. (2021) and in the Wesleyan Media Project. Interestingly, for all the outcomes, if researchers correct for errors in expert annotations, point estimates become larger, while standard errors are larger due to the additional uncertainty of the SIMEX estimation. For the outcome “Attack”, estimates are relatively insensitive to the misclassification, and therefore, the SIMEX estimate (when $\lambda = 0$) and the original estimate (when $\lambda = 1$) are similar. In contrast, for the outcomes “Contrast” and “Promote”, the absolute values of estimates are larger after correcting errors in expert annotations by SIMEX.

E Simulation Studies

Here we offer the details of the simulation study we reported in Section 3.4. The main purpose of this simulation is to clearly show that ignoring prediction errors can bias downstream estimates and standard errors. We also use empirical applications in Section 5 to show the problem of ignoring prediction errors and how DSL solves the issue under realistic real-world social science data settings.

As the entire population, we generate $n = 5000$ i.i.d. observations ($i \in \{1, \dots, 5000\}$) as follows.

- Covariates: $X_i \sim \mathcal{N}(\overrightarrow{0}, \Sigma^X)$ where $X_i = (X_{i1}, \dots, X_{i10})$, and $\overrightarrow{0}$ is a vector of 0 with length 10. For $i \in \{1, \dots, 10\}$, $\Sigma^X_{ii} = 1$ and for $i \neq j$, $\Sigma^X_{ij} = 0.3$. For the second covariate, we update it by binarizing $X_{i2} = \mathbf{1}\{X_{i2} > \text{qnorm}(0.8)\}$.

- Binary Outcome: $Y_i \sim \text{Bernoulli}(\text{expit}(W_i))$ where

  $$W_i = \frac{0.1}{1 + \exp(0.5X_{i3} - 0.5X_{i2})} + \frac{1.3X_{i4}}{1 + \exp(-0.1X_{i2})} + 1.5X_{i4}X_{i6} + 0.5X_{i1}X_{i2} + 1.3X_{i1} + X_{i2}$$

  This data-generating process is similar to the one in (Vansteelandt and Dukes 2022). It contains various nonlinear transformation of X and it is difficult to correctly model the outcome function.

- Prediction by the automated text annotation method: $\widehat{Y}_i = P_i Y_i + (1 - P_i)(1 - Y_i)$ where $P_i \sim \text{Bernoulli}(P_q)$ and P_q controls the accuracy of the prediction. When $P_q = 0.9$, $\widehat{Y}_i = Y_i$ with 90% and $\widehat{Y}_i = 1 - Y_i$ with 10%. We vary P_q in our simulation.

  To evaluate the general statistical behavior, here we do not use a specific automated text annotation method. Rather, by directly controlling the amount of prediction errors, we can understand how prediction errors, in general, affect downstream analyses. We used a very simple flipping error as used in Clayton et al. (2023). The realistic prediction errors, as we examine in our empirical applications in Section 5, are more likely to be complex. The main idea here is to show that bias from prediction errors are substantial even for this simple prediction errors.

- Expert Annotation: We use simple random sampling of 500 documents for expert annotation. Thus, $\Pr(R_i = 1) = 0.1$ for every observation.

Our estimand of interest is the coefficients of the oracle logistic regression where we regress Y_i on $(X_{i1}, X_{i1}^2, X_{i2}, X_{i4})$. We evaluated bias, coverage, RMSE of the estimator ignoring prediction errors and DSL in Figure 1.

We found that estimators ignoring prediction errors have large biases and invalid confidence intervals, which makes it unsuitable for social science downstream analyses. DSL has low bias and proper coverage of confidence intervals regardless of the accuracy of the underlying automated text annotation method.

F Introduction to LLMs

F.1 A Simple Guide to LLM Usage

To help readers incorporate LLMs into their research workflow, we include a simple guide on how to use LLMs to generate annotations. Note that this guide was written in early 2024; at the time that users are reading, various details may have changed. Our goal is to provide a helpful starting point.

F.1.1 Primer on Large Language Models

We begin with a high-level explanation of how LLM annotation works. Language models assign conditional probabilities over sequences of tokens (a token is basically a word or word fragment): $\Pr(x_n \mid \langle x_1, ..., x_{n-1}\rangle)$. These conditional probabilities are used autoregressively to generate tokens, successively appending each new token to the sequence and then predicting the next one.

In concrete terms, suppose we begin with the sentence "once upon", tokenized as $\langle x_1 =$ once, $x_2 =$ upon$\rangle$. A language model gives the probability of the next token, x_3, conditional on the sequence so far:

$$\Pr(x_3, \mid \langle x_1 = \text{once},\ x_2 = \text{upon}\rangle).$$

Suppose that the language model returns that the most likely next token corresponds to the word "a". We set $x_3 =$ a, and then proceed to calculate the probabilities for the next token in the sequence:

$$\Pr(x_4, \mid \langle x_1 = \text{once},\ x_2 = \text{upon},\ x_3 = \text{a}\rangle).$$

And so on. This recursive process is called *autoregressive language generation*, and underlies how generative language models like GPT and Llama produce text. The process sketched above is called "greedy decoding", where at each step the token with the maximum likelihood is chosen. Obviously, this strategy may lead to suboptimal sequences (i.e., where the resulting sequence is low probability), thus alternative decoding strategies exist that either incorporate stochasticity, explore multiple steps ahead before "committing", or combine aspects of both.

F.1.2 Using LLMs

In general, LLMs require more computational resources than a typical statistical or machine learning model that researchers can run on a laptop. Therefore, users are likely to run LLMs using computing resources located outside of their laptop (e.g., when users run GPT models, computation is not conducted in your laptop but in OpenAI's server). There are three broad solutions for this requirement.

1. **API-based Usage**: a company hosts and manages the computing equipment and models, and charges researchers for usage (e.g. per request or per token generated). For example, GPT models can be used in this way.

2. **Cloud Computing**: Researchers secure the necessary computing resources from a cloud computing provider such as AWS, Azure, Google Cloud, and so on.

3. **HPC-based Usage**: Researchers secure access to high-performance computing resources either from their own institution or via multi-institution agreements.

The second and third options also require researchers to manage deployment and interaction with the model. This requires cloud- and HPC-specific engineering skills that go beyond the scope of this guide. This guide primarily covers API-based usage of the two families of models used in this paper: GPT and Llama-2. API-based use is currently the easiest and fastest to set up.

F.1.3 Using GPT

GPT is a closed-source proprietary model, and therefore can only be used via an API. The main provider of the API is OpenAI, and will be the option most researchers would start at the time of writing.

OpenAI provides multiple services. At the time of writing, GPT inference is provided under their Developer Platform at https://platform.openai.com. OpenAI provide a detailed Quickstart guide (at https://platform.openai.com/docs/quickstart) that provides all the instructions needed to get up and running with GPT models with OpenAI. Official libraries for API usage are provided in Python, bash (curl) and javascript (node.js).

For `Python` users, researchers can call GPT models via `Python` when they finish setting up their OpenAI account with the aforementioned quick guide.

For `R` users, there is no official R library for OpenAI API usage. While OpenAI itself does not offer a package, there are multiple `R` packages that researchers can use to call GPT models from `R`, e.g., https://irudnyts.github.io/openai/, https://github.com/joeornstein/text2data, https:

//github.com/samterfa/openai, and https://openair-lib.org/. Note that the software ecosystem evolves rapidly, especially due to changes and updates from OpenAI themselves.

F.1.4 Using Llama

There is no "official" API for the Llama-2 family of models from Meta. The model can be downloaded after receiving permission from Meta (request access at https://llama.meta.com/llama-downloads/), but as noted above the largest model (Llama-2-70B) is far too large to be run on most consumer hardware without modification to the model.

F.1.5 Third-Party APIs

Several third-party API solutions exist, including:

- HuggingFace Inference Endpoints (https://ui.endpoints.huggingface.co): provides a simple graphical user interface to set up your own API. Most transparent solution, with the full software stack fully open source and replicable (https://github.com/huggingface/text-generation-inference). Billed by time.

- Azure AI Studio (https://ai.azure.com): provides a way to create API endpoint with open source and OpenAI models. Billed by usage.

- Replicate (https://replicate.com): provides an API for many open-source models. Billed by usage.

Google Colab

Google provides a cloud computing and research toolkit called Google Colab (https://colab.research.google.com). Colab provides Jupyter notebooks that can be attached to compute runtimes with an easy-to-use graphical interface. This may be a familiar and simple solution for many researchers.

It is possible to run the 70-billion parameter Llama model on a paid Google Colab runtime instance (specifically, the A100-high memory) by quantizing the model weights to 4-bit integers. We provide an example of how to do this in our replication materials.

Self-Hosting

For this paper, we managed our own deployment of Llama-2-70B on computing clusters. Non-quantized deployment for inference required two A100-80GB Nvidia GPUs. The codebase for this deployment is provided as part of the replication code.

F.2 Prompts Used in Social Science Applications

F.2.1 Examples

In this section, to show the wide applicability of LLM annotations, we provide examples of prompts used in a wide range of social science applications. As we see below, there are huge variations in how scholars provide prompts just like there are variations in how we write codebooks and train human-coders.

Please note that, in practice, users will find that the performance of LLM annotations changes when they change prompts (even if two prompts have substantively the same meaning). This black-box sensitivity is exactly one of the key challenges of directly using LLM annotations (and ignoring prediction errors) in downstream analyses. With DSL, users can obtain statistically valid estimates regardless of the choice of LLMs and prompts because DSL can fully take into account prediction errors in LLM annotations. Indeed, as we see more thoroughly in our empirical applications (Section 5), DSL estimates are stable regardless of the choice of LLMs.

- **Sentiment Classification** (Ornstein, Blasingame, and Truscott 2022)

  ⋆ Goal: Classify the sentiment of social media posts

  ⋆ Texts to be labeled:

  ```
  Congratulations to the SCOTUS. American confidence in the Supreme Court is now
  lower than at any time in history. Well done!
  ```

  ⋆ Prompt:

  ```
  Decide whether a Tweet’s sentiment is positive, neutral, or negative.


  Tweet:  Congratulations to the SCOTUS. American confidence in the Supreme Court
  is now lower than at any time in history. Well done!
  Sentiment:
  ```

- **Ideological Scaling Task** (Ornstein, Blasingame, and Truscott 2022)

  ⋆ Goal: Classify the ideology of a political manifesto

  ⋆ Texts to be labeled:

We will implement a comprehensive strategy for ending low pay, notably by the introduction of a statutory national minimum wage.

⋆ Prompt:

```
Decide whether this sentence from a British political manifesto is Liberal, Conservative, or Neither.


Sentence: We will implement a comprehensive strategy for ending low pay, notably by the introduction of a statutory national minimum wage.
Classification:
```

- **Classification of Topics** (Rytting et al. 2023)

⋆ Goal: Classify a topic of news headlines

⋆ Texts to be labeled:

House Panel Votes Tax Cuts, But Fight Has Barely Begun

⋆ Prompt:

```
Using only the following categories
"""
Macroeconomics
Civil Rights, Minority Issues, and Civil Liberties
Health
Agriculture
Labor
Education
Environment
Energy
Immigration
Transportation
Law, Crime, and Family Issues
Social Welfare
Community Development and Housing Issues
Banking, Finance, and Domestic Commerce
Defense
```

```
Space, Science, Technology and Communications
Foreign Trade
International Affairs and Foreign Aid
Government Operations
Public Lands and Water Management
State and Local Government Administration
Weather and Natural Disasters
Fires
Arts and Entertainment
Sports and Recreation
Death Notices
Churches and Religion
Other, Miscellaneous, and Human Interest
"""
Assign the following headlines to one of the categories:
House Panel Votes Tax Cuts, But Fight Has Barely Begun ->
```

- **Hate Speech** (Ziems et al. 2024)

⋆ Goal: Classify a type of hate speech

⋆ Texts to be labeled:

```
jewish harvard professor noel ignatiev wants to abolish the white race via #wr
```

⋆ Prompt:

```
jewish harvard professor noel ignatiev wants to abolish the white race via #wr

Which of the following categories of hate speech best describes the sentence
above?
A: White Grievance (frustration over a minority group's perceived privilege
and casting majority groups as the real victims of racism)
B: Incitement to Violence (flaunting in-group unity and power or elevating
known hate groups and ideologies)
C: Inferiority Language (implies one group or individual is inferior to
another, including dehumanization and toxification)
```

```
D: Irony (the use of sarcasm, humor, and satire to attack or demean a protected class or individual)
E: Stereotypes and Misinformation (associating a protected class with negative attributes)
F: Threatening and Intimidation (conveys a speaker commitment to a target's pain, injury, damage, loss, or violation of rights)

Constraint: Answer with one or more of the options above that is most accurate and nothing else. Always choose at least one of the options.
```

- **Attitudes toward Immigrants** (Mets et al. 2023)

⋆ Goal: Classify attitudes toward immigrants

⋆ Texts to be labeled:

```
Unfortunately, by now the violence has seeped from immigrant communities to all of the society.
```

⋆ Prompt:

```
Tag the following numbered sentences as being either "supportive", "against" or "neutral" towards the topic of immigration. "Supportive" means: "supports immigration, friendly to foreigners, wants to help refugees and asylum seekers". "Against" means: "against immigration, dislikes foreigners, dislikes refugees and asylum seekers, dislikes people who help immigrants". "Neutral" means: "neutral stance, neutral facts about immigration, neutral reporting about foreigners, refugees, asylum seekers". Don't explain, output only sentence number and stance tag.

1. Unfortunately, by now the violence has seeped from immigrant communities to all of the society.
2. [truncated]
3. [truncated]
```

- **Public Opinion on Wars** (Zhu et al. 2023)

⋆ Goal: Classify whether a social media post is pro-Russia, Pro-Ukraine, or unbiguous.

⋆ Texts to be labeled:

```
International Criminal Court: Stop Putin’s War Crimes - Sign the Petition!
https://t.co/NyaFp6TTNj via @Chang
```

⋆ Prompt:

```
Give the tweet about Russo-Ukrainian Sentiment a label from Pro-Russia,
Pro-Ukraine, or Not Sure.
Tweet: "International Criminal Court: Stop Putin’s War Crimes - Sign the
Petition! https://t.co/NyaFp6TTNj via @Chang"
Label:
Explanation:
```

F.2.2 Prediction Performance in Other Applications: Figure 3

To further illustrate this wide variation in prediction performance, we also analyze a diverse set of empirical validation studies. In particular, based on a review paper by Ollion et al. (2023), we collected eight recent papers that examine the performance of LLM annotations in the social sciences and report the F-1 scores (or publicly share data so that we could compute the F-1 scores). Eight papers are as follows: Heseltine and Clemm Von Hohenberg (2023), Kuzman, Ljubešić, and Mozetič (2023), Mellon et al. (2022), Mets et al. (2023), Møller et al. (2023), Rathje et al. (2023), Yang and Menczer (2023), and Ziems et al. (2024). In each paper, they evaluate multiple different tasks, so we have 113 tasks in total. We find that F-1 scores range from as low as 20% to more than 95%, and many tasks show about $70 \sim 80\%$ (see Panel (c) in Figure 3). This huge variation in prediction accuracy is a common feature of LLM annotations in the social sciences.

We want to emphasize that this set of tasks is not representative of text annotation tasks that social scientists might perform. The results are probably the over-estimation of the current LLM performance given that these papers are trying to show the promise of LLM annotations. At the same time, the LLM performance is also likely to go up quickly as we have better and larger LLMs over time. This descriptive analysis is only meant to show the promise but also the risk of using LLMs.

G Empirical Application based on Fowler et al. (2021)

G.1 LLM Annotations

G.1.1 Specification of LLM Annotations

Models. We use three LLMs for our empirical application based on Fowler et al. (2021): GPT-3.5 (`gpt-3.5-turbo-0613`), GPT-4 (`gpt-4-preview-1106`) and Llama-2-70B-chat. This choice is informed primarily by leaderboard performance and popularity in the scientific community. In general, we recommend using a state-of-the-art LLM (currently, GPT-4) and one state-of-the-art open-source LLM (currently, Llama-2).

The GPT models are a collection of proprietary decoder-only transformer models from OpenAI. Architecturally, GPT-3.5 is known to be a 175 billion parameter model whereas the details of GPT-4 are not publicly confirmed.[6] For these models, we set the temperature to 0.0 as we are predicting short sequences and are more interested in the probabilities that GPT places over the classes, and restrict the maximum number of new tokens to 10. Texts that would cause the maximum sequence length to be exceeded were truncated, and all generated texts were classified into the three target classes by searching for the final instance of one of the three class names after lowercasing the sequence. We used the GPT models via the OpenAI Python API.

Llama-2 is a series of decoder-only transformer model trained and shared by Meta. At the time of writing, Llama-2 is available in three sizes: 7 billion, 13 billion and 70 billion parameters. We use the largest Llama-2 model because a consistent finding in the computer science literature is that larger models generally perform better. To implement Llama-2, we used the supercomputing cluster, which houses compute nodes with 4×A100 80GB NVidia GPUs. Quantizing the model to 4-bit integers (`nf4`), we were able to fit the 70-billion parameter model on a single device, enabling data and tensor parallelism to quickly generate our labels. Researchers can also use the third-party API, such as HuggingFace Inference Endpoints (https://ui.endpoints.huggingface.co) to implement Llama-2 without self-hosting. In order to perform batch processing with Llama-2, it is necessary to pad the sequences so that all observations in each batch are the same length. Thus, during pre-processing we padded or truncated all prompts to be the same length (4000 tokens, to leave 96 tokens for the generated text). However, including the pad tokens does affect the generated texts, meaning that the results would be different if we were to do the texts one-at-a-time without padding.

[6] Unofficially, it is thought to be a 8×220 billion parameter mixture-of-experts (MoE) model.

Prompt. LLMs can be used to obtain predicted labels for texts by including the text to be labeled in the condition and then autoregressively generating the predicted label. In general, users should include (a) a codebook, (b) texts to be labeld, and (c) an answer box as prompts.

In our empirical application based on Fowler et al. (2021), we follow the wording in the WMP codebook instructions used in the original paper.

```
In your judgment, is the primary purpose of the ad text to promote a specific candidate, attack a candidate, or contrast the candidates? Answer either "contrast", "promote", or "attack".

text: """
{text}
”””


Answer:
```

Here, {text} denotes a text to be labeled (i.e., a text of a political advertisement). The combined prompt-plus-text is then given as an input to a LLM, which generates text using the autoregressive process described above. Thus, given the political ad text `Trump National Coalition Chair running for Congress here in NH`, the input to the language model would be:

```
In your judgment, is the primary purpose of the ad text to promote a specific candidate, attack a candidate, or contrast the candidates? Answer either "contrast", "promote", or "attack".

text: """
Trump National Coalition Chair running for Congress here in NH
”””


Answer:
```

Few-Shot Learning. In this section we consider “few-shot learning” (also known as “in-context” learning) to improve the prediction performance of LLM annotations. This consists of incorporating task-answer pairs, called “exemplars”, in the prompt. The previously seen prompts are referred to as “zero-shot” because they contain zero exemplars. We now generate predictions for the same prompts with six exemplars (6-shot). To select our exemplars, we labeled a randomly sampled subset of the data and selected our exemplars from this subset.

When selecting exemplars, we chose a balanced number of exemplars from each class. To reflect different types of cases, we aimed to include diverse examples.

Incorporating an exemplar in a prompt typically consists of repeating the observation and answer portion of the prompt. For our 6-shot prompt, we have the following.

```
In your judgment, is the primary purpose of the ad text to promote a specific
candidate, attack a candidate, or contrast the candidates? Answer either "contrast",
"promote", or "attack".

Text:"""
On Tuesday, Nevada voters will choose between a local problem solver and a con
man who funneled money from a children's charity into a failed political campaign.
It's no wonder voters have rejected Danny Tarkanian 5 times.
"""
Answer: contrast

Text:"""
My opponent supports raising income taxes on you by over $2,800 a year! I believe
that's just wrong and I'll fight for lower taxes so that you can keep more of your
hard-earned money. By now you know kim schreyer aka dr. tax you've heard about her
57 cents a gallon gas tax. i'm dino rossi and i approve this message.
"""
Answer: attack

Text:"""
Donald Trump wants to roll back women's choice through the Supreme Court.  I'll
protect women's health care, and fully fund Planned Parenthood, in Florida.
"""
Answer: promote

Text:"""
My commitment is to always be on the side of Arizona families, not big donors. As
voting nears, the choice couldn't be more clear -- the politician that is focused
on serving donors, or the new candidate that is focused on serving you. I ask you
```

```
to stand with me.
”””
Answer: contrast


Text:”””
Nobody thought a Democrat could win in Alabama. But last year, Doug Jones shocked
the country and flipped a historically deep-red seat blue. Now, it is our turn.
”””
Answer: promote


Text:”””
Lt Governor Dan Patrick cut public education funding by over five billion dollars,
cut more than ten thousand teaching positions, and cut support for pre-K. With
fewer teachers and larger class sizes, Dan Patrick won’t let teachers teach and
students learn. We need new leadership in Texas - vote Mike Collier for Lt Governor.
”””
Answer: attack


Text:”””
{text}
”””
Answer:
```

Other Recommendations. When formulating their prompt, researchers may also want to consider model-specific factors. For instance, HuggingFace suggests that the chat variants of Llama-2 be prompted using the following template, which is based on the model’s training procedure.

```
<s>[INST] <<SYS>>
{{ system_prompt }}
<</SYS>>


{{ user_message }} [/INST]
```

However, these guidelines are better thought of reasonable hypotheses extrapolated from aspects of the model training process rather than surefire ways to generate great predictions.

G.1.2 Estimates in Section 5.1

		Outcome = Contrast			Outcome = Promote		
Automated Annotation Methods		Estimates	S.E.	Coverage	Estimates	S.E.	Coverage
Benchmark		-0.13	0.02	⋆	0.17	0.02	⋆
Classical	Dropout Logit	-0.09	0.01	0.04	0.14	0.01	0.27
Supervised ML	Lasso	-0.04	0.00	0.00	0.11	0.01	0.02
	Ridge	-0.06	0.00	0.00	0.10	0.00	0.00
	Random Forest	-0.10	0.01	0.07	0.15	0.01	0.22
	XGBoost	-0.11	0.01	0.40	0.18	0.01	0.55
LLM-only	GPT-4: Zero-Shot	-0.01	0.01	0.00	0.08	0.03	0.13
estimation	GPT-4: Few-Shot	-0.02	0.02	0.00	0.16	0.03	0.96
	GPT-3.5: Zero-Shot	-0.04	0.01	0.00	0.16	0.03	0.99
	GPT-3.5: Few-Shot	0.00	0.01	0.00	0.18	0.03	0.96
	Llama-2: Zero-Shot	0.00	0.04	0.02	0.01	0.03	0.00
	Llama-2: Few-Shot	-0.10	0.02	0.82	0.07	0.03	0.10
DSL	GPT-4 : Zero-Shot	-0.14	0.04	0.96	0.18	0.05	0.96
	GPT-4: Few-Shot	-0.14	0.04	0.96	0.18	0.04	0.96
	GPT-3.5: Zero-Shot	-0.14	0.04	0.95	0.18	0.05	0.97
	GPT-3.5: Few-Shot	-0.14	0.04	0.95	0.18	0.05	0.97
	Llama-2: Zero-Shot	-0.14	0.05	0.95	0.18	0.05	0.95
	Llama-2: Few-Shot	-0.14	0.04	0.95	0.18	0.05	0.96

Note: There is no result on coverage for the "Benchmark" estimates because coverage rates are defined as the probability of each method's confidence interval covering the "Benchmark" estimate.

G.1.3 Additional LLM Results

In Section 2.1.2, we reported the F1 scores for the overall performance of LLM annotations. Here, we provide additional details.

Figure OA-3 shows the overall and disaggregated performance. Several points are worth noting. First, the performance can vary across models and prompts. Second, the prediction accuracy is not uniform across three categories. For all models, it is easiest to predict "Promote" and hardest to predict "Contrast". This directly means that prediction errors are affected by the outcome of interest (i.e., the tone of ads) itself, which implies that researchers cannot ignore prediction errors (see Section 3.3).

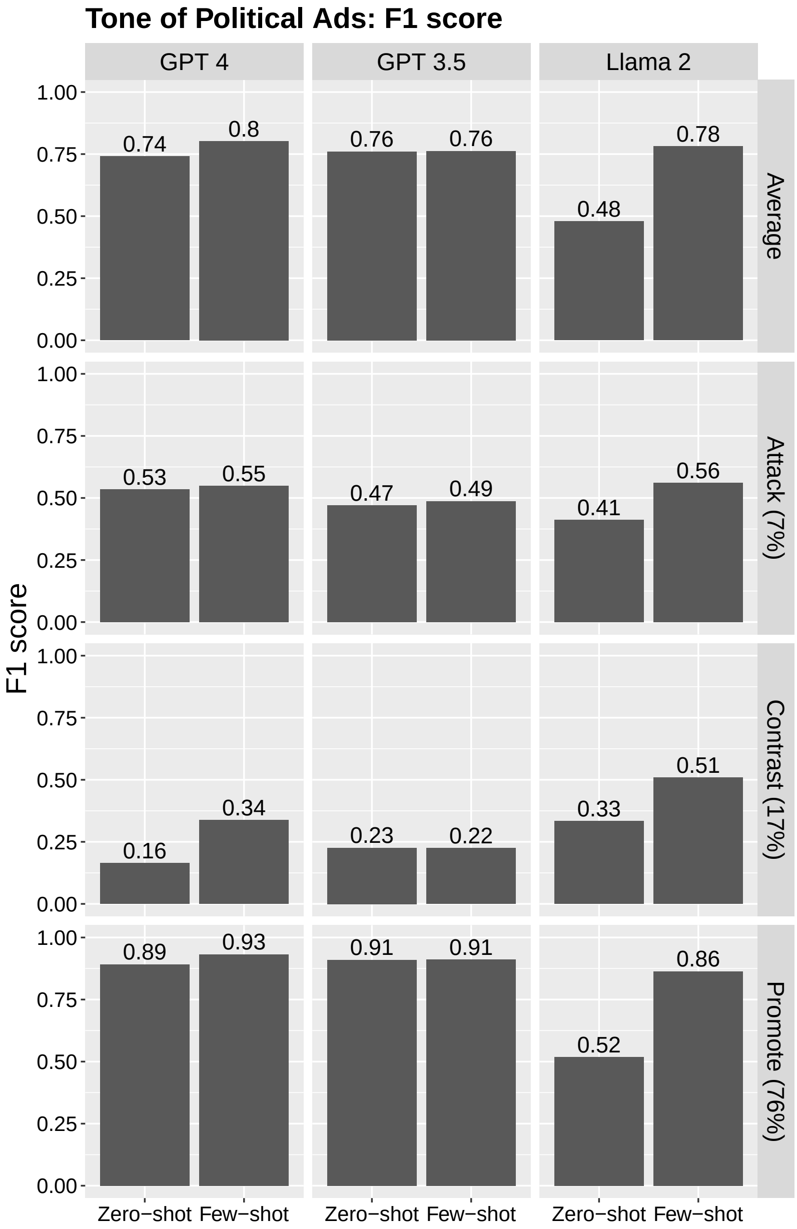


Figure OA-3: **F1 scores of LLM annotations: Fowler et al. (2021).**
Note: "Average" shows the overall performance which is the weighted average of F1 scores for three categories: "Attack", "Contrast", and "Promote" constitute 7%, 17%, and 76% of political ads.

In the main paper, we reported F1 score, which is the most common measure of prediction performance in the computer science and machine learning literature, instead of the classification accuracy. It is important to remember that the classification accuracy can be high just because the category is rare. For example, when the category takes 1 only with 5%, by predicting 0 for every observation, we can get 95% accuracy automatically. This is the main reason why F1 scores are generally recommended as a measure of prediction performance when classes are imbalanced.

Given this caveat, for some researchers who might be more familiar with accuracy, we also report the classification accuracy for LLM predictions in Figure OA-4.

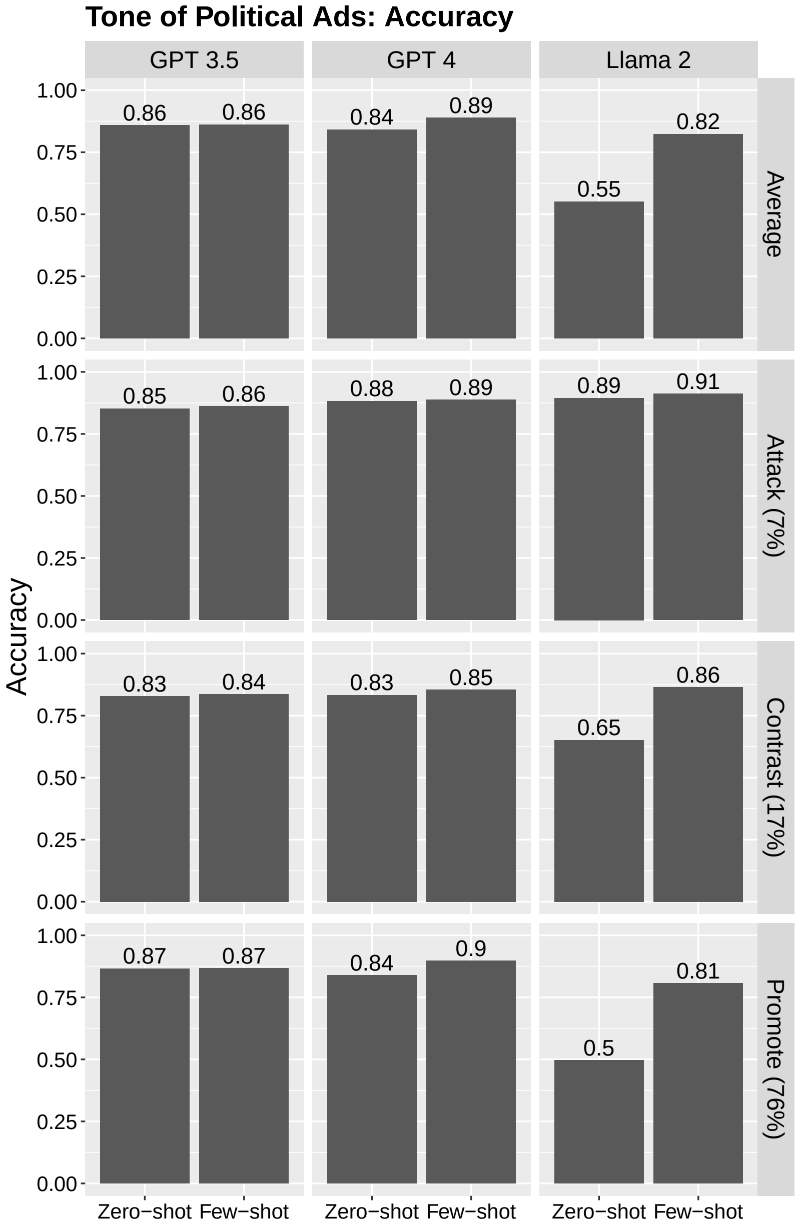


Figure OA-4: **Accuracy of LLM annotations: Fowler et al. (2021).**
Note: "Average" shows the overall performance which computes the weighted average of the classification accuracies for three categories: "Attack", "Contrast", and "Promote" constitute 7%, 17%, and 76% of political ads.

G.2 Additional DSL Results for Fowler et al. (2021)

In the main paper, we reported the results for two outcomes, “Contrast” and “Promote”. In this appendix, we also report the results for the third outcome “Attack.” The main findings are similar. First, estimates from the LLM-only estimation are biased, and substantive and statistical conclusions can flip depending on which LLMs users choose. Confidence intervals are also, in general, invalid. For this outcome, the LLM-only estimation with Llama-2 has reasonable coverages, but this is a statistical coincidence without any theoretical guarantee. Indeed, if we look back at two other outcomes, the same estimator has poor coverage. Second, estimates from the classical supervised learning method are similarly biased because they ignore prediction errors, and they have invalid confidence intervals. In contrast to these existing approaches, DSL has unbiased estimates and valid confidence intervals.

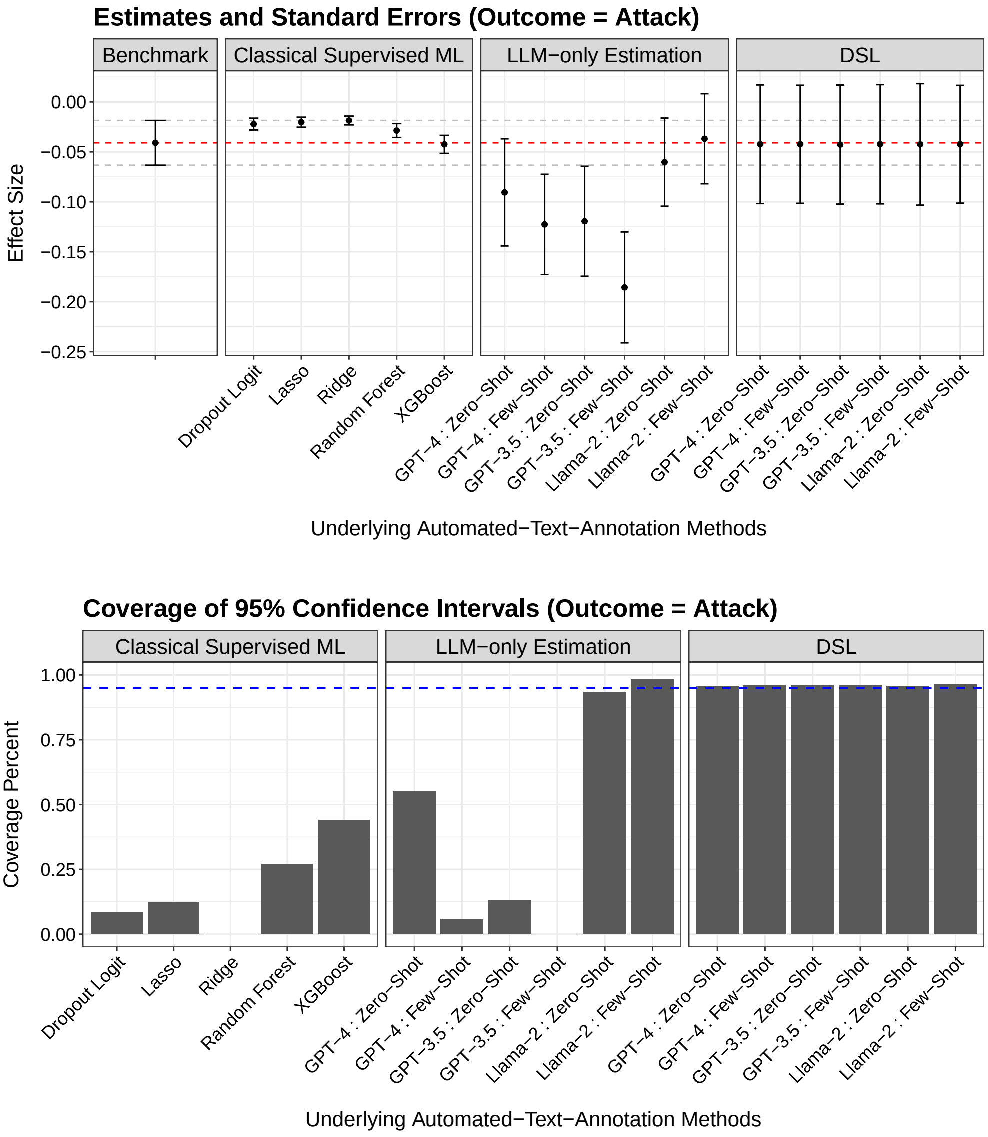


Figure OA-5: **Results for Outcome = "Attack" in Fowler et al. (2021).** *Note*: In the top panel, red dotted lines represent point estimates of the "Benchmark" estimates and gray dotted lines represent their 95% confidence intervals. To show the average performance across random sampling of expert-coding, we report the average point estimates and standard errors across 500 repeated sampling. In the bottom panel, blue dotted lines represent 95%.

H Empirical Application based on Pan and Chen (2018)

H.1 LLM Annotations for Pan and Chen (2018)

H.1.1 Specification of LLM Annotations

Models. As we did for the first application, we use three LLMs for our empirical application based on Pan and Chen (2018): GPT-3.5, GPT-4 and Llama-2-70B-chat.

Prompt. In this empirical application, we annotate two different variables: *Prefecture Wrongdoing* (whether each citizen complaint accuses of prefecture-level wrongdoing) and *County Wrongdoing* (whether each citizen complaint accuses of county-level wrongdoing). For *Prefecture Wrongdoing*, we used the following prompt for GPTs. For Llama-2, we found that the performance based on Chinese prompts is so poor that we decided to use the English version of the same prompt.

> 下面的中文文本是发生在中国江西省九江市的一起投诉，判断帖子是否指控了九江市地级层面的政府官员或政府机构有不当行为，包括指控腐败和暴力，以及违反法律和法规。请注意，评估仅考虑针对九江市地级的指控，不包括针对下辖地区及其子地区的任何指控（这些下辖地区包括’濂溪区’、’浔阳区’、’柴桑区’、’瑞昌市’、’共青城市’、’庐山市’、’武宁县’、’修水县’、’永修县’、’德安县’、’都昌县’、’湖口县’、’彭泽县’）。
>
> 如果帖子包含对九江市地级官员或机构的上述性质的指控，请返回’1’。否则，请返回’0’。
>
> 文本：”””
> {text}
> ”””
> 回答：

where {text} denotes that a text to be labeled (i.e., each online complaint). The combined prompt-plus-text is then given as an input to a LLM.

For *County Wrongdoing*, we used the following prompt for GPTs. Again, for Llama-2, we found that the performance based on Chinese prompts is so poor that we decided to use the English version of the same prompt.

> 下面的文本是发生在中国江西省九江市的一起投诉。判断帖子是否指控了九江市下辖地区及其子地区的政府官员或政府机构有不当行为，这些不当行为包括指控腐败和暴力，以及违反

法律和法规。请注意，评估仅考虑针对九江市地级下辖地区或其子地区的指控（下辖地区包括濂溪区、浔阳区、柴桑区、瑞昌市、共青城市、庐山市、武宁县、修水县、永修县、德安县、都昌县、湖口县、彭泽县），不包括任何针对九江市地级的指控。

如果帖子包含针对九江市地级市下辖地区或其子地区官员或机构的上述性质的指控，请返回’1’。否则，请返回’0’。

文本：”””
{text}
”””
回答:

Few-Shot Learning. As in the first example, we also consider few-shot learning by adding diverse examples. For *Prefecture Wrongdoing*, we used the following prompt.

下面的中文文本是发生在中国江西省九江市的一起投诉，判断帖子是否指控了九江市地级层面的政府官员或政府机构有不当行为，包括指控腐败和暴力，以及违反法律和法规。请注意，评估仅考虑针对九江市地级的指控，不包括针对下辖地区及其子地区的任何指控（这些下辖地区包括’濂溪区’、’浔阳区’、’柴桑区’、’瑞昌市’、’共青城市’、’庐山市’、’武宁县’、’修水县’、’永修县’、’德安县’、’都昌县’、’湖口县’、’彭泽县’）。

如果帖子包含对九江市地级官员或机构的上述性质的指控，请返回’1’。否则，请返回’0’。

文本: ”””
媒体报道：各地突击提拔干部透视：一把手权力过于集中所致。文中提及，今年6月,江西省委常委、秘书长赵智勇涉嫌违纪被免职后,就被媒体曝出,他2006年离开九江前1个月,曾突击提拔了一批女干部,有的学校老师直接被提拔为区团委副书记,不少属于破格提拔,但后来接任的领导在接到群众反映后,又把提拔的一部分女干部打回原单位。（人民网等36家媒体）”””
回答: 1

文本: ”””
网友举报九江市工商局局长原局长孔祥华贪污腐化，包养情妇等问题。（天涯社区）”””
回答: 1

文本: ”””

网曝九江市公路管理局九江分局局长李广金长期公车私用，出入酒店。（九江论坛）”””
回答: 1

文本: ”””
网友质疑永修县医保局敛财，2012年初办理医保时，永修县医保局强迫必须多缴两年无编制期间的医保统筹金，共计2700多元，否则不予办理医保。统筹金不打入个人医保帐户，单位也没有任何补助，卡里没有一分钱，等于是白交，还不开任何发票凭证。（问政江西）”””
回答: 0

文本: ”””
网友举报武宁县船滩镇党委委员黄少华违规违纪，以搭干股、圈矿山用地、贩卖土方石料的方式聚敛钱财，经常出入于大小宾馆酒店聚众赌博，拉帮结派。（大江论坛）”””
回答: 0

文本：”””
{text}
”””
回答:

For *County Wrongdoing*, we used the following prompt.

下面的文本是发生在中国江西省九江市的一起投诉。判断帖子是否指控了九江市下辖地区及其子地区的政府官员或政府机构有不当行为，这些不当行为包括指控腐败和暴力，以及违反法律和法规。请注意，评估仅考虑针对九江市地级下辖地区或其子地区的指控（下辖地区包括濂溪区、浔阳区、柴桑区、瑞昌市、共青城市、庐山市、武宁县、修水县、永修县、德安县、都昌县、湖口县、彭泽县），不包括任何针对九江市地级的指控。

如果帖子包含针对九江市地级市下辖地区或其子地区官员或机构的上述性质的指控，请返回’1’。否则，请返回’0’。

文本: ”””
网友质疑永修县医保局敛财，2012年初办理医保时，永修县医保局强迫必须多缴两年无编制期间的医保统筹金，共计2700多元，否则不予办理医保。统筹金不打入个人医保帐户，单位也没有任何补助，卡里没有一分钱，等于是白交，还不开任何发票凭证。（问政江西）

”””

回答: 1

文本: ”””
网友举报武宁县船滩镇党委委员黄少华违规违纪，以搭干股、圈矿山用地、贩卖土方石料的方式聚敛钱财，经常出入于大小宾馆酒店聚众赌博，拉帮结派。（大江论坛）
”””

回答: 1

文本: ”””
都昌县汪墩乡茅垅村质疑信访事项答复意见书，称镇领导在征收农田过程中大发横财。（天涯社区）
”””

回答: 1

文本: ”””
网曝九江市公路管理局九江分局局长李广金长期公车私用，出入酒店。（九江论坛）
”””

回答: 0

文本: ”””
网友举报九江市工商局局长原局长孔祥华贪污腐化，包养情妇等问题。（天涯社区）
”””

回答: 0

文本：”””
{text}
”””

回答:

H.1.2 Estimates in Section 5.2

		Prefecture Wrongdoing			County Wrongdoing		
Automated Annotation Methods		Estimates	S.E.	Coverage	Estimates	S.E.	Coverage
Benchmark		-0.99	0.30	⋆	0.27	0.20	⋆
Classical	Lasso	-0.13	1.27	0.79	0.01	0.45	0.89
Supervised ML	Ridge	-0.02	0.63	0.58	0.01	0.35	0.89
	Random Forest	0.08	0.70	0.63	-0.02	0.40	0.88
	XGBoost	-0.00	0.34	0.17	-0.01	0.23	0.74
LLM-only	GPT-4: Zero-Shot	-1.02	0.41	0.97	-0.71	0.20	0.00
estimation	GPT-4: Few-Shot	-2.09	0.34	0.04	-1.03	0.26	0.00
	GPT-3.5: Zero-Shot	-1.31	0.27	0.81	-0.26	0.25	0.43
	GPT-3.5: Few-Shot	-1.36	0.32	0.83	-0.72	0.22	0.00
	Llama-2: Zero-Shot	-0.92	0.26	0.91	0.21	0.23	0.92
	Llama-2: Few-Shot	-0.57	0.28	0.64	0.06	0.24	0.87
DSL	GPT-4: Zero-Shot	-1.08	0.52	0.96	0.26	0.32	0.96
	GPT-4: Few-Shot	-1.08	0.51	0.97	0.26	0.33	0.94
	GPT-3.5: Zero-Shot	-1.06	0.56	0.97	0.29	0.31	0.97
	GPT-3.5: Few-Shot	-1.06	0.55	0.96	0.27	0.30	0.95
	Llama-2: Zero-Shot	-1.09	0.53	0.97	0.25	0.31	0.95
	Llama-2: Few-Shot	-1.09	0.55	0.97	0.28	0.30	0.95

Note: There is no result on coverage for the "Benchmark" estimates because coverage rates are defined as the probability of each method's confidence interval covering the "Benchmark" estimate.

H.1.3 Additional LLM Results

In Section 2.1.2, we reported the F1 scores for the overall performance of LLM annotations. Here, we provide additional details.

Figure OA-3 shows F1 scores and the classification accuracy for both "Prefecture Wrongdoing" and "County Wrongdoing". Several points are worth noting. First, the performance varies across two tasks. It is easier to predict "Prefecture Wrongdoing" than to predict "County Wrongdoing" across models and prompts. Second, in this application, the prediction performance is relatively stable across models and prompts. However, as we see in Section 5, even though the average prediction performance is relatively similar, because prediction errors are non-random, LLM-only estimation using different LLM annotations produces very different results. This highlights the methodological problem of ignoring prediction errors.

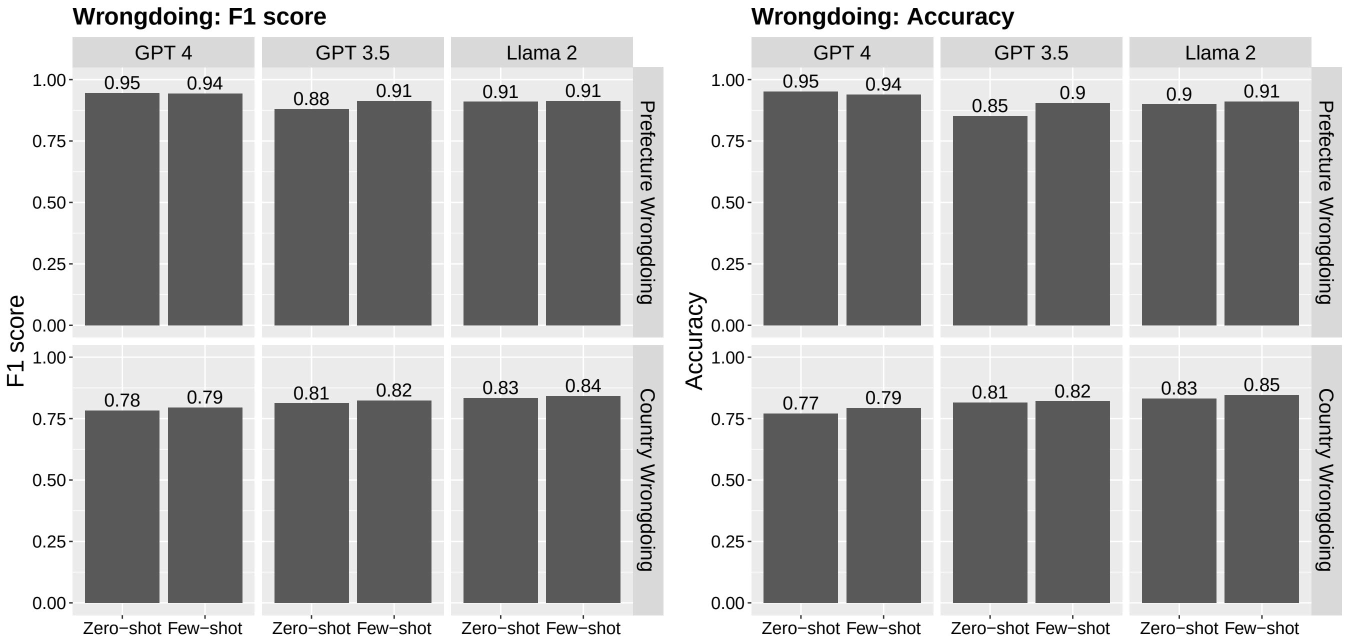


Figure OA-6: **Prediction Performance of LLM annotations: Pan and Chen (2018).**
Note: The left panel shows F1 scores and the right panel shows the classification accuracy.

H.2 Additional DSL Results for Pan and Chen (2018)

Setup

DSL requires simple four steps. First, we generate LLM annotations for the entire population of documents. As we discussed in Section 2, we here consider six versions: GPT 4, GPT 3.5, and Llama 2 with zero-shot and few-shot learning. In the second step, we randomly sample 500 documents for expert-coding.[7] In the third step, using the expert-coded data, we further improve LLM predictions by cross-fitting the generalized random forest (Athey, Tibshirani, and Wager 2019) to predict the expert-coded labels with LLM annotations produced in the first step. Finally, we combine expert-coded labels and predicted labels in the DSL logistic regression with exactly the same specification in the original paper. In particular, we regress the upward reporting (i.e., whether a given complaint is reported upward to provincial-level officials) on the aforementioned two independent variables (*Prefecture Wrongdoing* and *County Wrongdoing*) and other control variables with interactions.

$$\begin{aligned}\text{Upward Reporting} \;\sim\; & \text{Prefecture Wrongdoing} + \text{County Wrongdoing} + \text{Connections} + \\ & \text{County Wrongdoing} \times \text{Connections} + \text{Other Controls}\end{aligned}$$

where "Other Controls" include prevalence, group issue, sentiment, personal experience, collective action, petitions, and provincial jurisdiction. See Column (3) in Table 3 of the original paper. Importantly, "Prefecture Wrongdoing" and "County Wrongdoing" are text-based.

We compare DSL against the classical supervised learning approach and the LLM-only estimation. For the classical supervised learning approach, we examine five widely used supervised ML methods: lasso, ridge, random forest, and XGBoost. We use a term-document matrix used in the original paper, which has more than 5000 variables, as predictors. For the LLM-only estimation, we consider the same six versions of LLM annotations. We expect that these existing approaches can provide unbiased estimates with valid confidence intervals, only when prediction errors are completely random, while DSL provides valid statistical guarantees even with arbitrary prediction errors.

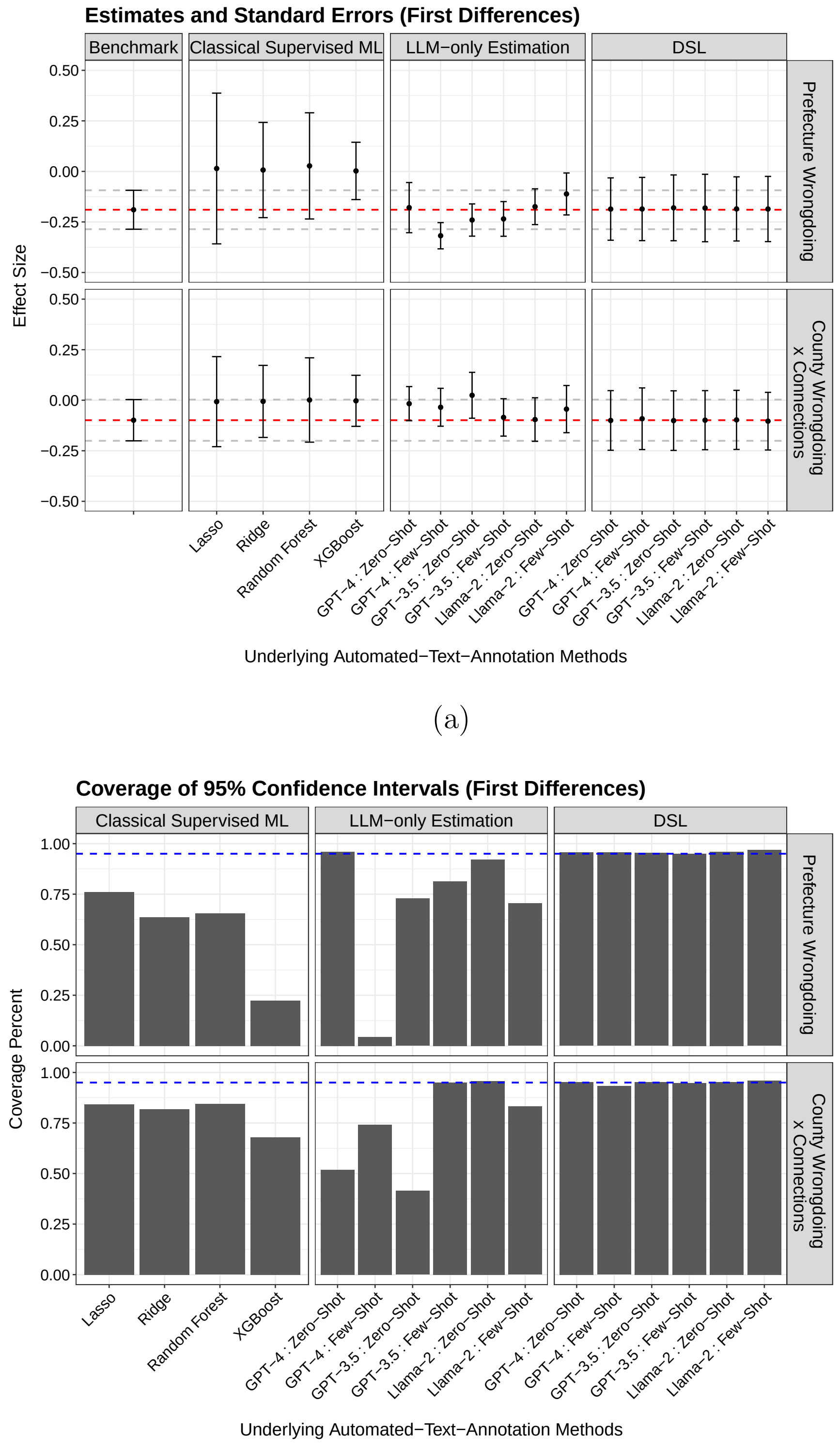
[7] In this empirical validation, we rely on expert-coding from the original authors, so we simply reveal expert-coding for sampled documents.

Results

In the main paper, we reported the results for two main coefficients, “Prefecture Wrongdoing” and “County Wrongdoing”. In this appendix, we also report the results based on the first differences because coefficients of logistic regression tend to be difficult to interpret and it is often recommended to report results based on the differences in predicted probabilities. As we emphasized in the paper, researchers can apply DSL and then use estimated coefficients to compute the first differences or any other function of estimated coefficients.

Here we specifically focus on the two effects that the original authors focused on most: the effect of “Prefecture Wrongdoing” and the effect of “Connection” for posts accusing of “Country Wrongdoing”. We report the results in Figure OA-7.

The main findings are similar. First, estimates from the LLM-only estimation are biased, and substantive and statistical conclusions can flip depending on which LLMs users choose. Confidence intervals are also, in general, invalid. Some LLM-only estimators have reasonable coverages for at least one outcome, but no LLM-only estimator has 95% coverages for both effects because they do not have any theoretical guarantees. Second, estimates from the classical supervised learning method are similarly biased because they ignore prediction errors, and they have invalid confidence intervals. In contrast to these existing approaches, DSL has unbiased estimates and valid confidence intervals, regardless of the underlying automated text annotation method.



(a)

(b)

Figure OA-7: **Comparisons of DSL and Existing Approaches in terms of First Differences using Pan and Chen (2018).** *Note*: In Panel (a), red dotted lines represent point estimates of the "Benchmark" estimates and gray dotted lines represent their 95% confidence intervals. To show the average performance across random sampling of expert-coding, we report the average point estimates and standard errors across 500 repeated sampling. In Panel (b), blue dotted lines represent 95%.

I Literature Review

To evaluate the current practice of text annotations in text-as-data applications in political science, we conducted a review of academic articles published in the top 10 political science journals: American Political Science Review (APSR), American Journal of Political Science (AJPS), Journal of Politics (JOP), Political Behavior (PB), Quarterly Journal of Political Science (QJPS), British Journal of Political Science (BJPS), Comparative Political Studies (CPS), World Politics (WP), International Organization (IO), and Journal of Experimental Political Science (JEPS). These journals represent a group of highly cited and influential journals in political science. For example, these 10 journals together have total citations of over 7,800 on average as compared to the 1,315 average total citation counts across all academic journals in the field of political science. Furthermore, the 5-year journal impact factor among these 10 journals is 5.8 on average, more than twice as large as the average score across all political science journals.[8]

We first searched for all articles published in the years 2015 through 2022 (inclusive) using a keyword "text as data" and "text analysis" in Web of Science. We also included the articles published in the above-mentioned top journals that cite Grimmer and Stewart (2013). In total, we reviewed 88 papers. We note that this number is the lower bound of the actual number of papers using text-as-data methods because some papers do not explicitly use terminologies, such as "text as data" and "text analysis".

We then manually coded the following information for each paper. (1) Whether a paper uses some forms of text annotations (if so, we continue to code for the remaining items): (2) A type of downstream analyses that use text-based variables: (3) Whether text-based variables are used as the outcome and/or independent variables in the downstream analyses: (4) Whether a paper explicitly acknowledges the potential biases due to prediction errors: (5) Whether a paper statistically addresses the potential biases due to prediction errors: (6) If a paper uses the classical supervised learning approach, what is the F-1 score and the classification accuracy of the text classification step?

References

Angelopoulos, Anastasios N., Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. 2023. "Prediction-powered inference." *Science* 382 (6671): 669–674. https://doi.org/10.1126/science.adi6000. https://www.science.org/doi/abs/10.1126/science.adi6000.

[8] These values are based on a total of 307 political science journals recorded in the Journal Citation Reports provided by Web of Science.

Argyle, Lisa P, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. 2023. "Out of One, Many: Using Language Models to Simulate Human Samples." *Political Analysis* 31 (3): 337–351.

Athey, Susan, Julie Tibshirani, and Stefan Wager. 2019. "Generalized Random Forests." *The Annals of Statistics* 47 (2): 1148–1178. https://doi.org/10.1214/18-AOS1709. https://doi.org/10.1214/18-AOS1709.

Benoit, Kenneth, Michael Laver, and Slava Mikhaylov. 2009. "Treating Words as Data with Rrror: Uncertainty in Text Statements of Policy Positions." *American Journal of Political Science* 53 (2): 495–513.

Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. "On the Opportunities and Risks of Foundation Models." *arXiv preprint arXiv:2108.07258.*

Bosley, Mitchell, Saki Kuzushima, Ted Enamorado, and Yuki Shiraito. 2022. "Improving Probabilistic Models in Text Classification via Active Learning." *arXiv preprint arXiv:2202.02629.*

Card, Dallas, and Noah A Smith. 2018. "The importance of calibration for estimating proportions from annotations." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers),* 1636–1646.

Chakrabortty, Abhishek, and Tianxi Cai. 2018. "Efficient and adaptive linear regression in semi-supervised settings." *Annals of Statistics.*

Chakrabortty, Abhishek, Guorong Dai, and Eric Tchetgen Tchetgen. 2022. "A General Framework for Treatment Effect Estimation in Semi-Supervised and High Dimensional Settings." *arXiv preprint arXiv:2201.00468.*

Chen, Yi-Hau, and Hung Chen. 2000. "A Unified Approach to Regression Analysis under Double-Sampling Designs." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 62 (3): 449–460.

Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. 2018. "Double/Debiased Machine Learning for Treatment and Structural Parameters." *Econometrics Journal* 21:C1–C68.

Chernozhukov, Victor, Juan Carlos Escanciano, Hidehiko Ichimura, Whitney K Newey, and James M Robins. 2022. "Locally robust semiparametric estimation." *Econometrica* 90 (4): 1501–1535.

Clayton, Katherine, Yusaku Horiuchi, Aaron R Kaufman, Gary King, and Mayya Komisarchik. 2023. *Correcting Measurement Error Bias in Conjoint Survey Experiments.* Technical report. Working Paper.

Davidian, Marie. 2022. "Methods based on semiparametric theory for analysis in the presence of missing data." *Annual Review of Statistics and Its Application* 9:167–196.

Egami, Naoki, Christian J. Fong, Justin Grimmer, Margaret E. Roberts, and Brandon M. Stewart. 2022. "How to make causal inferences using texts." *Science Advances* 8 (42): eabg2652. https://doi.org/10.1126/sciadv.abg2652. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.abg2652. https://www.science.org/doi/abs/10.1126/sciadv.abg2652.

Egami, Naoki, Musashi Hinck, Brandon Stewart, and Hanying Wei. 2023. “Using Imperfect Surrogates for Downstream Inference: Design-based Supervised Learning for Social Science Applications of Large Language Models.” *Advances in Neural Information Processing Systems* 36.

Feder, Amir, Katherine A Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E Roberts, et al. 2022. “Causal inference in natural language processing: Estimation, prediction, interpretation and beyond.” *Transactions of the Association for Computational Linguistics* 10:1138–1158.

Fong, Christian, and Justin Grimmer. 2021. “Causal Inference with Latent Treatments.” *American Journal of Political Science.*

Fong, Christian, and Matthew Tyler. 2021. “Machine learning predictions as regression covariates.” *Political Analysis* 29 (4): 467–484.

Fowler, Erika Franklin, Michael M Franz, Gregory J Martin, Zachary Peskowitz, and Travis N Ridout. 2021. “Political Advertising Online and Offline.” *American Political Science Review* 115 (1): 130–149.

Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli. 2023. “ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks.” *arXiv preprint arXiv:2303.15056.*

Grimmer, Justin, and Brandon M Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political analysis* 21 (3): 267–297.

Hager, Anselm, and Hanno Hilbig. 2020. “Does Public Opinion Affect Political Speech?” *American Journal of Political Science* 64 (4): 921–937.

Heseltine, Michael, and B Clemm Von Hohenberg. 2023. “Large Language Models as A Substitute for Human Experts in Annotating Political Text.” *preprint SocArxiv: cx752.*

Hopkins, Daniel J, and Gary King. 2010. “A Method of Automated Nonparametric Content Analysis for Social Science.” *American Journal of Political Science* 54 (1): 229–247.

Jerzak, Connor T, Gary King, and Anton Strezhnev. 2023. “An improved method of automated nonparametric content analysis for social science.” *Political Analysis* 31 (1): 42–58.

Kallus, Nathan, and Xiaojie Mao. 2020. “On the role of surrogates in the efficient estimation of treatment effects with limited outcome data.” *arXiv preprint arXiv:2003.12408.*

Katsumata, Hiroto, and Soichiro Yamauchi. 2023. *Statistical Analysis with Machine Learning Predicted Variables.* Technical report. Working Paper.

Keith, Katherine, and Brendan O’Connor. 2018. “Uncertainty-aware generative models for inferring document class prevalence.” In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* 4575–4585. Brussels, Belgium: Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1487. https://aclanthology.org/D18-1487.

Kennedy, Edward H. 2022. “Semiparametric doubly robust targeted double machine learning: a review.” *arXiv preprint arXiv:2203.06469.*

Kennedy, Edward H, Sivaraman Balakrishnan, and Max G’Sell. 2020. “Sharp instruments for classifying compliers and generalizing causal effects.” *Annals of Statistics.*

Knox, Dean, and Christopher Lucas. 2021. “A Dynamic Model of Speech for the Social Sciences.” *American Political Science Review* 115 (2): 649–666.

Knox, Dean, Christopher Lucas, and Wendy K Tam Cho. 2022. “Testing Causal Theories with Learned Proxies.” *Annual Review of Political Science* 25:419–441.

Küchenhoff, Helmut, Samuel M Mwalili, and Emmanuel Lesaffre. 2006. “A General Method for Dealing with Misclassification in Regression: the Misclassification SIMEX.” *Biometrics* 62 (1): 85–96.

Kuzman, Taja, Nikola Ljubešić, and Igor Mozetič. 2023. “ChatGpt: Beginning of An End of Manual Annotation? Use Case of Automatic Genre Identification.” *arXiv preprint arXiv:2303.03953.*

Li, Minzhi, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy F Chen, Zhengyuan Liu, and Diyi Yang. 2023. “CoAnnotating: Uncertainty-guided Work Allocation between Human and Large Language Models for Data Annotation.” *arXiv preprint arXiv:2310.15638.*

Linegar, Mitchell, Rafal Kocielnik, and R Michael Alvarez. 2023. “Large Language Models and Political Science.” *Frontiers in Political Science* 5:1257092.

Mellon, Jonathan, Jack Bailey, Ralph Scott, James Breckwoldt, and Marta Miori. 2022. “Does GPT-3 Know What the Most Important Issue Is? Using Large Language Models to Code Open-Text Social Survey Responses At Scale.” *SSRN preprint: 4310154.*

Mets, Mark, Andres Karjus, Indrek Ibrus, and Maximilian Schich. 2023. “Automated Stance Detection in Complex Topics and Small Languages: The Challenging Case of Immigration in Polarizing News Media.” *arXiv preprint arXiv:2305.13047.*

Mikhaylov, Slava, Michael Laver, and Kenneth R Benoit. 2012. “Coder Reliability and Misclassification in the Human Coding of Party Manifestos.” *Political Analysis* 20 (1): 78–91.

Møller, Anders Giovanni, Jacob Aarup Dalsgaard, Arianna Pera, and Luca Maria Aiello. 2023. “Is A Prompt and A Few Samples All You Need? Using GPT-4 for Data Augmentation in Low-Resource Classification Tasks.” *arXiv preprint arXiv:2304.13861.*

Mozer, Reagan, and Luke Miratrix. 2023. “Decreasing the Human Coding Burden in Randomized Trials with Text-based Outcomes via Model-Assisted Impact Analysis.” *arXiv preprint arXiv:2309.13666.*

Newey, Whitney K, and Daniel McFadden. 1994. “Large Sample Estimation and Hypothesis Testing.” *Handbook of econometrics* 4:2111–2245.

Ollion, Etienne, Rubing Shen, Ana Macanovic, and Arnault Chatelain. 2023. “Chatgpt for Text Annotation? Mind the Hype!” *SocArXiv. October* 4.

Ornstein, Joseph T, Elise N Blasingame, and Jake S Truscott. 2022. *How to Train Your Stochastic Parrot: Large Language Models for Political Texts.* Technical report. Working Paper.

Palmer, Alexis, and Arthur Spirling. 2023. *Large Language Models Can Argue in Convincing and Novel Ways About Politics: Evidence from Experiments and Human Judgement.* Technical report. Working paper.

Pan, Jennifer, and Kaiping Chen. 2018. "Concealing Corruption: How Chinese Officials Distort Upward Reporting of Online Grievances." *American Political Science Review* 112 (3): 602–620.

Pangakis, Nicholas, Samuel Wolken, and Neil Fasching. 2023. "Automated Annotation with Generative AI Requires Validation." *arXiv preprint arXiv:2306.00176.*

Rathje, Steve, Dan-Mircea Mirea, Ilia Sucholutsky, Raja Marjieh, Claire Robertson, and Jay J Van Bavel. 2023. "GPT is An Effective Tool for Multilingual Psychological Text Analysis."

Robins, James M, and Andrea Rotnitzky. 1995. "Semiparametric efficiency in multivariate regression models with missing data." *Journal of the American Statistical Association* 90 (429): 122–129.

Robins, James M, Andrea Rotnitzky, and Lue Ping Zhao. 1994. "Estimation of Regression Coefficients When Some Regressors Are Not Always Observed." *Journal of the American Statistical Association* 89 (427): 846–866.

Rotnitzky, Andrea, and Stijn Vansteelandt. 2014. "Double-robust methods." In *Handbook of missing data methodology,* 185–212. CRC Press.

Rytting, Christopher Michael, Taylor Sorensen, Lisa Argyle, Ethan Busby, Nancy Fulda, Joshua Gubler, and David Wingate. 2023. "Towards Coding Social Science Datasets with Language Models." *arXiv preprint arXiv:2306.02177.*

Tarr, Alexander, June Hwang, and Kosuke Imai. 2023. "Automated Coding of Political Campaign Advertisement Videos: An Empirical Validation Study." *Political Analysis* 31 (4): 554–574.

Torres, Michelle, and Francisco Cantú. 2022. "Learning to See: Convolutional Neural Networks for the Analysis of Social Science Data." *Political Analysis* 30 (1): 113–131.

Tsiatis, Anastasios A. 2006. *Semiparametric theory and missing data.* Springer.

Vansteelandt, Stijn, and Oliver Dukes. 2022. "Assumption-lean Inference for Generalised Linear Model Parameters." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 84, no. 3 (July): 657–685. ISSN: 1369-7412. https://doi.org/10.1111/rssb.12504. eprint: https://academic.oup.com/jrsssb/article-pdf/84/3/657/49322532/rssb12504-sup-0001-supinfo.pdf. https://doi.org/10.1111/rssb.12504.

Wang, Siruo, Tyler H McCormick, and Jeffrey T Leek. 2020. "Methods for Correcting Inference based on Outcomes Predicted by Machine Learning." *Proceedings of the National Academy of Sciences* 117 (48): 30266–30275.

Wu, Patrick Y, Joshua A Tucker, Jonathan Nagler, and Solomon Messing. 2023. "Large Language Models Can be Used to Estimate the Ideologies of Politicians in A Zero-Shot Learning Setting." *arXiv preprint arXiv:2303.12057.*

Yang, Kai-Cheng, and Filippo Menczer. 2023. "Large Language Models Can Rate News Outlet Credibility." *arXiv preprint arXiv:2304.00228.*

Zhang, Han. 2021. "How Using Machine Learning Classification as a Variable in Regression Leads to Attenuation Bias and What to Do About It." SocArXiv.

Zhu, Yiming, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. "Can ChatGpt Reproduce Human-Generated Labels? A Study of Social Computing Tasks." *arXiv preprint arXiv:2304.10145.*

Ziems, Caleb, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. "Can large language models transform computational social science?" *Computational Linguistics* 50 (1): 237–291.