Bayesian LLM Finetuning for Statistical Inference

1 Introduction

Large language models (LLMs) have increasingly been employed in social science research. A common workflow involves three stages: (1) fine-tuning pre-trained LLMs on domain-specific datasets to improve performance on specialized tasks, (2) using the fine-tuned models to extract latent variables or classify text data, and (3) incorporating these outputs as outcomes or covariates in downstream statistical inference.

However, this approach ignores uncertainty quantification in the LLM output estimations, potentially leading to invalid statistical inference. This is especially problematic because fine-tuning on small, domain-specific datasets, while necessary for task alignment, often produces overconfident predictions. The limited size of typical annotation datasets (often hundreds or thousands of examples) relative to the model's billions of parameters creates conditions where models memorize training patterns and become artificially overconfident about unseen cases.

In this research, we propose a framework that properly accounts for uncertainty from LLM fine-tuning in downstream statistical inference. Our approach jointly estimates LLM fine-tuning parameters and statistical model parameters within a Bayesian framework, rather than the conventional three-stage approach that treats LLM outputs as deterministic point estimates. Specifically, we obtain posterior distributions over fine-tuning parameters using tractable approximations, then propagate this uncertainty to statistical models, ensuring valid statistical inference. (We don't have a solution yet)

A key advantage of our approach is that it enables joint estimation of fine-tuning parameters specifically for the downstream statistical task. Traditional workflows treat fine-tuning and statistical analysis as separate stages, models are first fine-tuned on domain-specific classification or regression tasks, then the resulting features are used in entirely different downstream analyses. This separation ignores the potential mismatch between the fine-tuning objective and the research question. Our unified framework allows the fine-tuning process to be jointly optimized with the statistical model, enabling the LLM to learn representations that are specifically tailored to the downstream statistical task rather than just the intermediate labeling task. This joint estimation not only improves predictive performance but also ensures that the uncertainty quantification reflects the specific requirements of the statistical analysis, leading to more reliable statistical inference for the research question of interest.

2 Problem Setting

2.1 LLM Output with Uncertainty Quantification

Fine-tuning pre-trained LLMs often employs parameter-efficient methods like Low-Rank Adaptation (LoRA) (Hu et al., 2022). Standard LoRA fine-tuning proceeds as follows: the pre-trained weights of LLM, W_0, are adapted using low-rank decomposition $W = W_0 + BA$ where $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times n}$ with rank $r \ll \min(m, n)$. During fine-tuning, only the LoRA parameters $\theta_{LoRA} = [A, B]$ are optimized on a domain-specific dataset $D_{train} = \{(x_i, y_i)\}_{i=1}^N$ to obtain point estimates $\hat{A}, \hat{B}$. The fine-tuned model is then applied deterministically for feature extraction: $z_i = f_{LLM}(x_i; W_0 + \hat{B}\hat{A})$ where $D_{text} = \{x_i\}_{i=1}^n$ is raw text data and $z_i \in \mathbb{R}^d$.

However, this approach ignores uncertainty in the fine-tuning parameters, particularly problematic when fine-tuning on small datasets. Bayesian fine-tuning (Yang et al., 2023; Wang et al., 2024) addresses this by characterizing the full posterior distribution over parameters:

$$p(\theta_{LoRA}|D_{train}) \propto p(D_{train}|\theta_{LoRA}, W_0)p(\theta_{LoRA}) \tag{1}$$

where $p(D_{train}|\theta_{LoRA}, W_0)$ is the likelihood of the fine-tuning data and $p(\theta)$ is the prior over parameters. For feature extraction, instead of deterministic outputs z_i, we obtain posterior predictive distributions that integrate over parameter uncertainty:

$$p(z_i|x_i, D_{train}) = \int p(z_i|x_i, W_0 + \theta_{LoRA})p(\theta_{LoRA}|D_{train})d\theta_{LoRA} \tag{2}$$

This posterior predictive distribution properly reflects uncertainty from the fine-tuning process, providing calibrated feature representations that account for the limited size of domain-specific training data.

2.2 Inference with LLM outputs: As Covariates

We first consider the case where LLM outputs are used as covariates in downstream statistical models. When covariates are measured with error, this leads to attenuation bias—coefficient estimates are biased toward zero—if errors are independent of the input data and can result in underestimating the true effects of LLM-derived variables on outcomes of interest. However, this classical measurement error framework does not apply to fine-tuned LLMs, where prediction errors systematically depend on the training dataset used for fine-tuning. Fine-tuning on small, domain-specific datasets often creates overconfidence issues that lead to systematic prediction biases rather than independent measurement errors. The resulting measurement error violates the independence assumption, meaning bias can occur in either direction depending on how the limited training data relates to the broader application domain.

The standard downstream regression with LLM outputs as covariates perform $Y_i = z_i^T\gamma + Q_i^T\beta + \epsilon_i$, where $Q \in \mathbb{R}^{n\times p}$ is structured covariates, and $Y \in \mathbb{R}^n$ is an outcome variable. This approach considers z_i, LLM outputs, as deterministic data, thus ignoring uncertainty in the LoRA parameters θ_{LoRA} from fine-tuning and leading to invalid statistical inference.

We propose a unified Bayesian framework that addresses this by jointly modeling both the feature extraction and statistical modeling stages. Rather than treating z_i as observed covariates, we recognize that z_i are uncertain predictions from a fine-tuned LLM with parameter uncertainty. The joint model specifies:

$$\begin{aligned}
\text{Feature extraction:} &\quad p(z_i|x_i, \theta_{LoRA}, W_0) && (3)\\
\text{Downstream regression model:} &\quad Y_i|z_i, Q_i, \gamma, \beta \sim N(z_i^T\gamma + Q_i^T\beta, \sigma^2) && (4)\\
\text{LoRA parameter posterior:} &\quad p(\theta_{LoRA}|D_{train}) \propto p(D_{train}|\theta_{LoRA}, W_0)p(\theta_{LoRA}) && (5)
\end{aligned}$$

The joint inferential target:

$$p(\gamma, \beta, \sigma^2, \theta_{LoRA}|Y, Q, D_{train}) \propto \tag{6}$$
$$\prod_i p(Y_i|z_i, Q_i, \gamma, \beta, \sigma^2)p(z_i|x_i, \theta_{LoRA}, W_0)p(\theta_{LoRA}|D_{train})p(\gamma, \beta, \sigma^2)$$

This joint posterior properly accounts for uncertainty propagation from fine-tuning through to statistical inference, enabling valid statistical inferences that reflect the true uncertainty in LLM-derived measurements.

2.3 Inference with LLM outputs: As Outcomes

We next consider the alternative case where LLM-generated measures serve as dependent variables in statistical models. Under classical assumptions, where measurement errors are independent of the regressors and mean-zero, errors in the dependent variable do not bias regression coefficients but do inflate the residual variance, leading to larger standard errors and reduced statistical power. However, this classical framework is often violated when LLMs are used as predictive instruments for outcomes. Fine-tuned LLMs, especially those trained on limited or domain-specific data, may exhibit systematic errors that correlate with the input variables or vary in structure across contexts. These violations can introduce endogeneity-like concerns, resulting in biased coefficient estimates and invalid inference. In particular, biased or structured prediction errors from LLMs may confound the relationship between covariates and outcomes, undermining the reliability of statistical conclusions drawn from such models.

The standard approach estimates $z_i = Q_i^T \beta + \epsilon_i$, where z_i represents the LLM-generated measure of interest and Q_i are explanatory variables. However, researchers observe $\hat{z}_i$ rather than the true construct z_i, where $\hat{z}_i$ contains measurement error from the fine-tuning process. This measurement error in the dependent variable contaminates all coefficient estimates β, and leads to inconsistent standard errors that invalidate hypothesis testing and confidence intervals.

Our unified Bayesian framework addresses this by jointly modeling the statistical relationship and the LLM measurement process. Rather than treating the observed LLM outputs $\hat{z}_i$ as the true dependent variable, our approach recognizes that we observe noisy measurements of an underlying latent construct z_i. The joint model specifies:

$$\text{Statistical relationship:} \quad z_i|Q_i, \beta, \sigma^2 \sim N(Q_i^T \beta, \sigma^2) \tag{7}$$

$$\text{LLM measurement process:} \quad \hat{z}_i|x_i, z_i, \theta_{LoRA} \sim p(\hat{z}_i|x_i, z_i, W_0 + \theta_{LoRA}) \tag{8}$$

$$\text{LoRA parameter posterior:} \quad p(\theta_{LoRA}|D_{train}) \propto p(D_{train}|\theta_{LoRA}, W_0)p(\theta_{LoRA}) \tag{9}$$

The joint inferential target:

$$p(\beta, \sigma^2, Z, \theta_{LoRA}|\hat{Z}, Q, x_i, D_{train}) \propto \tag{10}$$
$$\prod_i p(z_i|Q_i, \beta, \sigma^2)p(\hat{z}_i|x_i, z_i, \theta_{LoRA})p(\theta_{LoRA}|D_{train})p(\beta, \sigma^2)$$

where z_i are treated as latent variables to be estimated jointly with the regression parameters β and LoRA parameters θ_{LoRA}. This approach enables valid inference about the factors driving the latent construct while properly accounting for both the uncertainty in measuring z_i from text and the parameter uncertainty from fine-tuning.

References

Battaglia, L., Christensen, T., Hansen, S., and Sacher, S. (2025). Inference for regression with variables generated by ai or machine learning.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3. https://arxiv.org/abs/2106.09685

Yang, A. X., Robeyns, M., Wang, X., and Aitchison, L. (2023). Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv:2308.13111*. https://arxiv.org/abs/2308.13111

Wang, Y., Shi, H., Han, L., Metaxas, D., and Wang, H. (2024). Blob: Bayesian low-rank adaptation by backpropagation for large language models. *Advances in Neural Information Processing Systems*, 37:67758–67794. https://proceedings.neurips.cc/paper_files/paper/2024/hash/2eebfcf61fdf4fd8ae1aa1c1dbc50066-Abstract-Conference.html

Zhang, J., Xue, W., Yu, Y., and Tan, Y. (2025). Debiasing ml-or ai-generated regressors in partial linear models. Available at SSRN 4636026.

A Bias and Inference Issues of Ignoring Fine-Tuning Uncertainty

The LLM output uncertainty comes from two sources: (i) baseline uncertainty, reflecting errors inherent in the pre-trained model $f_{LLM}(x_i; W_0)$, and (ii) fine-tuning uncertainty, reflecting the variability induced by adapting to a small domain-specific dataset.

Let the latent construct of interest (e.g., sentiment, topic proportion, policy position) be denoted by z_i. When we apply a pre-trained LLM with parameters W_0 to text x_i, we obtain an approximation $z_i^{(0)} = f_{LLM}(x_i; W_0) = z_i + u_i^{(0)}$, where $u_i^{(0)}$ represents baseline error from the pre-trained model. After fine-tuning with LoRA on a small domain dataset D_{train}, we obtain $\tilde{z}_i = f_{LLM}(x_i; W_0 + \theta_{LoRA}) = z_i^{(0)} + u_i^{(ft)} = z_i + u_i^{(0)} + u_i^{(ft)}$, where $u_i^{(ft)}$ captures fine-tuning uncertainty.

In principle, both $u_i^{(0)}$ and $u_i^{(ft)}$ contribute to measurement error. However, in this work, we focus on $u_i^{(ft)}$ and consider $u_i^{(0)}$ as absorbed into the latent construct z_i. The motivation is twofold: (i) baseline error from pretraining is essentially unidentifiable without access to the massive training corpus, and (ii) fine-tuning error is tractable and often dominates uncertainty when adapting to small domain datasets. By specifying priors on the LoRA parameters, θ_{LoRA}, and conditioning on the domain-specific dataset D_{train}, its posterior distribution can be approximated, making this error both estimable and consequential for downstream inference.

With this simplification, we write the observed LLM-derived surrogate as $\tilde{z}_i = z_i + u_i$, where $u_i \equiv u_i^{(ft)}$ denotes the fine-tuning error. Importantly, u_i may be correlated with the latent construct z_i, the structured covariates Q_i, and even the regression error ϵ_i.

For example, suppose z_i is the true stance of a political text, and the fine-tuned LLM was trained on a small, imbalanced dataset with mostly liberal examples. Then the fine-tuned model tends to systematically "over-predict" liberal stance, especially for borderline cases. In this case, the fine-tuning error u_i is larger (and systematically biased) when z_i is conservative, $\text{Cov}(u_i, z_i) \neq 0$. Similarly, if a structured covariate Q_i captures demographic attributes (e.g., age or region) that affect writing style, fine-tuning errors may be larger for certain groups, leading to $\text{Cov}(u_i, Q_i) \neq 0$. Finally, if the same unobserved factors affect both the downstream regression and misclassification by the LLM, then $\text{Cov}(u_i, \epsilon_i) \neq 0$.

We now examine the statistical consequences of ignoring this uncertainty in downstream regression analysis. We use the residual-maker $M_Q = I - Q(Q'Q)^{-1}Q'$ to partial-out Q. For any vector v, denote $\dot{v} = M_Q v$.

A.1 LLM outputs used as covariates

True model: $Y_i = z_i^T \gamma + Q_i^T \beta + \epsilon_i$, $E[\epsilon_i | z_i, Q_i] = 0$.

But we regress on $\tilde{z}_i$ (and Q_i), ignoring uncertainty in $\tilde{z}_i$. First, we apply the residual maker to all variables:

- $\dot{z} = M_Q \tilde{z}$ (LLM outputs after partialling out Q),
- $z^* = M_Q z$ (true values after partialling out Q),
- $\dot{u} = M_Q u$ (error after partialling out Q),
- $\dot{\epsilon} = M_Q \epsilon$ (regression errors after partialling out Q).

The OLS estimator for γ: $\hat{\gamma} = (\dot{z}'\dot{z})^{-1}\dot{z}'\tilde{Y}$, where $\tilde{Y} = \gamma z^* + \dot{\epsilon}$. By substituting the true model:

$$\hat{\gamma} = ((z^* + \dot{u})'(z^* + \dot{u}))^{-1}(z^* + \dot{u})'(\gamma z^* + \dot{\epsilon}). \tag{11}$$

A.2 LLM outputs used as outcomes

[The mathematical derivation would continue here following the same pattern from the original document]