

GenAI-Powered Inference

Kosuke Imai

Harvard University

Machine-Generated Experimental Designs and The Future of Social Science

Columbia University

March 31, 2026

Joint work with Kentaro Nakamura (Harvard Kennedy School)

Motivation

- Generative AI is transforming medicine, education, marketing, etc.
- Can methodologists get some help from GenAI too?

How can a methodologist use LLM to help improve research? Give me a short answer.



A methodologist can use large language models (LLMs) to enhance research by automating literature reviews, generating hypotheses, designing study frameworks, and analyzing data. LLMs can also assist in drafting and refining research papers, coding qualitative data, and even suggesting relevant methodologies or statistical approaches, making the research process faster and more rigorous.

Using GenAI to Improve Statistical Inference with Unstructured Data

- GenAI-Powered Inference (GPI)
 - ① generate new unstructured data (or regenerate existing unstructured data) at scale
 - ② obtain a “true” internal representation from GenAI
 - ③ use it directly for machine learning without fine tuning GenAI

- Advantages:
 - no need to estimate representation
 - avoid functional form assumptions
 - better empirical performance

Generative AI: Definition and Assumption

- Deep generative model:

$$\mathbb{P}(\mathbf{X}_i \mid \mathbf{h}_\gamma(\mathbf{R}_i)),$$
$$\mathbb{P}(\mathbf{R}_i \mid \mathbf{P}_i).$$

- \mathbf{P}_i : prompt
 - \mathbf{X}_i : unstructured object generated by GenAI
 - \mathbf{R}_i : internal representation (hidden states of neural network)
 - $\mathbf{h}_\gamma(\mathbf{R}_i)$: deterministic function, mapping hidden states to the last layer
- **Deterministic decoding:**

$\mathbb{P}(\mathbf{X}_i \mid \mathbf{h}_\gamma(\mathbf{R}_i))$ is degenerate

- can be achieved by setting a hyperparameter
- use of open-source GenAI and deterministic encoding for replicability

Motivating Application: Texts-as-Treatments

- Candidate Biography Experiment (Fong and Grimmer, 2016)
 - 1246 biographies of American politicians scraped from Wikipedia
 - 1,886 voters as respondents
 - randomly assign biographies to voters
 - feeling thermometer $[0, 100]$ as the outcome
- Analysis
 - supervised topic model to discover 10 treatment features
 - estimate the average treatment effects of estimated topic proportions
- Existing approaches for texts-as-treatments:
 - 1 model-based approach (e.g., Egami *et al.* 2022; Fong and Grimmer, 2023)
 - 2 causal representation learning based on fine-tuned BERT embedding (e.g., Veitch *et al.* 2020; Pryzant *et al.* 2021; Gui and Veitch, 2023)

Example Biographies

Candidate biography with military background

Anthony Higgins was born in Red Lion Hundred in New Castle County, Delaware. He attended Newark Academy and Delaware College, and graduated from Yale College in 1861, where he was a member of Skull and Bones. After studying law at the Harvard Law School, he was admitted to the bar in 1864 and began practice in Wilmington, Delaware. He also served for a time in the United States Army in 1864.

Candidate biography without military background

Benjamin Tappan was born in Northampton, Massachusetts, the second child and oldest son of Benjamin Tappan and Sarah (Homes) Tappan, who was a grandniece of Benjamin Franklin. Two of his younger brothers were abolitionists Arthur Tappan and Lewis Tappan. He attended the public schools in Northampton and traveled to the West Indies in his youth. He apprenticed as a printer and engraver, also studying painting with Gilbert Stuart. He read law to be admitted to the bar in Hartford, Connecticut, in 1799. Later that year, he moved to the Connecticut Western Reserve and founded what is now Ravenna, Ohio, laying out the original village in 1808. He married, March 20, 1801, Nancy Wright, sister of John C. Wright (congressman), afterwards a United States House of Representatives from Ohio. They had one son, Benjamin, born in 1812.

Setup

- Notation

- $Y_i(\mathbf{x})$: Potential outcome when exposed to treatment object \mathbf{x}
- $Y_i = Y_i(\mathbf{X}_i)$: Outcome (collected from the survey respondents)
- T_i : Binary treatment feature (e.g., military experiences)
- \mathbf{U}_i : Confounding features (e.g., college education)

- Assumptions

- ① Treatment Feature:

$$T_i = g_T(\mathbf{X}_i)$$

- ② Confounding Features:

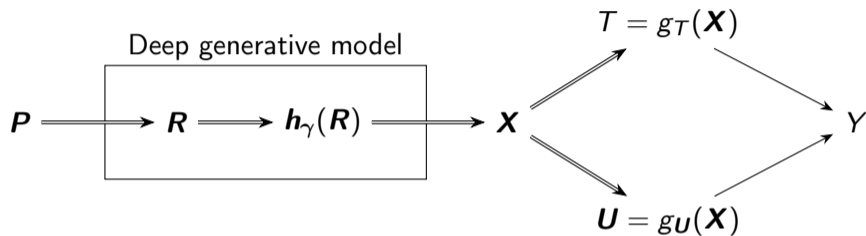
$$\mathbf{U}_i = \mathbf{g}_U(\mathbf{X}_i) \quad \text{where } \dim(\mathbf{U}_i) \ll \dim(\mathbf{X}_i)$$

- ③ Separability:

$$Y_i(\mathbf{x}) = Y_i(g_T(\mathbf{x}), \mathbf{g}_U(\mathbf{x}))$$

- ① T is not a function of \mathbf{U}
 - ② \mathbf{U} is not a function of T

Summary of Assumptions



- Assumptions:
- **Overlap:** The above assumptions imply that for any $t \in \{0, 1\}$ and $\mathbf{u} \in \mathcal{U}$, we have

$$\mathbb{P}(T_i = t \mid \mathbf{U}_i = \mathbf{u}) > 0.$$

- In practice, one must monitor the estimated propensity score for overlap violation

Nonparametric Identification

- Average treatment effect (ATE):

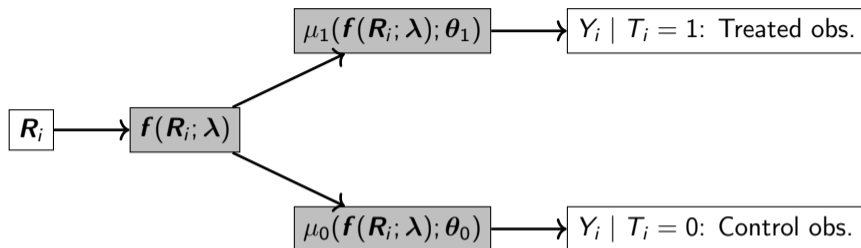
$$\tau := \mathbb{E}[Y_i(1, \mathbf{U}_i) - Y_i(0, \mathbf{U}_i)]$$

- Under these assumptions, there exists a **Deconfounder** $\mathbf{f} : \mathbb{R}^r \rightarrow \mathbb{R}^q$ with $q \leq r$ such that

$$Y_i \perp\!\!\!\perp \mathbf{R}_i \mid T_i = t, \mathbf{f}(\mathbf{R}_i), \quad t \in \{0, 1\}$$

- Deconfounder does not have to be unique
 - Example: Confounding Features \mathbf{U}_i (deterministic function of \mathbf{R}_i)
-
- By adjusting for this Deconfounder, we can identify the ATE
 - Direct adjustment for \mathbf{R}_i leads to the lack of overlap

Estimation and Inference



- 1 Estimate the outcome models and deconfounder via TarNet (Shalit et al. 2017):

$$\{\hat{\lambda}, \hat{\theta}_0, \hat{\theta}_1\} = \underset{\lambda, \theta_0, \theta_1}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \{Y_i - \mu_{T_i}(f(R_i; \lambda); \theta_{T_i})\}^2$$

- 2 Estimate the propensity score using the estimated Deconfounder

$$\pi(f(R_i, \hat{\lambda})) = \mathbb{P}(T_i = 1 | f(R_i, \hat{\lambda}))$$

- 3 Estimate ATE by Double Machine Learning (DML) (Chernozhukov et al. 2018)

Practical Implementation Details

- Internal representation extracted from LLM is still high-dimensional:

$$\dim(\mathbf{R}) = \text{number of tokens} \times 4096 \quad \text{for Llama 3 (8 billion parameters)}$$

- Pooling strategies depend on deep generative models
 - Llama 3: the hidden states of the last token
- TarNet requires hyperparameter tuning
 - size and depth of layers
 - learning rate
 - maximum epoch size
- Use of automatic hyperparameter optimization methods (e.g., Optuna)

Simulation Study Setup

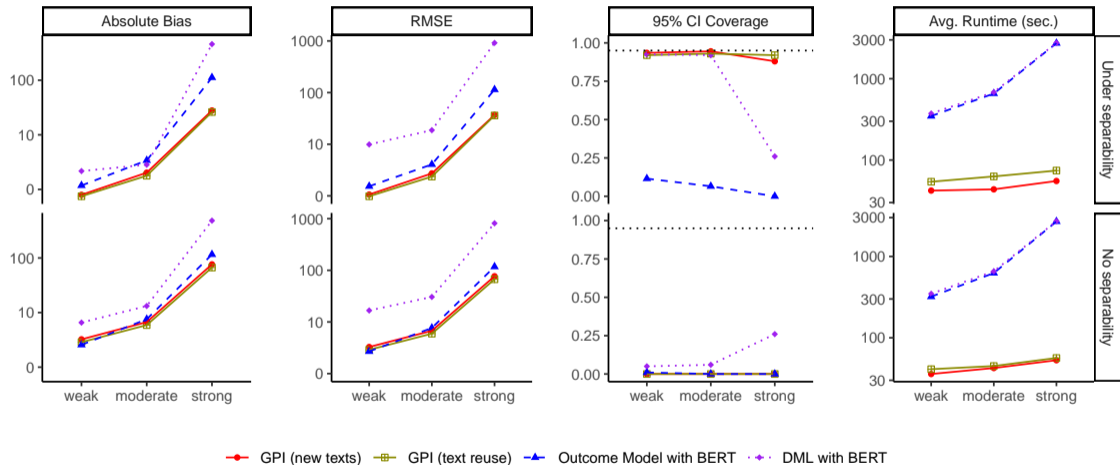
- A simulation based on the candidate biography experiment
 - Create 4,000 sets of the first, middle, and last names of political candidates via randomly sampling from the Fong and Grimmer data
 - Use Llama 3 to generate a biography for each US political candidate's
 - Instruct LLM to repeat the same texts for reuse
- The data generating process:

$$Y_i = \alpha_1 T_i + \alpha_2 T_i h_1(\mathbf{X}_i) - \alpha_3 h_1(\mathbf{X}_i) - \alpha_4 h_2(\mathbf{X}_i) + \epsilon_i$$

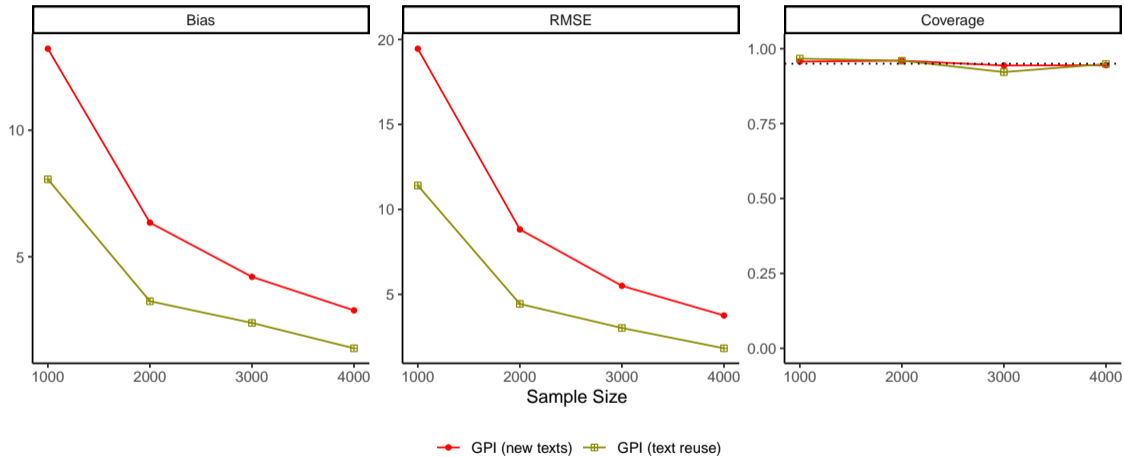
where $\epsilon_i \sim \mathcal{N}(\mu_i, 1)$ and

- T_i : military background (binary)
 - $h_1(\mathbf{X}_i)$: topic-model based confounder
 - $h_2(\mathbf{X}_i)$: sentiment-analysis based confounder
- $2 \times 3 = 6$ scenarios:
 - ① separability holds or does not hold (separate or overlapping topics)
 - ② weak, medium, or strong confounding

Simulation Results



Performance across Different Sample Sizes



Empirical Analysis

- Analyze the original survey by Fong and Grimmer (2016)
 - 1,246 Congressional candidate biographies from Wikipedia
 - 1,886 survey participants with a total of 5,291 observations
 - evaluate a biography using the feeling thermometer [0, 100]
 - Keyword-based treatment coding: “military”, “war”, “veteran”, or “army”
 - use text-reuse approach with Llama 3

Methods	ATE	95% Conf. Int.	Runtime
Proposed method (reuse)	5.462	[2.790, 8.135]	28.9 sec.
Outcome model with BERT	-2.557	[-2.608, -2.505]	6139.7
DML with BERT	-67.777	[-109.967, -25.587]	6210.3

Predictive Effects of Image Features (Lindholm et al. 2024)

- How does the visual appearance of political candidate predict their electoral success?
- Data: 7,080 Danish politicians with candidate photos



- Prediction variables: facial features (continuous scores; discretized into 10 bins)
 - 1 attractiveness
 - 2 trustworthiness
 - 3 dominance
- Outcome: Election results (number of votes standardized via z-score)
- Structured confounding variables: age, gender, education
- We wish to adjust other confounding features

Empirical Analysis

- Reproduce all images using **Stable diffusion** (ver. 1.5)
- Original image:

$$\dim(\mathbf{X}) = 304(\text{width}) \times 304(\text{height}) \times 3(\text{RGB}) = 277,248$$

- Internal representation:

$$\dim(\mathbf{R}) = 16,384$$

- Neural network architecture:

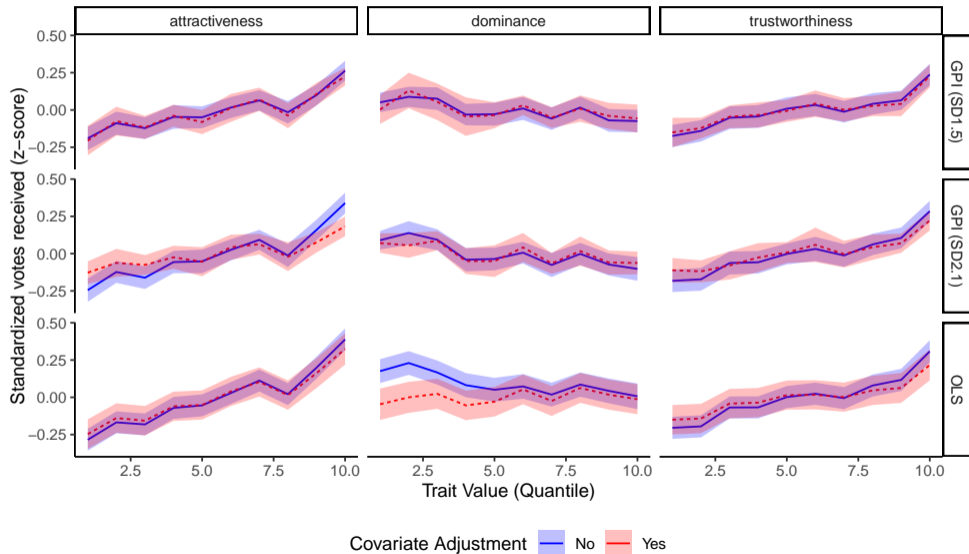
- $\dim(\mathbf{f}(\mathbf{R})) = 1024$
- depth of hidden layers = 2
- size of hidden layers after deconfounder = [200, 1]

- Nonparametrically estimate the average predictive effect

$$\xi_t := \mathbb{E}[Y_i(t, \mathbf{U}_i)]$$

- DML with a multi-valued treatment

Empirical Results



Concluding Remarks

- Generative AI can be used to improve causal inference
 - can generate treatments at scale
 - enables the extraction of true internal representation
 - better causal representation learning

- Open-source software **GPI** is available at

<https://gpi-pack.github.io/>

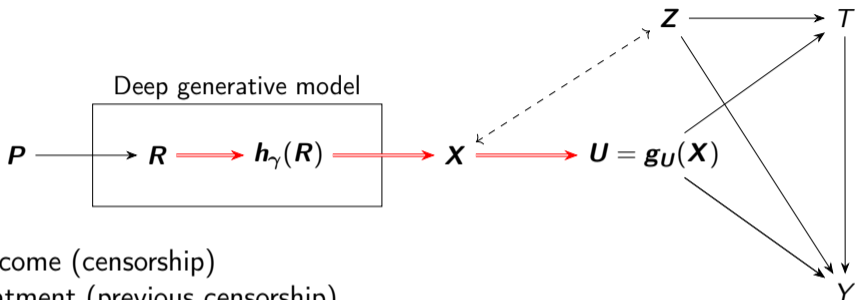
- Further extensions
 - causal inference with multimodal data (e.g., videos)
 - interpretation of estimated deconfounder
 - discovery of treatment concepts
 - policy learning with unstructured treatments

Appendix

Text as Confounder: Chinese Censorship (Roberts et al. 2020)

- Do Chinese social media users who had their post censored become more likely to be censored for later posts or self-censor themselves?
 - Treatment: whether or not a post was censored
 - Outcomes: censorship during four weeks after a censored post
 - 1 number of posts
 - 2 proportion of censored posts
 - 3 proportion of missing posts
 - structural confounders: lagged outcomes, date of the post (dummies)
 - text-as-confounder: contents of posts
- Original analysis: Matching (CEM) with topic proportions (STM) and propensity score (inverse regression)
- Our reanalysis:
 - Text reuse with Llama 3
 - Apply the proposed method:
 - 1 entire sample (4155 users; 75324 Weibo posts)
 - 2 matched sample (628 users; 879 posts)

Assumptions



- Y : outcome (censorship)
- T : treatment (previous censorship)
- Z : observed structured confounding variables
- X : unstructured confounding object
- $U = g_U(X)$: unknown and unstructured confounding variables
- **Strong latent ignorability:**

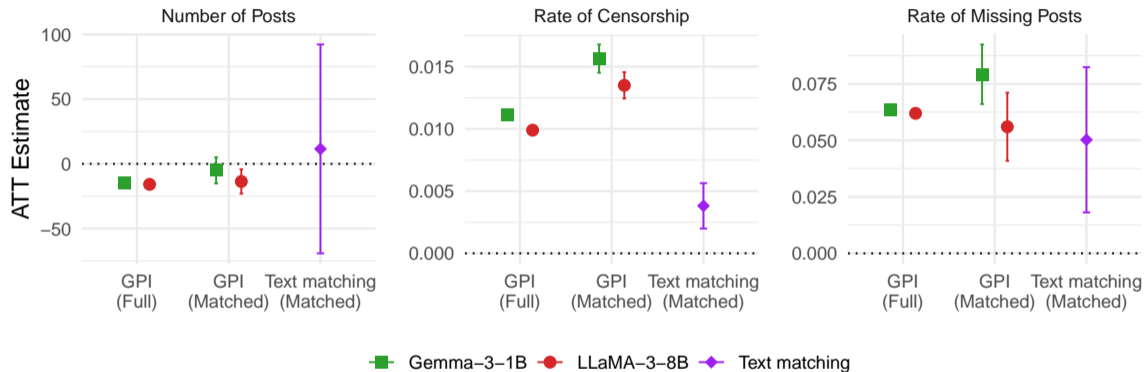
$$\{Y_i(t)\}_{t \in \mathcal{T}} \perp\!\!\!\perp T_i \mid \mathbf{Z}_i = \mathbf{z}, \mathbf{U}_i = \mathbf{u}, \quad \text{for all } \mathbf{z} \in \mathcal{Z}, \mathbf{u} \in \mathcal{U}$$

$$\mathbb{P}(T_i = t \mid \mathbf{Z}_i = \mathbf{z}, \mathbf{U}_i = \mathbf{u}) > 0 \quad \text{for all } t \in \mathcal{T}, \mathbf{z} \in \mathcal{Z}, \mathbf{u} \in \mathcal{U}$$

Empirical Analysis

- Reproduced all the texts using open-source LLaMa3–8B
- Internal representation: last token of the final layer, $\dim(\mathbf{R}) = 4080$
- Automated hyperparameter tuning via Optuna (Akiba et al. 2019)
 - $\dim(\mathbf{f}(\mathbf{R})) = 2048$
 - depth of hidden layers = 2
 - size of hidden layers after deconfounder = [50, 1]
- 2-fold cross-fitting:
 - clustered standard errors at the user level
 - truncation of extreme propensity scores (Dorn, 2025)

Empirical Results



- Our analysis shows higher rates of censorship and self-censorship
- Full sample analysis is much more efficient

Residual Correlations with Candidate Confounder

- Candidate confounder: proportion of 60 keywords related to censorship and self-censorship (Fu et al. 2013)
- Spearman's rank correlation between GPI's efficient score and the candidate confounder
- Residual correlation from the implied weighted OLS for text matching
- p -value in parentheses

Outcome	GPI (LLaMA3-8B)		GPI (Gemma3-1B)		Text matching
	Full	Matched	Full	Matched	Matched
Number of posts	0.004 (0.259)	0.028 (0.421)	0.000 (0.849)	-0.036 (0.318)	0.039 (0.078)
Rate of censorship	-0.001 (0.779)	0.020 (0.580)	0.001 (0.676)	0.007 (0.189)	0.074 (0.001)
Rate of missing posts	-0.002 (0.612)	-0.038 (0.307)	0.005 (0.102)	-0.012 (0.777)	-0.022 (0.344)

Structural Model of Texts: Persuasion and Rhetoric

(Blumenau and Lauderdale, 2022)

- Which types of political rhetorics are most persuasive?
- Forced choice conjoint experiment with texts
- Total of 336 political arguments
 - 12 policy issues: tuition fees, fracking, etc.
 - 14 rhetorical elements: cost and benefit, morality, etc.
 - for or against
- Outcome: Persuasiveness of arguments
 - one argument is more persuasive than the other
 - equally persuasive

Example Text Pair

- Policy topic: building a third runway at Heathrow:

Appeal to authority / For

The Airports Commission, an independent body established to study the issue, have argued that expanding Heathrow is the most effective option to address the UK's aviation capacity challenge

Appeal to history / Against

History show us that most large infrastructure projects do not lead to significant economic growth, which suggests that the expansion of Heathrow will fail to pay for itself

- Can we adjust for the unstructured confounding features of texts?

The Structural Model

- The original **Bradley-Terry** type model:

$$\log \left[\frac{\mathbb{P}(Y_{jj'(i)} \leq k)}{\mathbb{P}(Y_{jj'(i)} > k)} \right] = \delta_k + (\alpha_{P_j S_j} + \beta_{T_j} + \gamma_j) - (\alpha_{P_{j'} S_{j'}} + \beta_{T_{j'}} + \gamma_{j'})$$

where i indexes respondents, j indexes arguments, P_j denotes policy area, S_j denotes for/against, and T_j denotes rhetoric

- Our **semiparametric model**:

$$\log \left[\frac{\mathbb{P}(Y_{j(i),j'(i)} \leq k)}{\mathbb{P}(Y_{j(i),j'(i)} > k)} \right] = \delta_k + \mu(T_j, \mathbf{U}_j) - \mu(T_{j'}, \mathbf{U}_{j'})$$

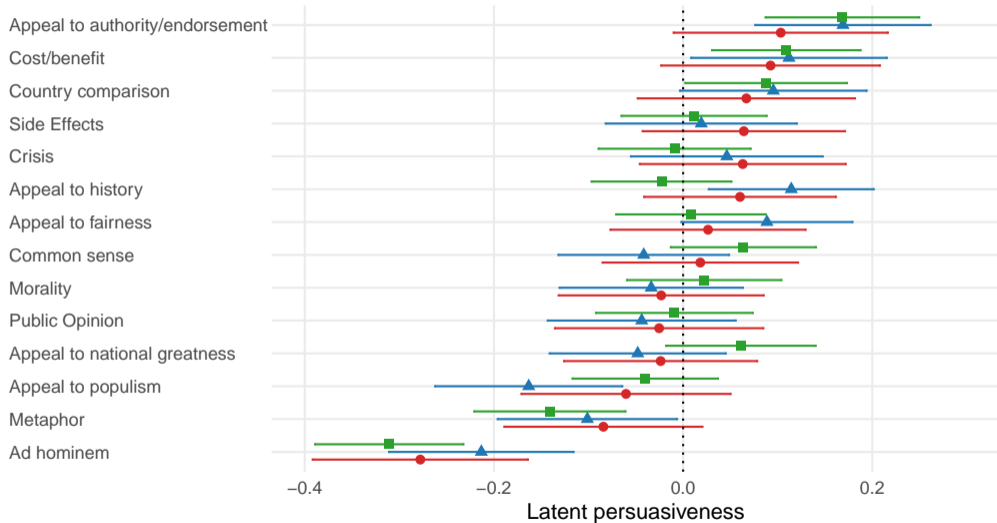
- Persuasiveness of rhetoric $T_j = t$

$$\beta(t) := \mathbb{E}[\mu(t, \mathbf{U}_j)]$$

- Estimate $\beta(t)$ using the deconfounder $\mathbf{f}(\mathbf{R}_j)$

- Reproduce all texts using **Llama3-8B**
- Internal representation: last token of the final layer, $\dim(\mathbf{R}) = 4096$
- Neural network architecture:
 - $\dim(\mathbf{f}(\mathbf{R})) = 1024$
 - depth of hidden layers = 2
 - size of hidden layers after deconfounder = [200, 1]
- Quantify uncertainty via Monte Carlo dropout (Gal and Ghahramani 2016)

Empirical Findings



Model ● LLaMA-3-8B ▲ LLaMA-3.3-70B ■ Gemma-3-1B