

Causal Representation Learning with Generative Artificial Intelligence: Application to Texts as Treatments

Kosuke Imai Kentaro Nakamura

Harvard University

Japanese Society for Quantitative Political Science

2025 Winter Meeting

January 5, 2025

Motivation

- Generative AI is transforming medicine, education, marketing, etc.
- Can methodologists get some help from generative AI 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 Generative AI to Improve Causal Inference

- LLM-assisted causal inference with unstructured data
 - 1 use LLM to generate treatment texts
 - 2 randomly assign generated texts to survey respondents and measure their reactions
 - 3 extract the internal representation of generated texts from LLM
 - 4 machine learning using the extracted true vector representation
 - estimate a **deconfounder** that summarizes all confounding information
 - double machine learning for valid statistical inference
- Advantage: no need to estimate the representation
 - nonparametric modeling: no topic modeling
 - computational efficiency: no fine tuning
 - better empirical performance
- Extension: instrumental variable approach to perceived treatments

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 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 embedding (e.g., Veitch *et al.* 2020; Pryzant *et al.* 2021; Gui and Veitch, 2023)

Using LLM to Generate Biographies (Llama 3)

Prompts	Generated candidate biography
<p>(System): You are a text generator who always produces a biography of the instructed person.</p> <p>(User): Create a biography of an American politician named Nathaniel C. Gilchrist</p>	<p>Nathaniel C. Gilchrist was an American politician who served as a member of the United States House of Representatives from Maryland's 4th congressional district from 1895 to 1899. Born on November 24, 1847, in Baltimore, Maryland, Gilchrist was the son of a prominent family of merchants and shipbuilders. He was educated at the Baltimore City College and later attended the University of Maryland, where he studied law. After graduating, he was admitted to the bar and began practicing law in Baltimore.</p>

- Requirements:

- ① open-source LLM: GPT, PaLM, Llama, OPT, etc.
- ② assume LLM is compliant with prompt or use human coder

Text Reuse: Using the Existing Texts as Treatments

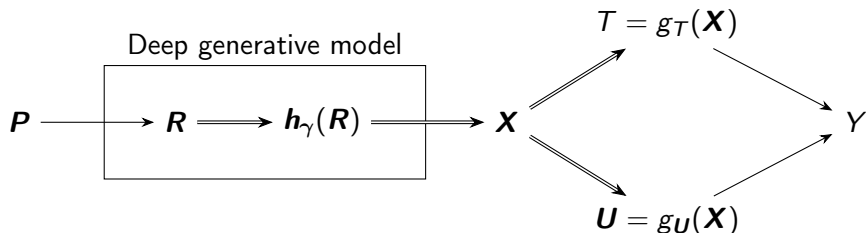
Prompts	Generated candidate biography
<p>(System): You are a text generator who just repeats the input text.</p> <p>(User): Higgins was born in Red Lion Hundred in New Castle County, Delaware. He attended Newark Academy and Delaware College, and graduated from Yale College in 1861, where he was a member 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.</p>	<p>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.</p>

- 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 : treatment object (generated texts)
 - \mathbf{R}_i : hidden states or internal representations
 - $\mathbf{h}_\gamma(\mathbf{R}_i)$: deterministic function from hidden states to last layer
-
- Other variables
 - $Y_i(\mathbf{x})$: Potential outcome when exposed to treatment object \mathbf{x}
 - Y_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



- **Separability:**

$$Y_i(\mathbf{X}_i) = Y_i(g_T(\mathbf{X}_i), g_U(\mathbf{X}_i)) = Y_i(T_i, U_i)$$

- Lemma: separability implies **overlap**

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

- **Deterministic decoding:** $\mathbb{P}(\mathbf{X}_i \mid h_\gamma(R_i))$ is degenerate

Nonparametric Identification

- Average treatment effect (ATE):

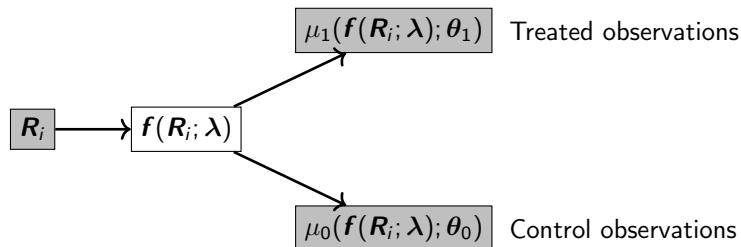
$$\tau := \mathbb{E}[Y_i(1, \mathbf{U}_i) - Y_i(0, \mathbf{U}_i)]$$

- 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

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

Double Machine Learning for asymptotic inference

Simulation Study Setup

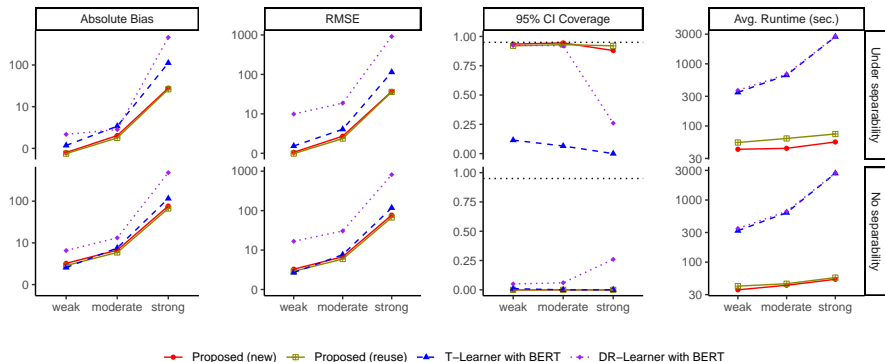
- A simulation based on the candidate biography experiment
 - Create 4,000 sets of the first, middle, and last names of political candidates via randomly sampling from the Fong and Grimmer data
 - Use Llama 3 to generate a biography for each US political candidate's
 - Instruct LLM to repeat the same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

$$Y_i = \alpha_1 T_i + \alpha_2 T_i h_1(\mathbf{X}_i) - \alpha_3 h_1(\mathbf{X}_i) - \alpha_4 h_2(\mathbf{X}_i) + \epsilon_i$$
$$\epsilon_i \sim \mathcal{N}(\mu_i, 1)$$

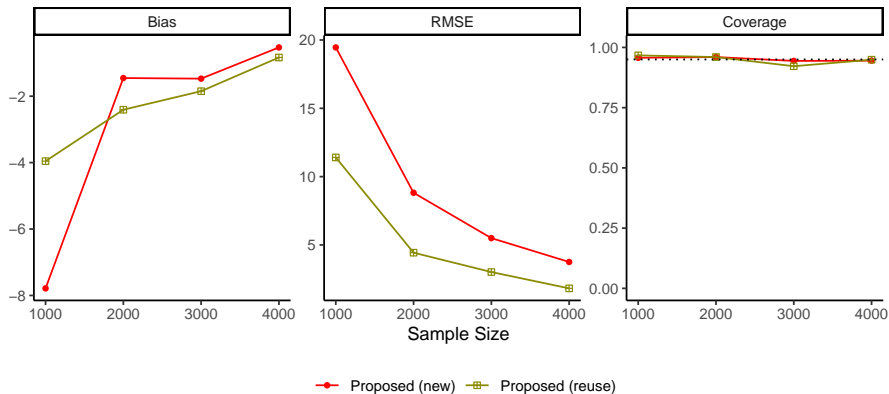
where

- 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 (sec.)
Proposed method (reuse)	5.462	[2.790, 8.135]	28.9
T-learner with BERT	-2.557	[-2.608, -2.505]	6139.7
DR-learner with BERT	-67.777	[-109.967, -25.587]	6210.3

Concluding Remarks

- Generative AI can be used to improve causal inference
 - generate treatments at scale
 - enables the extraction of true internal representation
 - better causal representation learning
- Further extensions
 - images and videos
 - interpretation of estimated deconfounder
 - discovery of treatment concepts
 - policy learning with unstructured treatments