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

Kosuke Imai

Harvard University

School of Political Economics, Waseda University
November 8, 2024

Joint work with Kentaro Nakamura

# 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 text 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 embeddings
  - nonparametric modeling
  - computational efficiency
  - better empirical performance
- Extension: instrumental variable approach to perceived treatments

# Motivating Application: Candidate Biography Experiment

(Fong and Grimmer, 2016)

- Survey experiment
  - 1246 biographies of American politicians from Wikipedia
  - 1,886 voters as respondents
  - randomly assign biographies to voters
  - feeling thermometer as the outcome
  - supervised topic model to discover 10 treatment features
  - estimate the average treatment effects of estimated topic proportions

- Existing methods 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)

**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.

# Using LLM to Generate Biographies (Llama 3)

| Prompts | Generated candidate biography |
|---|---|
| **(System)**: You are a text generator who always produces the biography of the instructed person.<br>**(User)**: Create a biography of an American politician named Nathaniel C. Gilchrist | 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. |

- 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 |
|---|---|
| **(System)**: You are a text generator who just repeats the input text.<br>**(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. | 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. |

# Setup

- Deep Generative Model

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

  - $\boldsymbol{P}_i$: prompt
  - $\boldsymbol{X}_i$: treatment object (generated texts)
  - $\boldsymbol{R}_i$: hidden states or internal representations
  - $\boldsymbol{h}_\gamma(\boldsymbol{R}_i)$: deterministic function from hidden states to last layer

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

# Assumptions

1. Consistency:
$$Y_i \;=\; Y_i(\boldsymbol{X}_i)$$
2. Randomization of Prompts:
$$Y_i(\boldsymbol{x}) \perp\!\!\!\perp \boldsymbol{P}_i$$
3. Treatment Feature:
$$T_i \;=\; g_T(\boldsymbol{X}_i)$$
4. Confounding Features:
$$\boldsymbol{U}_i \;=\; \boldsymbol{g_U}(\boldsymbol{X}_i) \quad \text{where } \dim(\boldsymbol{U}_i) \ll \dim(\boldsymbol{X}_i)$$
5. Separability:
$$Y_i(\boldsymbol{x}) = Y_i(g_T(\boldsymbol{x}), \boldsymbol{g_U}(\boldsymbol{x})),$$
where $g_T$ and $\boldsymbol{g_U}$ are separable such that there exist no function $\tilde{g}$ and $\tilde{\boldsymbol{g}}$ such that $g_T(\boldsymbol{x}) = \tilde{g} \circ \boldsymbol{g_U}(\boldsymbol{x})$ or $\boldsymbol{g_U}(\boldsymbol{x}) = \tilde{\boldsymbol{g}} \circ g_T(\boldsymbol{x})$

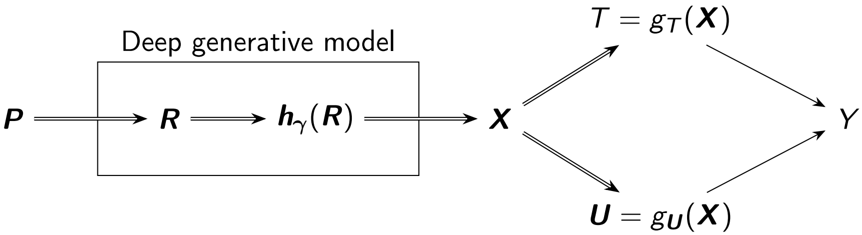
Assumptions 3, 4, and 5 imply that for any $t \in \{0, 1\}$ and $\boldsymbol{u} \in \mathcal{U}$,
$$\text{(overlap)} \quad \mathbb{P}(T_i = t \mid \boldsymbol{U}_i = \boldsymbol{u}) > 0.$$

6. Deterministic Decoding:

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

- stochastic decoding may induce dependence across observations
- it will also confound the treatment-outcome relation
- most LLMs have this option; greedy, beam, or contrastive searches
- exception includes diffusion models

Assumptions by picture:



# Nonparametric Identification Result

- Average treatment effect (ATE):

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

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

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

- Example: Confounding Features $\boldsymbol{U}_i$ (deterministic function of $\boldsymbol{R}_i$)
- By adjusting for this Deconfounder, we can identify the marginal distribution of potential outcome as

$$\mathbb{P}(Y_i(t, \boldsymbol{U}_i) = y) \;=\; \int_{\mathbb{R}^r} \mathbb{P}(Y_i = y \mid T_i = t, \boldsymbol{f}(\boldsymbol{R}_i)) dF(\boldsymbol{R}_i),$$

- Deconfounder does not have to be unique
- Direct adjustment for $\boldsymbol{R}_i$ leads to the lack of overlap

# Estimation and Inference

$$
\begin{tikzcd}
 & & \boxed{\mu_1(\boldsymbol{f}(\boldsymbol{R}_i;\boldsymbol{\lambda});\boldsymbol{\theta}_1)} \quad \text{Treated observations} \\
\boxed{\boldsymbol{R}_i} \arrow[r] & \boxed{\boldsymbol{f}(\boldsymbol{R}_i;\boldsymbol{\lambda})} \arrow[ur] \arrow[dr] & \\
 & & \boxed{\mu_0(\boldsymbol{f}(\boldsymbol{R}_i;\boldsymbol{\lambda});\boldsymbol{\theta}_0)} \quad \text{Control observations}
\end{tikzcd}
$$

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

$$\{\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\theta}}_0, \hat{\boldsymbol{\theta}}_1\} \;=\; \underset{\boldsymbol{\lambda},\boldsymbol{\theta}_0,\boldsymbol{\theta}_1}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\{Y_i - \mu_{T_i}(\boldsymbol{f}(\boldsymbol{R}_i;\boldsymbol{\lambda});\boldsymbol{\theta}_{T_i})\}^2$$

2. Estimate the propensity score using the estimated Deconfounder

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

Popular DragonNet (Shi et al. 2019) jointly estimates the outcome models, propensity score, and deconfounder, leading to the lack of overlap

# Double Machine Learning (Chernozhukov et al. 2018)

- Cross-fitting:
  1. randomly divide the data into $K$ folds
  2. for each $k = 1, \dots, K$, use the $k$th fold as the test set and the remaining $k-1$ folds as the training set
     1. randomly split the training set further into two subsets
     2. use the first subset to estimate outcome models and deconfounder
     3. use the second subset to estimate propensity score given the estimated deconfounder
  3. Compute the ATE estimator as:

$$
\begin{aligned}
\hat{\tau} \; = \; & \frac{1}{nK} \sum_{k=1}^{K} \sum_{i:I(i)=k} \hat{\mu}_1^{(-k)}(\hat{\boldsymbol{f}}^{(-k)}(\boldsymbol{R}_i)) - \hat{\mu}_0^{(-k)}(\hat{\boldsymbol{f}}^{(-k)}(\boldsymbol{R}_i)) \\
& + \frac{T_i\{Y_i - \hat{\mu}_1^{(-k)}(\hat{\boldsymbol{f}}^{(-k)}(\boldsymbol{R}_i))\}}{\hat{\pi}^{(-k)}(\hat{\boldsymbol{f}}^{(-k)}(\boldsymbol{R}_i))} - \frac{(1-T_i)\{Y_i - \hat{\mu}_0^{(-k)}(\hat{\boldsymbol{f}}^{(-k)}(\boldsymbol{R}_i))\}}{1-\hat{\pi}^{(-k)}(\hat{\boldsymbol{f}}^{(-k)}(\boldsymbol{R}_i))}
\end{aligned}
$$

- Double robustness, asymptotic normality

# Practical Implementation Details

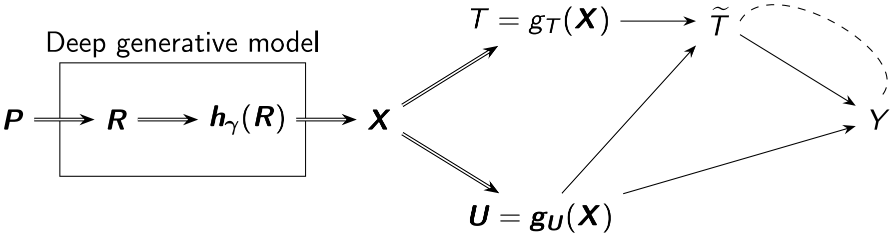
- Internal representation extracted from LLM is still high-dimensional:

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

- Pooling strategies depend on deep generative models
  - BERT: the first special classification token [CLS]
  - 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)

# Extension: Perceived Treatment via Instrumental Variables




- Local average treatment effect (LATE):

$$\mathbb{E}[Y_i(1, \boldsymbol{U}_i) - Y_i(0, \boldsymbol{U}_i) \mid \widetilde{T}(1, \boldsymbol{U}_i) = 1, \widetilde{T}(0, \boldsymbol{U}_i)]$$

- Separability assumption:

$$Y_i(\boldsymbol{x}) = Y_i(\widetilde{T}_i(g_T(\boldsymbol{x}), \boldsymbol{g_U}(\boldsymbol{x})), g_T(\boldsymbol{x}), \boldsymbol{g_U}(\boldsymbol{x}))$$

- Nonparametric identification under separability, monotonicity, exclusion restriction
- Estimation and inference with double machine learning

## Simulation Study Setup

- A simulation based on the candidate biography experiment
- Generate 4,000 US political candidates' profiles with Llama 3 by randomly sampling the first, middle, and last names from the Fong and Grimme data
- Instruct LLM to repeat the same texts for reuse
- The data generating process:

$$Y_i = 10 \times T_i + 10 \times TC_{1i} + \beta_{C1} C_{1i} - \beta_{C2} C_{2i} + \epsilon_i$$
$$\epsilon_i \sim \mathcal{N}(\mu_i, 1)$$

  where

  - $T_i$: military background (binary)
  - $C_{1i}$: topic-model based confounder
  - $C_{2i}$: 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: $\beta_{C1} = \beta_{C2} = 50$, 100, or 1000

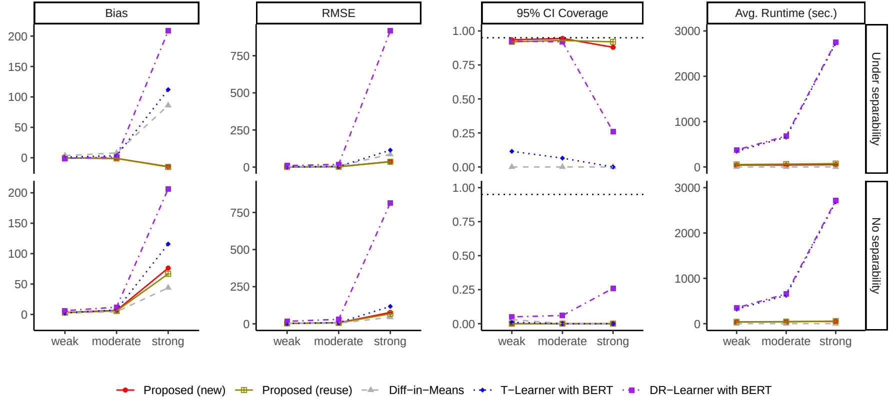
# Estimators to Be Compared

1. The proposed estimator:
   - neural network with one linear layer for the deconfounder with the output dimension of 2048
   - neural network with two consecutive linear layers with ReLU activation function for the outcome model
   - hyperparameter tuning based on the weak confounding setting
2. Estimators based on BERT embedding
   - $T$-learner (Pryzant et al. 2021)
   - $DR$-learner (Gui and Veitch 2023)

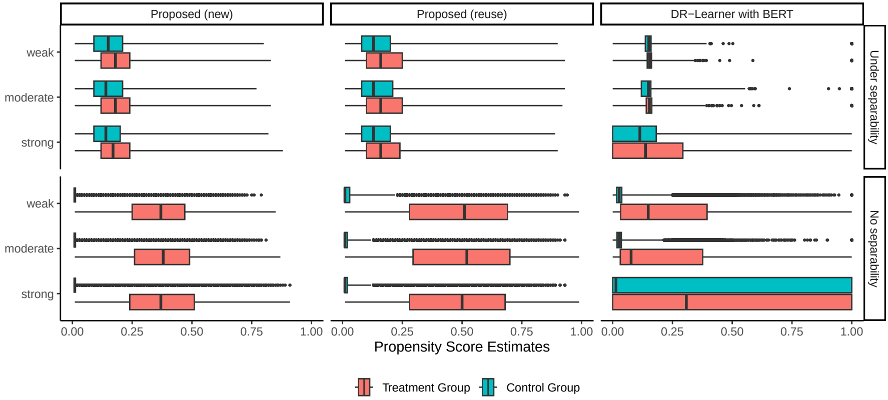
$$\text{Loss function} = \underbrace{\sum_{i=1}^{n} B(\boldsymbol{b}_{\text{full}}(\boldsymbol{X}_i))}_{\text{BERT fine-tuning}} + \underbrace{\frac{\lambda}{n}\sum_{i=1}^{n}\{Y_i - Q_{T_i}(\boldsymbol{b}(\boldsymbol{X}_i))\}^2}_{\text{outcome model}}$$
$$+ \underbrace{\frac{\alpha}{n}\sum_{i=1}^{n}\left\{T_i \log g(\boldsymbol{b}(\boldsymbol{X}_i)) + (1-T_i)\log[1-g(\boldsymbol{b}(\boldsymbol{X}_i))]\right\}}_{\text{propensity score}}$$

   - truncate propensity score at 0.01 and 0.99

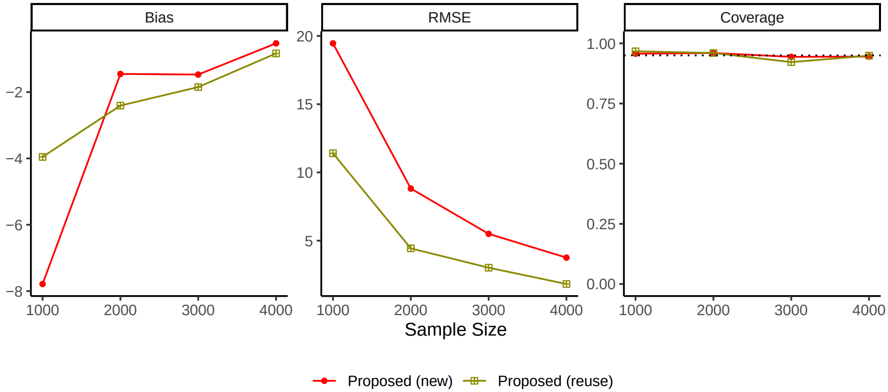
# Simulation Results



# Distribution of Estimated Propensity Score



# 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) | 3.09 | $[-1.17, \ 7.36]$ | 16.5 |
| $T$-learner with BERT | $-7.30$ | $[-7.36, -7.24]$ | 1837.0 |
| $DR$-learner with BERT | $-363.72$ | $[-444.44, -283.01]$ | 1898.7 |

# 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