Causal Inference with Video Features as Treatments

Kentaro Nakamura (Harvard) Adam Breuer (Dartmouth)
Michael H. Crespin (Oklahoma) Bryce J. Dietrich (Purdue)
Kosuke Imai (Harvard)

Annual Meeting of the Society for Political Methodology
Michigan State University
July 16, 2026

Motivation

- Video is a dominant medium of modern communication

- How do we conduct causal inference with video?
- Standard approach:
  1. classify videos into treatment and control groups based on features of interest
  2. randomly assign viewers to videos
  3. compare viewers' reactions to treated and control videos
- Challenge: treatment features are confounded with many other visual and auditory features

- How can we identify the causal effects of specific video features?
- How can we estimate their dynamic effects on viewers' reactions over time?
- No causal inference method is available for video data

Outline of Talk

1. Dynamic GenAI Powered Inference (Dynamic GPI)
   - generalize GPI of Imai and Nakamura (*JASA*, forthcoming) to the dynamic setting
   - leverage generative AI (GenAI) to adjust for confounding features

2. Validation study
   - Super Mario Bros.™ benchmark data with known ground truth
   - AI agent plays the game

3. Empirical study
   - 2020 Presidential campaign ads data
   - estimate the causal effects of candidate appearance on viewers' real-time rating

GenAI-Powered Inference (GPI)

- Causal and statistical inference framework for unstructured data
  1. leverage a GenAI model to (re)generate the unstructured data
  2. extract its internal representations
  3. estimate feature-specific causal effects using extracted representations

- Advantages
  1. representations contain *all* the information of generated video in a low-dimensional format
  2. use neural network to avoid functional-form assumptions
  3. asymptotic statistical guarantees with good performance under moderate sample sizes

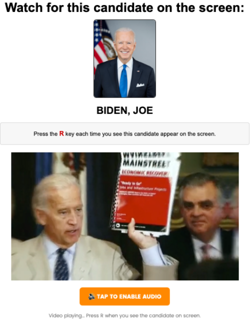
- We extend this framework to the dynamic settings with video data

Empirical Application: Presidential Campaign Advertisement

- How does the candidate's appearance in campaign ads affect a voter's real-time evaluation?

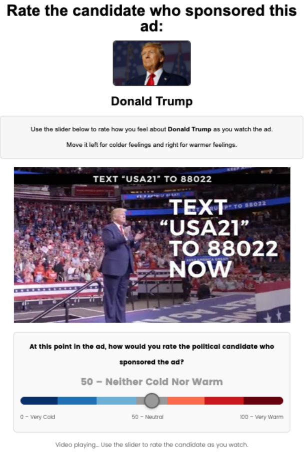
- Data: 2020 campaign videos from the Wesleyan Media Project
  - 849 videos (10–120 seconds; median: 30 seconds)
  - Joe Biden (389 videos) / Donald Trump (128 videos)

- Treatment feature: candidate appearance
  - survey conducted on Prolific
  - asked respondents to press the "R" key when candidates appeared
  - each video coded by up to five coders
  - videos divided into 5-second segments
  - segment is "treated" if a majority of coders identify the candidate



Real-time Evaluation of Campaign Advertisement

- separate survey conducted on Prolific ($N = 1,108$)
- randomly assign videos to respondents
- respondents continuously evaluated the candidate using a feeling thermometer (0–100)
- dial positions were recorded throughout the video
- each respondent rated on average 9.81 advertisements

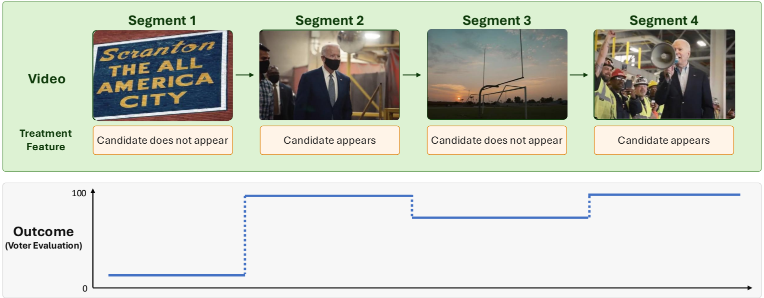


Methodological Challenges

1. High-dimensionality of videos
    - even a short video contains many audiovisual features
    - pre-trained embedding is less developed for video data and its quality is unknown
2. Confounding by other video features
    - candidates do not appear in ads randomly
    - e.g., candidates are more likely to appear in ads with positive emotions
    - random assignment of videos is not enough
3. Dynamic real-time outcome
    - viewers' current evaluation might depend on any of the previous frames
    - unknown carryover effects

Can we estimate the causal effects of video features on real-time responses?

Setup



- $\boldsymbol{X}_i$: videos assigned to individual i
- $\boldsymbol{X}_{is}$: s-th segment of video assigned to individual i
- S_i: length of video segments assigned to individual i
- Y_{is}: outcome of respondent i at segment s (e.g., voter evaluation)
- $Y_{is}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s)$: potential outcome for individual i at segment s
- W_{is}: Treatment feature of videos assigned to individual i at segment s

Assumptions

1. Consistency: $Y_{is} = Y_{is}(\boldsymbol{X}_{i1}, \cdots, \boldsymbol{X}_{is})$
2. Randomization: $Y_{is}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s) \perp\!\!\!\perp \boldsymbol{X}_i$
3. Treatment Feature: $W_{is} = g_W(\boldsymbol{X}_{is})$ (i.e., objective feature rather than perceived one)
4. Confounding Features: Confounding features in $\boldsymbol{X}_{is}$ for $Y_{is'}$ are unknown

$$\boldsymbol{U}_{is}^{(s')} = \boldsymbol{g}_{\boldsymbol{U}_s}^{(s')}(\boldsymbol{X}_{is})$$

5. Sequential Separability:

$$Y_{is}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s) = Y_{is}(g_W(\boldsymbol{x}_1), \boldsymbol{g}_{\boldsymbol{U}_1}^{(s)}(\boldsymbol{x}_1), \ldots, g_W(\boldsymbol{x}_s), \boldsymbol{g}_{\boldsymbol{U}_s}^{(s)}(\boldsymbol{x}_s)).$$

- we can intervene on the treatment feature while holding the confounding features constant
- implies independent support of treatment and confounding features
- also implies positivity given confounding features Details

- Deep Generative Models

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

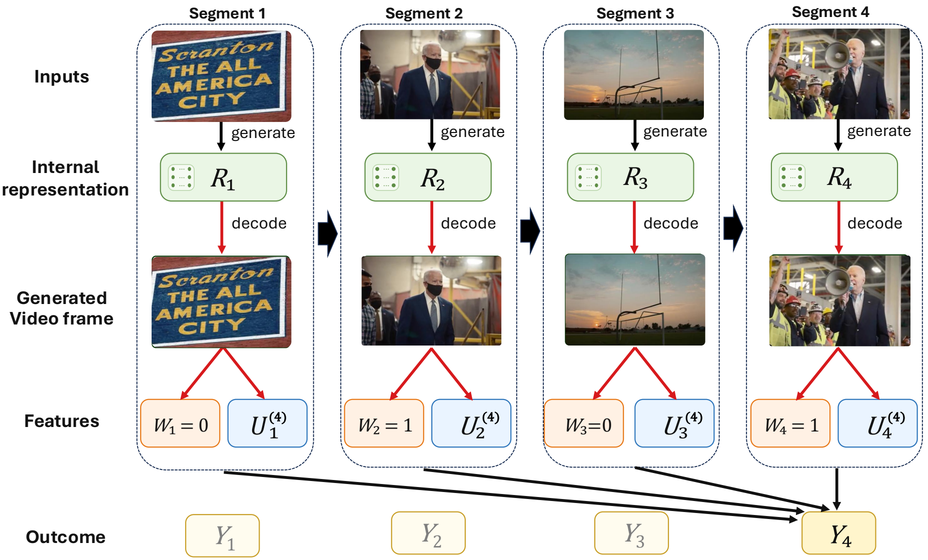
  - $\boldsymbol{P}_i$: the input of deep generative models
  - $\boldsymbol{R}_i$: the low-dimensional internal representations

- Factorized Deterministic Decoding: Each segment is separately generated so that

$$\mathbb{P}(\boldsymbol{X}_i \mid \boldsymbol{h}_\gamma(\boldsymbol{R}_i), S_i) = \prod_{s=1}^{S_i} \mathbb{P}(\boldsymbol{X}_{is} \mid \boldsymbol{h}_\gamma(\boldsymbol{R}_{is}), S_i)$$

and $\mathbb{P}(\boldsymbol{X}_{is} \mid \boldsymbol{R}_{is}, S_i)$ is degenerate.

Illustration of Setup



Stochastic Intervention

- High-dimensional data $\rightsquigarrow$ violation of overlap assumption
- Want our counterfactuals to be close to the observed data
- Stochastic intervention: for each segment s,

$$q_s(\delta_s; \overline{\boldsymbol{w}}_{s-1}) := \frac{\delta_s p_s(\overline{\boldsymbol{w}}_{s-1})}{\delta_s p_s(\overline{\boldsymbol{w}}_{s-1}) + 1 - p_s(\overline{\boldsymbol{w}}_{s-1})}, \quad \delta_s \in (0, \infty)$$

  where $p_s(\overline{\boldsymbol{w}}_{s-1}) = \mathbb{P}(W_{is} = 1 \mid \overline{\boldsymbol{W}}_{i,s-1} = \overline{\boldsymbol{w}}_{s-1}, S_i \geq s)$
- Greater value of δ_s $\rightarrow$ higher assignment probability Details
- q_s intervenes only when possible, given the treatment history
  - When $p_s(\overline{\boldsymbol{w}}_{s-1}) = 1$, then intervention q_s always assigns 1
  - When $p_s(\overline{\boldsymbol{w}}_{s-1}) = 0$, then intervention q_s always assigns 0

$\rightarrow$ We only require the positivity given confounding features

Causal Identification

- Average potential outcome under stochastic intervention:

$$\mathbb{E}\left[\int_{\mathcal{W}^s} Y_{is}(W_{i1} = w_1, \boldsymbol{U}_{i1}^{(s)}, \ldots, W_{is} = w_s, \boldsymbol{U}_{is}^{(s)}) \prod_{s'=1}^{s} dQ_{s'}(w_{s'}; \overline{\boldsymbol{w}}_{s'-1}, \delta_{s'})\right]$$

- We can identify the target parameter by adjusting a low-dimensional deconfounder

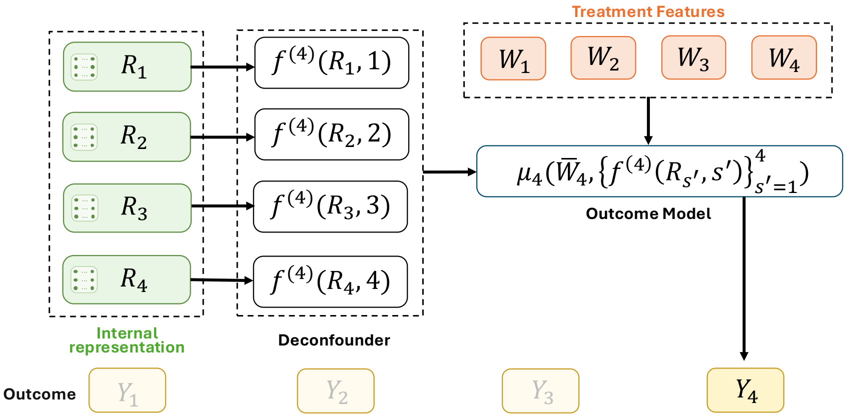
$$\begin{aligned}
&\mathbb{E}[Y_{is} \mid \overline{\boldsymbol{W}}_{is}, \boldsymbol{f}_1^{(s)}(\boldsymbol{R}_{i1}), \ldots, \boldsymbol{f}_s^{(s)}(\boldsymbol{R}_{is}), S_i \geq s] \\
&\quad = \mathbb{E}[Y_{is} \mid \overline{\boldsymbol{W}}_{is}, \boldsymbol{f}_1^{(s)}(\boldsymbol{R}_{i1}), \ldots, \boldsymbol{f}_s^{(s)}(\boldsymbol{R}_{is}), \overline{\boldsymbol{R}}_{is}, S_i \geq s]
\end{aligned}$$

    - a deconfounder always exists (e.g., confounding features)
    - identification works with any deconfounder

- Direct adjustment of $\boldsymbol{R}_{is}$ leads to overlap violation

Estimation

- Longitudinal neural network architecture with mean independence Details



- Use longitudinal influence function for estimation Procedure

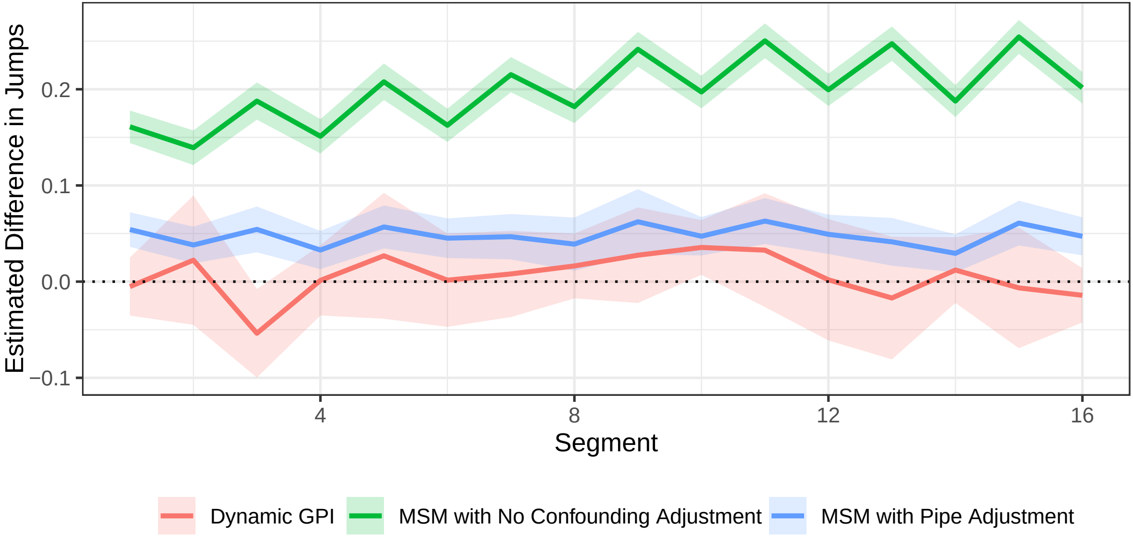
Super Mario Bros.™ Causal Benchmark Dataset

- Setup
  - Treatment: appearance of Princess Peach
  - Confounder: number of pipes
  - Outcome: number of jumps by Mario
- Confounding: Princess Peach appears more frequently when there are more pipes in each segment. Details
- Mario is played by a reinforcement learning agent
  - realistic outcome model (complex black-box policy)
  - Princess Peach does not affect Mario's jumps
  - True causal effect is zero
- $N = 10,000$ videos with equal length and autoscroll

Analysis Setup

- 1-second segments
- Regenerate each video segment by Cosmos Tokenizer
  - original video dimension: 460,800 = (320, 480, 3)
  - dimension of internal representations: 38,400 = (16, 40, 60)
- Estimate the trajectory of the average number of jumps
- contrast between $\delta = 5.0$ (more Peach appearance) and $\delta = 0.5$ (less Peach appearance)
- Three estimators
  1. Dynamic GPI (proposed)
  2. Marginal Structural Models (MSM) adjusting for the number of pipes
  3. Marginal Structural Models without confounding adjustment
- Same interventions and estimation strategies: only difference is confounding adjustment

Validation Results

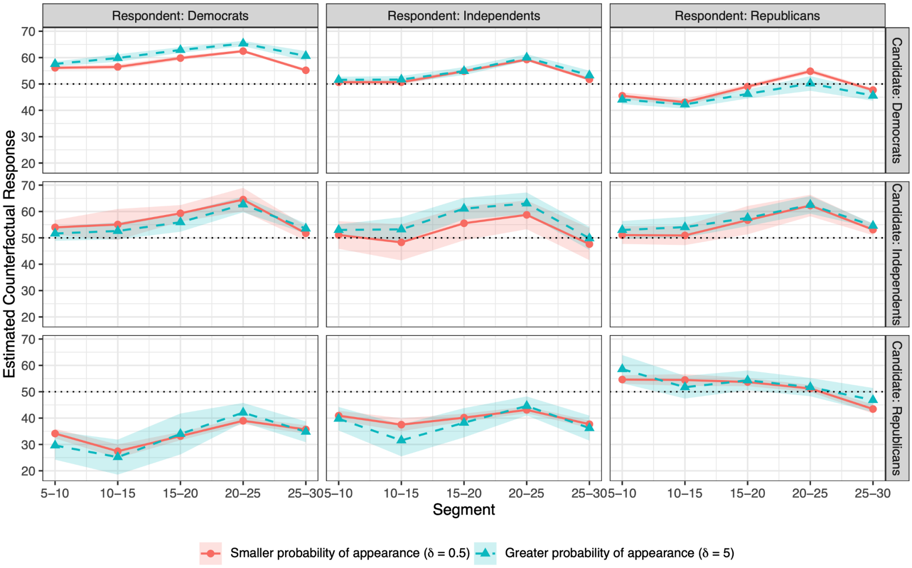


Empirical Application: Setup

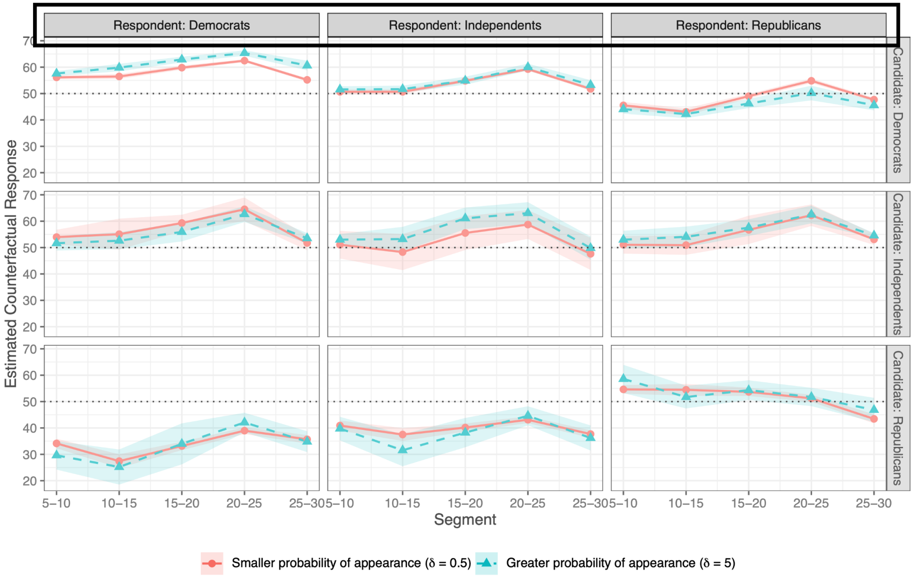
- 5-second segment, regenerated by Cosmos tokenizer as before
  - Original video frame (left) and reconstructed frame (right)

- For audio data,
  1. use Whisper for auto-transcription with time stamps
  2. use LLaMa3.1 (8B) to regenerate transcripts and obtain internal representations
- Dynamic GPI: $\delta = 0.5$ (less appearance) and $\delta = 5.0$ (more appearance)

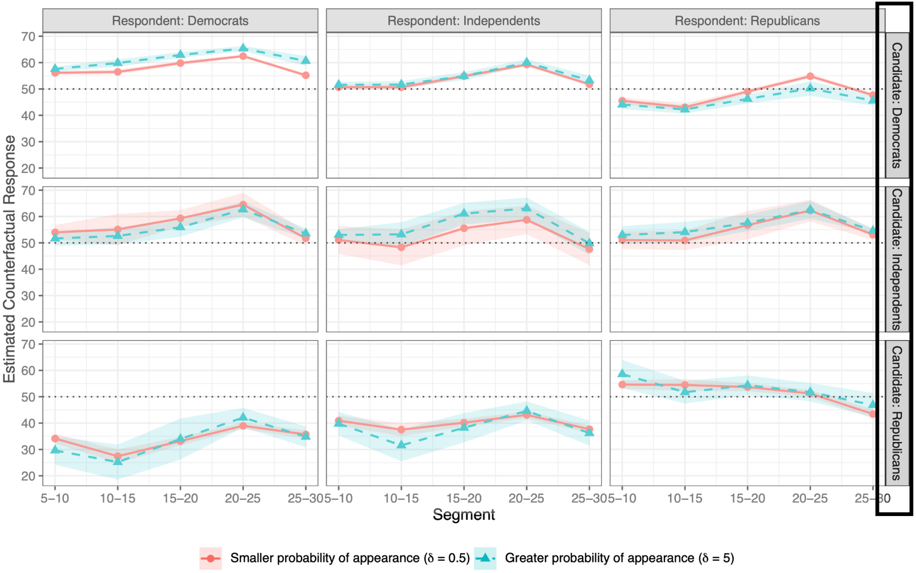
Empirical Application: Results



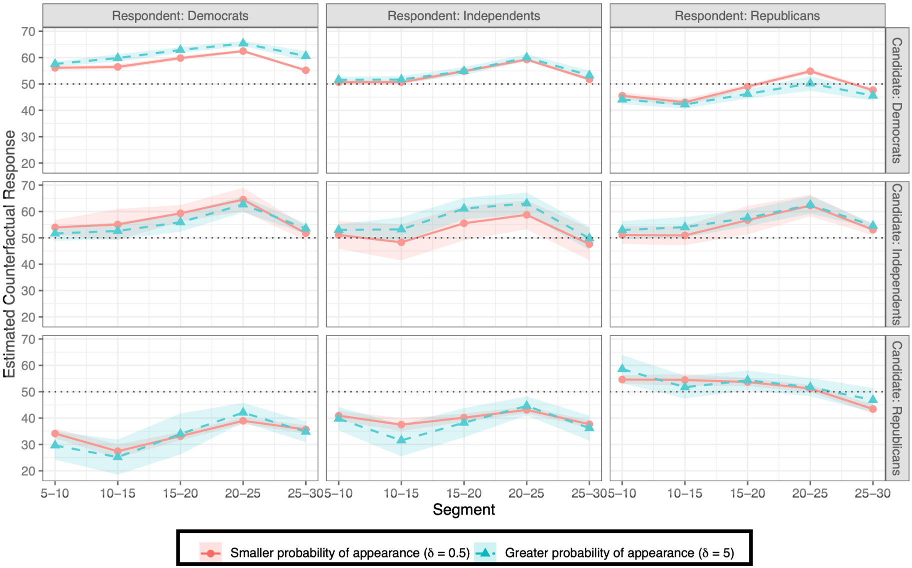
Empirical Findings



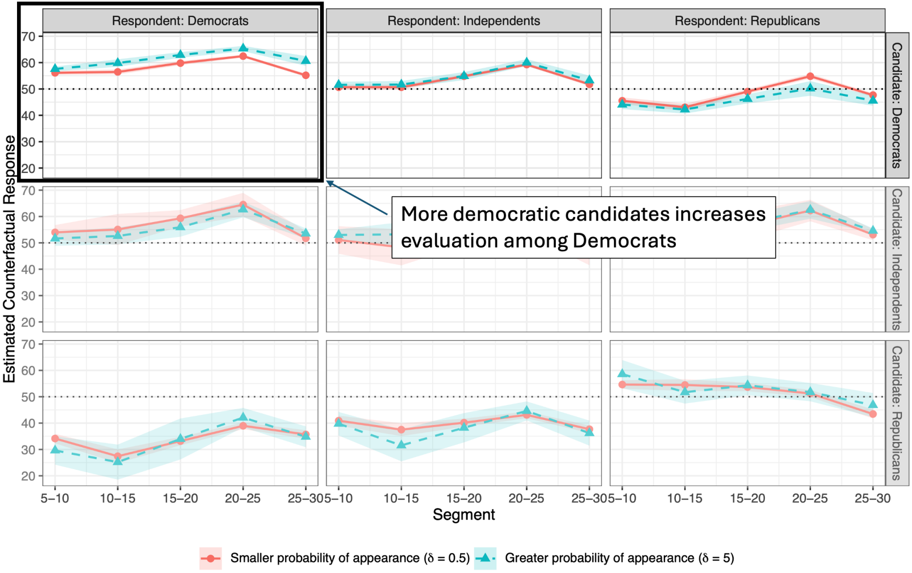
Empirical Findings



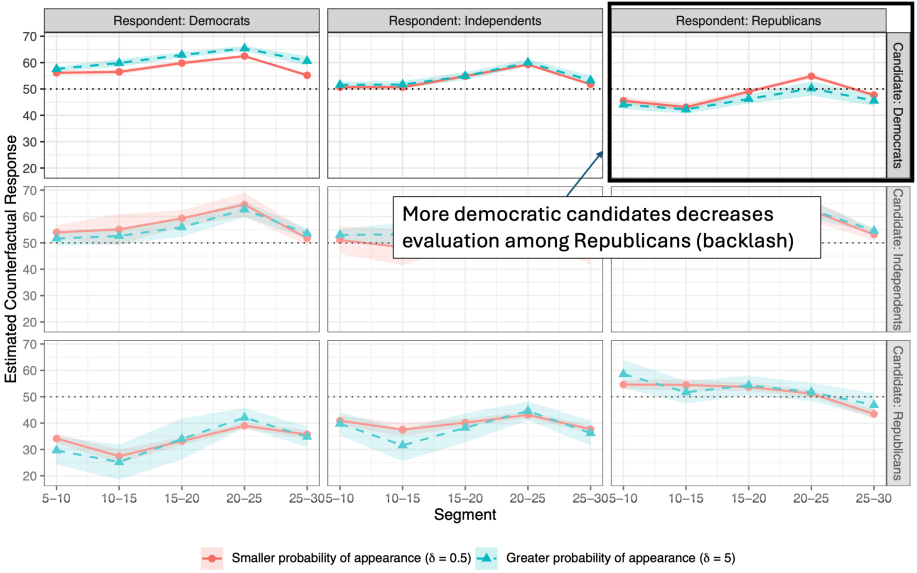
Empirical Findings



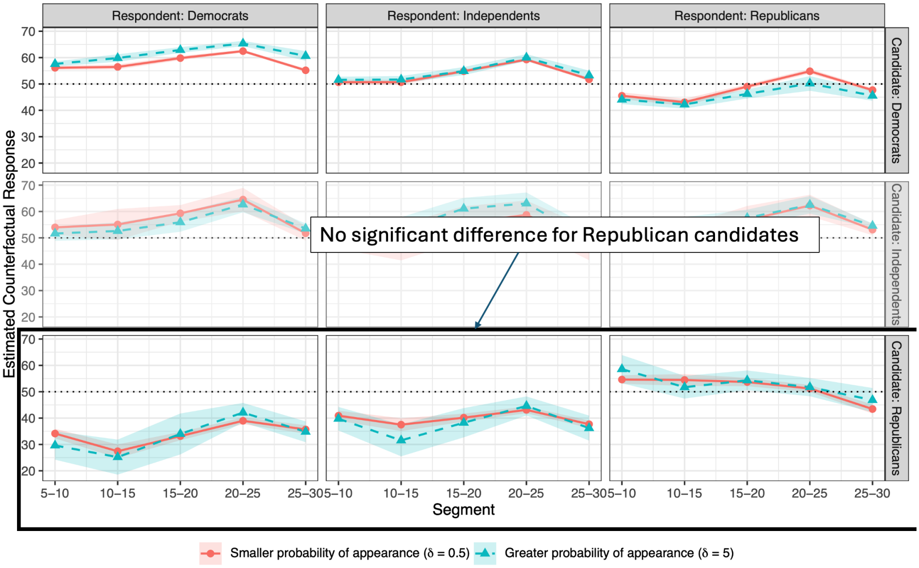
Empirical Findings



Empirical Findings



Empirical Findings



Conclusion

- Generative AI can be used to improve causal/statistical inference
  - GenAI can (re)generate high-quality videos
  - Its internal representations contain all the information in a low-dimensional format
  - We can use them to estimate the causal effects of specific video features in a dynamic setting
- Paper:

  `https://arxiv.org/abs/2607.06126`

- Open-source software GPI (GenAI Powered Inference):

  `https://gpi-pack.github.io/`

- Further extensions
  - better use of speech / audio information
  - fully continuous time (no segmentation)
  - policy learning with unstructured treatments

Sequantial Separabiltiy implies Independent Support

Proposition 1

Define the joint support induced by the sth video interval as

$$\mathcal{S}^{(s')}_{WU,s} := \left\{ \left(g_W(\boldsymbol{x}), \boldsymbol{g}^{(s')}_{\boldsymbol{U}_s}(\boldsymbol{x}) \right) : \boldsymbol{x} \in \mathcal{X}_s \right\}.$$

Under separability, we have

$$\mathcal{S}^{(s')}_{WU,s} = \mathcal{W}_s \times \mathcal{U}^{(s')}_s.$$

- Separability requires that we can intervene treatment feature while fixing confounding feature
- This requirement is violated if the support of confounding features depend on treatment feature
- Treatment levels must be attainable for every confounder values

$\rightarrow$ Independent support is pre-condition for positivity given confounding features

Interpretation of Stochastic Intervention Parameter

- Stochastic intervention is defined as

$$q_s(\delta_s; \overline{\boldsymbol{w}}_{s-1}) := \frac{\delta_s p_s(\overline{\boldsymbol{w}}_{s-1})}{\delta_s p_s(\overline{\boldsymbol{w}}_{s-1}) + 1 - p_s(\overline{\boldsymbol{w}}_{s-1})}.$$

- We can show that

$$\underbrace{\frac{q_s(\delta_s; \overline{\boldsymbol{w}}_{s-1})}{1 - q_s(\delta_s; \overline{\boldsymbol{w}}_{s-1})}}_{\text{Odds of Intervention}} = \delta_s \cdot \underbrace{\frac{p_s(\overline{\boldsymbol{w}}_{s-1})}{1 - p_s(\overline{\boldsymbol{w}}_{s-1})}}_{\text{Odds of Treatment History}}$$

- **Interpretation**
  - $\delta_s = 1$: Set treatment mechanism to observed treatment history
  - $\delta_s > 1$: More treatment than observed treatment history
  - $\delta_s < 1$: Less Treatment than observed treatment history

Bounded Overlap

- Bounded Overlap: There exists a constant $c > 0$ such that

$$\pi_s(\overline{\boldsymbol{w}}_{s-1}, \overline{\boldsymbol{u}}_s) \geq c \cdot p_s(\overline{\boldsymbol{w}}_{s-1})$$
$$1 - \pi_s(\overline{\boldsymbol{w}}_{s-1}, \overline{\boldsymbol{u}}_s) \geq c \cdot (1 - p_s(\overline{\boldsymbol{w}}_{s-1}))$$

where $\pi_s(\overline{\boldsymbol{W}}_{i,s-1}, \overline{\boldsymbol{U}}_{is}^{(s)}) := \mathbb{P}(W_{is} = 1 \mid \overline{\boldsymbol{W}}_{i,s-1}, \overline{\boldsymbol{U}}_{is}^{(s)})$

- Substantially weaker requirement than standard overlap in longitudinal setting
  - If $p_s(\overline{\boldsymbol{w}}_{s-1}) \in \{0, 1\}$, then the propensity score $\pi_s(\overline{\boldsymbol{W}}_{i,s-1}, \overline{\boldsymbol{U}}_{is}^{(s)})$ takes the same value
  - No intervention so no overlap is needed
- Similar idea to incremental propensity score, but use treatment history rather than propensity score π_t

Estimating Deconfounder

- Deconfounder is a low-dimensional representations that satisfy the mean-independence

$$\mathbb{E}[Y_{is} \mid \overline{\boldsymbol{W}}_{is}, \boldsymbol{f}_1^{(s)}(\boldsymbol{R}_{i1}), \ldots, \boldsymbol{f}_s^{(s)}(\boldsymbol{R}_{is}), S_i \geq s] \\ = \mathbb{E}[Y_{is} \mid \overline{\boldsymbol{W}}_{is}, \boldsymbol{f}_1^{(s)}(\boldsymbol{R}_{i1}), \ldots, \boldsymbol{f}_s^{(s)}(\boldsymbol{R}_{is}), \overline{\boldsymbol{R}}_{is}, S_i \geq s]$$

  - Deconfounder is not a function of treatment (not post-treatment)
  - Low-dimensional so that overlap is satisfied

- We estimate such deconfounder by minimizing the loss function

$$\min_{\boldsymbol{\theta}_s, \boldsymbol{\lambda}_s} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{S_i \geq s\} \left\{ Y_{is} - \mu_s(\overline{\boldsymbol{W}}_{is}, \{\boldsymbol{f}^{(s)}(\boldsymbol{R}_{is'}, s'; \boldsymbol{\lambda}_s)\}_{s'=1}^{s}; \boldsymbol{\theta}_s) \right\}^2$$

Estimation Procedure

1. Choose the incremental intervention parameter $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_{s_{\max}})$ and partition the data into K folds.
2. For each fold, estimate all nuisance functions using the training data:
   - deconfounder and outcome regression $(\hat{f}, \hat{\mu})$,
   - propensity score $\hat{\pi}$,
   - observed treatment probability $\hat{p}$.
3. For the held-out fold, compute the incremental propensity weights $\hat{\omega}$ and cumulative weights $\tilde{\omega}$.
4. Recursively compute the pseudo-outcome regressions $\hat{m}_{\ell|s}$ by backward induction $(\ell = s, \ldots, 1)$.
5. Solve the cross-fitted estimating equation

$$\frac{1}{N} \sum_{i=1}^{N} \bar{\psi}\Big(\mathcal{D}_i; \boldsymbol{\delta}, \hat{\boldsymbol{\eta}}^{(-I(i))}, \widehat{\bar{\Psi}}\Big) = 0,$$

and estimate the variance using the empirical variance of the estimated influence function.

Inference on Average Potential Outcome Trajectory

- Estimand: Average Potential Outcome Trajectory

$$\vec{\boldsymbol{\Psi}}(\boldsymbol{\delta}) = (\Psi_1(\boldsymbol{\delta}), \cdots, \Psi_{s_{\max}}(\boldsymbol{\delta}))$$

- Inference requires the uniform confidence bands

$$\mathbb{P}\left(\Psi_s(\boldsymbol{\delta}) \in \left[\widehat{\Psi}_s(\boldsymbol{\delta}) \pm \widehat{c}_{1-\alpha}(\boldsymbol{\delta})\frac{\widehat{\sigma}_s(\boldsymbol{\delta})}{\sqrt{N_s}}\right] \text{ for all } s\right) \to 1-\alpha.$$

- Multiplier bootstrap for uniform confidence bounds
  1. Estimate the influence function
  2. For each $b = 1, \ldots, B$, draw a set of independent Rademacher multipliers and form the bootstrapped process

$$\widehat{G}_s^{(b)}(\boldsymbol{\delta}) = \frac{1}{\sqrt{N_s}} \sum_{k=1}^{K} \sum_{I(i)=k} \xi_i^{(b)} \frac{\mathbb{1}\{S_i \geq s\}\, \psi_s\left(\mathcal{D}_{is}; \boldsymbol{\delta}, \hat{\boldsymbol{\eta}}^{(-k)}, \widehat{\Psi}_s\right)}{\widehat{\sigma}_s(\boldsymbol{\delta})}$$

  3. Compute the maximum of the absolute bootstrapped process,

$$T^{(b)}(\boldsymbol{\delta}) = \max_{1 \leq s \leq s_{\max}} \left|\widehat{G}_s^{(b)}(\boldsymbol{\delta})\right|.$$

Mario Benchmark Data

- We partition each video into 2-second segments
- Pipe generation:
  - We first randomly assign 0 or 1 pipe
  - The number of pipe in each subsequent segments follows random-walk
    - Decrease by one w/ probability 0.4
    - Remains unchanged w/ probability 0.4
    - Increase by one w/ probability 0.2
- Peach generation: Generate peach by

$$W_{is} \sim \mathrm{Bernoulli}\left(0.05 + 0.9 \cdot \sigma(-1.5 + U_{is})\right),$$

which makes peach more likely as pipe increases.

  - $\Pr(W_{is} = 1 \mid U_{is} = 0) = 0.214$
  - $\Pr(W_{is} = 1 \mid U_{is} = 1) = 0.390$
  - $\Pr(W_{is} = 1 \mid U_{is} = 2) = 0.610$
  - $\Pr(W_{is} = 1 \mid U_{is} = 3) = 0.786.$