Leveraging LLM-Generated Structural Prior for Causal Inference with Concurrent Causes

Xingjian Zhang **Shixuan Liu** **Yixin Wang** **Qiaozhu Mei**
University of Michigan
Ann Arbor, MI
{jimmyzxj, shixuanl, yixinw, qmei}@umich.edu

Abstract

Causal inference with many potential concurrent causes presents significant challenges across various fields, from biomedicine to policy analysis. The core challenge lies in understanding how combinations of potential causes influence an outcome, which becomes exponentially more complex as the number of potential concurrent causes increases. To address this challenge, we propose to incorporate structural prior information that describes the interrelations between causes. Specifically, we use a large language model (LLM) to systematically curate this structural information, effectively reducing the complexity of the causal inference task. We validate our method using both a semi-synthetic dataset and a real-world case study from the film industry.[1]

1 Introduction

Causal inference from observational data is a critical but challenging task. Traditional settings often focus on binary or continuous treatments [30, 10] while a lot of real-world situations involve complex treatments made up of many concurrent causes. This complexity arises in fields such as medicine, where the combined effects of multiple drugs are evaluated [41], or political science, where multiple policies are studied [6]. In the film industry, producers must understand how different combinations of actors (i.e. a cast) impact the box office performance, where each actor is a *concurrent cause* and a specific cast constitutes a *treatment*, with return on investment (ROI) as the *outcome*. **This illustrative example will be used throughout the paper to demonstrate our methodology**.

Many challenges arise in these novel settings. As the number of potential concurrent causes m increases, the possible combinations grow exponentially, making it infeasible to consider each combination independently. To mitigate this, a common approach is to assume a multilinear relationship between concurrent causes and the outcome [38, 31, 25]. However, this approach is complicated by imbalanced data, where certain causes are observed more frequently, resulting in a higher variance for less common causes. Furthermore, the fundamental problem of causal inference—the inability to observe counterfactual outcomes—becomes more intractable as the number of causes grows, further complicating the estimation of causal effects.

To address these challenges, a potential way is to incorporate prior knowledge to guide the estimation process. Large language models (LLM), which have shown exceptional reasoning abilities in recent years, present a promising solution for systematically curating this prior knowledge by extracting and synthesizing information from vast text corpora, such as Wikipedia. Inspired by recent work on the decomposition of complex tasks into simpler pairwise comparisons [44, 28], we propose a method that takes advantage of LLMs to construct a similarity graph representing the interrelationships

[1] Our experiment code is openly available at
`https://anonymous.4open.science/r/causal-inference-with-graph-prior-59B5`.

NeurIPS 2024 Workshop on Causality and Large Models (CaLM).

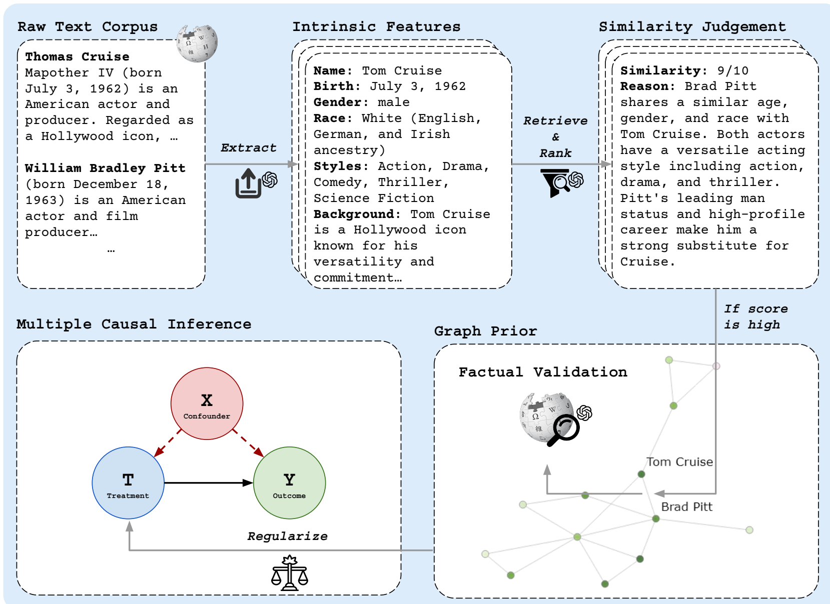

Figure 1: Overview of the proposed methodology for analyzing film ROI (outcome), where actors are considered as concurrent causes. LLMs are used to extract structured actor information from raw text sources (e.g., Wikipedia). Next, we prompt LLMs to retrieve and rank actor pairs based on similarity, constructing a similarity graph where nodes represent actors and edges indicate the belief that one actor can be replaced by another without impacting the film's ROI. We assess the reliability of these similarity judgments using `FActScore` [24]. Finally, the similarity graph serves as prior information to regularize causal inference.

between potential concurrent causes. This graph serves as a structural prior that regularizes the estimation process in causal inference. Our methodology is illustrated in Figure 1.

2 Method

2.1 Causal inference with many potential concurrent causes

Traditional approaches of causal inference from observational data aim to estimate the distribution of potential outcomes Y given a binary treatment T, where T can take values $t \in \{0, 1\}$, in the presence of n confounders $\boldsymbol{x} \in \mathbb{R}^n$. The potential outcome function $Y_i(t)$ represents the outcome of subject i under treatment t. When multiple concurrent causes are considered, the treatment t is extended to a binary vector $\boldsymbol{v} \in \{0, 1\}^m$, where m represents the number of potential concurrent causes[2]. Consequently, the dataset consists of tuples $\mathcal{D} = \{(\boldsymbol{v}_i, \boldsymbol{x}_i, y_i)\}_{i=1}^N$, where N denotes the number of observations.

Our goal is to estimate the average treatment effect (ATE) $\mu(\boldsymbol{v}) = \mathbb{E}[Y_i(\boldsymbol{v})]$ for any treatment $\boldsymbol{v}$. However, a naive Monte Carlo estimate of $\mu(\boldsymbol{v})$ are biased due to the *fundamental problem of causal inference*, expressed as

$$\mathbb{E}[Y_i(\boldsymbol{v})|T = \boldsymbol{v}] \neq \mathbb{E}[Y_i(\boldsymbol{v})]. \tag{1}$$

Given the assumptions of no unmeasured confounders, the Stable Unit Treatment Value Assumption (SUTVA), and overlap (a.k.a. positivity) [11], the ATE $\mu(\boldsymbol{v})$ can be identified using a plug-in estimator:

$$\tau(\boldsymbol{v}) = \mathbb{E}[\mathbb{E}[Y_i(\boldsymbol{v})|X, V = \boldsymbol{v}]] = \mu(\boldsymbol{v}). \tag{2}$$

[2]While our method is presented using binary causes and a continuous outcome, it can be generalized to accommodate continuous causes and discrete outcomes.

In practice, we approximate the inner expectation with a parametric function f_θ, often assumed to be linear [38, 31, 25], and estimate $\tau(\boldsymbol{v})$ via Monte Carlo sampling:

$$\hat{\tau}(\boldsymbol{v}) = \frac{1}{k}\sum_{j=1:\boldsymbol{x}_j\sim\mathcal{X}}^{k} f_\theta(\boldsymbol{x}_j, \boldsymbol{v}), \text{ where } f_\theta(\boldsymbol{x}, \boldsymbol{v}) = \mathbb{E}[Y_i(\boldsymbol{v})|X, V=\boldsymbol{v}] = \theta_x^\top \boldsymbol{x} + \theta_v^\top \boldsymbol{v} \tag{3}$$

Here k is the sample size, and $\mathcal{X}$ represents the empirical distribution of covariates $\boldsymbol{x}$ in $\mathcal{D}$. To learn f_θ, a straightforward method is to use ordinary least square (OLS) linear regression, which serves as our baseline.

2.2 Structural Prior Knowledge

Definition In practice, we often have access to prior information regarding the similarity between potential concurrent causes. This information is based on the belief that *substituting one cause for another is unlikely to significantly affect the potential outcome*. For instance, as illustrated in Figure 1, replacing Tom Cruise with Brad Pitt may not drastically alter a film's box office. This pairwise similarity relationship can be effectively represented by an undirected graph $\mathcal{G} = (V, A)$ predetermined by domain expertise. Here, the node set $V = \{v_1, \ldots, v_m\}$ comprises all potential causes, and the edges are encoded in an adjacency matrix A. We use an unweighted graph in the demonstration.

Graph-Based Regularization We use Laplacian regularization, a graph-based regularization technique that incorporates graph structure information into training, to regularize the linear regression in Eq 3. Laplacian regularization has wide applications in multiple machine learning fields, such as semi-supervised learning and graph learning [47, 46, 2, 40]. It introduces an explicit regularization term into the objective function, leveraging the graph Laplacian, $L = D - A$, where D is the diagonal degree matrix, to promote parameter similarity among strongly connected nodes. Therefore, the new objective is

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(f_\theta(\boldsymbol{x}_i, \boldsymbol{v}_i) - y_i)^2 + \lambda\theta_v^T L \theta_v \tag{4}$$

Here, λ controls the strength of regularization. Intuitively, the Laplacian regularization penalizes the parameter discrepancies between connected nodes, thereby enforcing smoothness in the parameter space in accordance with the graph structure.

2.3 Prior Knowledge Generation with LLMs

Relying on domain experts to curate prior knowledge is often inefficient due to the significant time and financial costs involved. Inspired by recent advancements that leverage LLMs for atomic tasks such as pairwise comparison [44, 28], we propose LLMs to automate the curation of this prior knowledge. We illustrate our method through the aforementioned film example in Figure 1.

Information Retrieval and Intrinsic Feature Extraction To minimize hallucination, we integrate a simplified version of retrieval-augmented generation (RAG) for prior knowledge generation. We retrieve and process a collection of documents for each actor using the Wikipedia API. During the generation phase, these preprocessed documents are provided to LLMs accordingly. This approach enhances reproducibility and stability of the experiment and can be generalized or extended to other applications where domain-specific background information is accessible. However, the raw data retrieved from Wikipedia is not guaranteed to be free from confounding effects. For instance, actor biographies may include information about their collaborations with other actors, which could introduce bias. To address this issue, we employ LLMs to clean the prior information by filtering out extraneous details and preserving only those intrinsic characteristics. Only this refined information is used in subsequent steps.

Graph Construction and Factual Validation A straightforward graph construction process would involve pairwise evaluations of all actors. To reduce the number of API calls, we first prompt LLMs to retrieve the $k_{\text{retrieval}}$ most similar candidates for each actor without providing biographical information. Subsequently, biographical information is added, and LLMs are prompted again to narrow down

Model	MSE (SE.)	
	No Control	Control
Linear Regression	0.349 (0.001)	0.115 (0.003)
Laplacian Reg Linear Regression	0.231 (0.001)	**0.018 (0.002)**

Table 1: Performance comparison between the proposed method and the baseline under an ideal synthetic scenario (rich observations and accurate prior). Standard errors (in parentheses) are computed across 5 random seeds. The "Control" column indicates results when confounder control is applied using a plug-in estimator, while "No Control" represents the setting without this adjustment.

and re-rank these candidates to obtain at most k_{rank} final choices. A detailed example of this process can be found in Appendix D. To ensure the faithfulness of this process, we solicit both a similarity score and an explanation for each proposed similarity from the LLM. We then use `FActScore` [24] to evaluate the extent to which the rationale is supported by evidence from the curated documents, ensuring that the generated knowledge is accurate and reliable.

3 Semi-Synthetic Experiment Validation

In this section, we evaluate the effectiveness of the proposed regularization technique using a semi-synthetic dataset. Our goal is to estimate the underlying parameters, θ_v, assuming a linear data generation process. Since obtaining ground truth in real-world scenarios is challenging, we create a semi-synthetic dataset based on the `TMDB5000` dataset[3] for quantitative validation.

Settings The original `TMDB5000` dataset includes data of 901 actors, each having appeared in at least 9 movies, along with revenue information for 2,828 movies. The movies in this dataset span 18 genres and are delivered in 58 languages. In our semi-synthetic dataset, we utilize the genre information as confounders, thus representing each film i with covariates $\boldsymbol{x}_i \in \{0,1\}^{18}$. We then generate the treatments $\boldsymbol{v}_i$ based on the confounders $\boldsymbol{x}_i$, and create the true parameters θ from a uniform distribution. The outcomes y_i are generated using a linear relationship as defined in Eq.3. The graph prior is constructed according to pairwise similarity within $\theta_v + \epsilon$, where ϵ is noise that controls the accuracy of the prior. Further details on the data generation process can be found in Appendix C. We report the mean squared error (MSE) between the learned parameters $\widehat{\theta_v}$ and the true parameters θ_v.

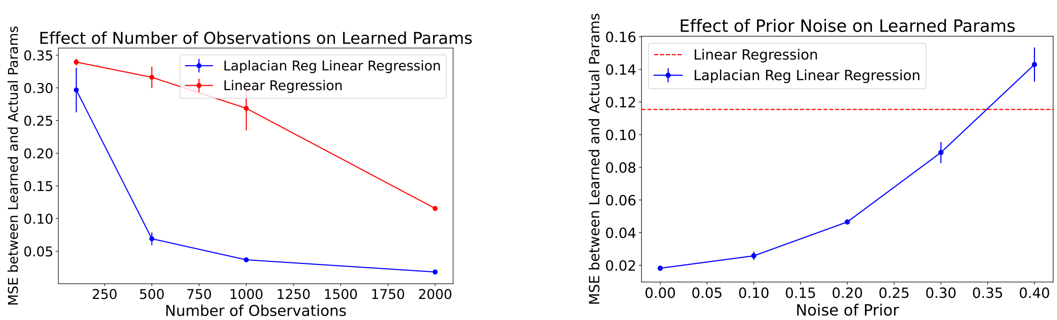

Figure 2: The effects of dataset properties on performance of the models. Error bars indicate the standard error derived by 5 different random seeds. **Left**: The performance of the models vs. the number of observation. **Right**: The performance of graph regularization method vs. the level of noise. The x-axis indicates the standard error of noise σ_θ.

Results We evaluate our methods under three scenarios. (a) In the ideal scenario, the number of observations $N = 2,828$ is large, and the prior information (represented by $\mathcal{G}$) is accurate. As shown in Table 1, the proposed method significantly outperforms the baseline, yielding more accurate

[3]https://www.kaggle.com/tmdb

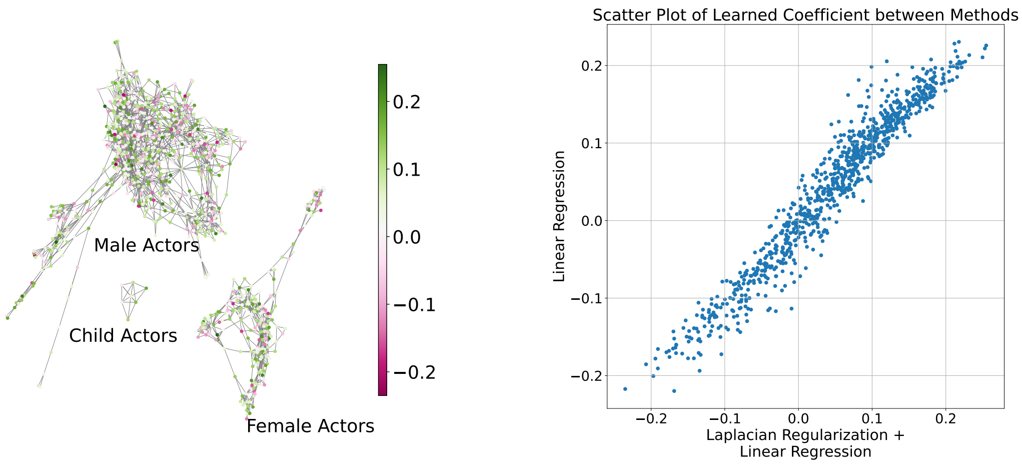

Figure 3: **Left**: Graph prior generated by LLM, showing connections among 901 actors. The presence of an edge indicates the two connected actors are similar. Node colors indicate learned parameters (green: positive; red: negative). **Right**: Coefficients learned by two methods. Despite a very small regularization, certain data points are affected significantly.

estimates of θ_v regardless of whether confounder control is applied. (b) Next, we vary the number of observations to evaluate the performance of both methods under sparse data conditions. Figure 2 (Left) shows that our method consistently outperforms the baseline. The graph prior proves especially useful when the number of observations is moderate, as nodes with limited data benefit from similar neighboring nodes. However, when the number of observations is reduced to as few as 100, the performance gap between the two methods narrows due to the limited data available for accurate prediction. Conversely, when the number of observations is abundant, the performance of graph regularization saturates, reducing the disparity between the methods. (c) Finally, we investigate the robustness of the graph regularization method under varying levels of noise in the graph prior. Specifically, we examine scenarios where the noise standard deviation σ_ϵ is adjusted. Given that the standard deviation of θ_v is $\sigma_\theta = 1/\sqrt{3} \approx 0.58$, the injected noise is substantial compared to the scale of the ground truth parameters. As seen in Figure 2 (Right), our approach continues to outperform basic linear regression until σ_ϵ reaches 0.4, demonstrating its resilience in noisy environments.

4 Real-World Case Study

In this section, we demonstrate the effectiveness of the proposed pipeline through a real-world case study using the original `TMDB5000` dataset.

4.1 Prior Knowledge Generation with LLMs

Graph Construction and Validation We follow the pipeline described in Sec 2.3 to generate the graph prior. For each actor, intrinsic features such as *gender*, *birth*, *race*, *acting styles*, and *background* information are extracted from Wikipedia. We set $k_{\text{retrieval}} = 10$ and $k_{\text{rank}} = 5$ during graph construction, utilizing the `gpt-4o-2024-05-13` model for all experiments[4].

Graph Validation We use FActScore [24] as a measurement of the trustworthiness of the LLM-generated graph prior. It decomposes the LLM generation into atomic facts and checks the average groundedness of each atomic fact against a trusted corpus. In our case, we evaluates the FActScore of the reasoning provided along with the similarity judgement. Using the RAG, our final average score for the generation is 89.5%, indicating most of the statements are supported by facts[5].

[4]The prompt and sample response from the LLM are provided in Appendix D.
[5]The implementation details of `FActScore` are shown in Appendix E.

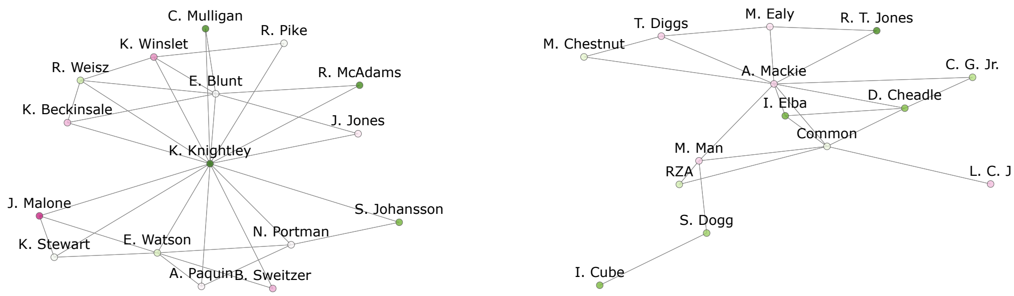

Figure 4: Ego networks of actors with high discrepancy in coefficients. Node colors indicate learned parameters (same as Figure 3). **Left**: 1-hop ego graph centered on Keira Knightley. **Right**: 2-hop ego graph centered on Method Man.

Results The extracted graph prior is visualized in Figure 3 (Left), comprising 2078 edges with an average node degree of 2.31. Notably, the graph features three major components corresponding to male, female, and child actors, suggesting that it effectively captures essential information on actor similarity. A few isolated nodes are also present, such as Aasif Mandvi, who is the only actor in the dataset with an Indian background, demonstrating his irreplaceability.

4.2 Comparison Study

Settings Utilizing the generated graph prior, we apply Laplacian regularization to linear regression and compare it with standard linear regression without regularization. A small regularization weight of $\lambda = 0.001$ is chosen to ensure that the scale of the learned coefficients remains largely unaffected.

Results Figure 3 (Right) shows the disparity in the learned coefficients between the two methods, with a Pearson correlation of 0.97. Despite the minimal regularization applied, there is a noticeable difference in the coefficients attributed to specific actors. We present case studies of two actors with notable coefficient discrepancies between standard and Laplacian-regularized linear regressions. Figure 4 (Left) shows Keira Knightley's ego graph, where her positive coefficient in the standard regression is significantly reduced under the graph-regularized model by 0.099. This reduction reflects the lower coefficients of her neighbors, with regularization penalizing her value accordingly. In contrast, Method Man's coefficient increases with regularization by 0.071, influenced by positive coefficients of nearby actors like RZA and Richard T. Jones, shown in Figure 4 (Right). Notably, Method Man is recognized more for his contributions to hip hop than acting, leading to fewer collaborations with prominent directors. The directors of his notable films, including Mike Devine and Jonathan Levine, are not widely recognized. This highlights a possible neglect of important confounding variables, such as a movie's director, underscoring the necessity for additional control measures in the analysis.

5 Discussion

Conclusion We propose a graph-based method for causal inference with multiple concurrent causes. By using LLMs to extract prior knowledge and construct the graph, we enhance the model's ability to capture complex relationships that traditional methods overlook. This approach demonstrates the value of LLMs in enriching causal inference with structured, context-aware insights.

Limitations and Future Work One limitation of the current approach lies in the similarity judgment by LLM, which requires inputting the full list of actors into the LLM's context window. For larger datasets, this can exceed the LLM's context limit, though this issue could potentially be mitigated by splitting the list across multiple windows. Additionally, the proposed method assumes a linear data generation process, which may not fully capture the complexities of real-world scenarios. A promising direction for future work is to extend the approach by using the graph prior for data augmentation, enabling the model to better handle non-linear relationships and more intricate causal structures.

References

[1] Weihua An. Causal inference with networked treatment diffusion. *Sociological Methodology*, 48(1):152–181, 2018.

[2] Rie Ando and Tong Zhang. Learning on graph with laplacian regularization. *Advances in neural information processing systems*, 19, 2006.

[3] Raquel Aoki, Yizhou Chen, and Martin Ester. Multi-treatment effect estimation from biomedical data. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2023: Kohala Coast, Hawaii, USA, 3–7 January 2023*, pages 299–310. World Scientific, 2022.

[4] Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, and Huanhuan Chen. Causal structure learning supervised by large language model. *arXiv preprint arXiv:2311.11689*, 2023.

[5] Taiyu Ban, Lyvzhou Chen, Xiangyu Wang, and Huanhuan Chen. From query tools to causal architects: Harnessing large language models for advanced causal discovery from data, 2023.

[6] Matthew Blackwell. A framework for dynamic causal inference in political science. *American Journal of Political Science*, 57(2):504–520, 2013.

[7] Dean Eckles, Brian Karrer, and Johan Ugander. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1):20150021, 2017.

[8] Ruocheng Guo, Jundong Li, and Huan Liu. Learning individual causal effects from networked observational data. In *Proceedings of the 13th international conference on web search and data mining*, pages 232–240, 2020.

[9] Shonosuke Harada and Hisashi Kashima. Graphite: Estimating individual effects of graph-structured treatments. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 659–668, 2021.

[10] Kosuke Imai and David A Van Dyk. Causal inference with general treatment regimes: Generalizing the propensity score. *Journal of the American Statistical Association*, 99(467):854–866, 2004.

[11] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

[12] Song Jiang and Yizhou Sun. Estimating causal effects on networked observational data via representation learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 852–861, 2022.

[13] Edward K Kao. *Causal inference under network interference: A framework for experiments on social networks*. PhD thesis, Harvard University, 2017.

[14] Vamsi Krishna Kommineni, Birgitta König-Ries, and Sheeba Samuel. From human experts to machines: An llm supported approach to ontology and knowledge graph construction, 2024.

[15] Dehan Kong, Shu Yang, and Linbo Wang. Identifiability of causal effects with multiple causes and a binary outcome. *Biometrika*, 109(1):265–272, 2022.

[16] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality, 2023.

[17] Fan Li and Fan Li. Propensity score weighting for causal inference with multiple treatments. 2019.

[18] Wenrui Li, Daniel L Sussman, and Eric D Kolaczyk. Causal inference under network interference with noise. *arXiv preprint arXiv:2105.04518*, 2021.

[19] Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*, 2024.

[20] Stephanie Long, Tibor Schuster, and Alexandre Piché. Can large language models build causal graphs?, 2024.

[21] Michael J Lopez and Roee Gutman. Estimation of causal effects with multiple treatments: a review and new ideas. *Statistical Science*, pages 432–454, 2017.

[22] Daniel F McCaffrey, Beth Ann Griffin, Daniel Almirall, Mary Ellen Slaughter, Rajeev Ramchand, and Lane F Burgette. A tutorial on propensity score estimation for multiple treatments using generalized boosted models. *Statistics in medicine*, 32(19):3388–3414, 2013.

[23] Lars-Peter Meyer, Claus Stadler, Johannes Frey, Norman Radtke, Kurt Junghanns, Roy Meissner, Gordian Dziwis, Kirill Bulert, and Michael Martin. Llm-assisted knowledge graph engineering: Experiments with chatgpt. In *Working conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow*, pages 103–115. Springer Fachmedien Wiesbaden Wiesbaden, 2023.

[24] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation, 2023.

[25] Preetam Nandy, Marloes H. Maathuis, and Thomas S. Richardson. Estimating the effect of joint interventions from observational data in sparse high-dimensional settings, 2016.

[26] Elizabeth L Ogburn, Oleg Sofrygin, Ivan Diaz, and Mark J Van der Laan. Causal inference for social network data. *Journal of the American Statistical Association*, 119(545):597–611, 2024.

[27] Zhaozhi Qian, Alicia Curth, and Mihaela van der Schaar. Estimating multi-cause treatment effects via single-cause perturbation. *Advances in Neural Information Processing Systems*, 34:23754–23767, 2021.

[28] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*, 2023.

[29] Rajesh Ranganath and Adler Perotte. Multiple causal inference with latent confounding. *arXiv preprint arXiv:1805.08273*, 2018.

[30] Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American statistical association*, 75(371):591–593, 1980.

[31] Arman Alam Siddique, Mireille E Schnitzer, Asma Bahamyirou, Guanbo Wang, Timothy H Holtz, Giovanni B Migliori, Giovanni Sotgiu, Neel R Gandhi, Mario H Vargas, Dick Menzies, et al. Causal inference with multiple concurrent medications: A comparison of methods and an application in multidrug-resistant tuberculosis. *Statistical methods in medical research*, 28(12):3534–3549, 2019.

[32] Dustin Tran and David M Blei. Implicit causal models for genome-wide association studies. *arXiv preprint arXiv:1710.10742*, 2017.

[33] R Tu, C Ma, and C Zhang. Causal-discovery performance of chatgpt in the context of neuropathic pain diagnosis, jan 2023.

[34] Tyler J VanderWeele and Weihua An. Social networks and causal inference. *Handbook of causal analysis for social research*, pages 353–374, 2013.

[35] Tyler J VanderWeele and Miguel A Hernan. Causal inference under multiple versions of treatment. *Journal of causal inference*, 1(1):1–20, 2013.

[36] Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. Causal inference using llm-guided discovery. *arXiv preprint arXiv:2310.15117*, 2023.

[37] Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. Causal inference using llm-guided discovery, 2023.

[38] Yixin Wang and David M. Blei. The blessings of multiple causes, 2019.

[39] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581*, 2018.

[40] Han Yang, Kaili Ma, and James Cheng. Rethinking graph regularization for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4573–4581, 2021.

[41] Afsaneh Yazdani and Eric Boerwinkle. Causal inference in the age of decision medicine. *Journal of data mining in genomics & proteomics*, 6(1), 2015.

[42] Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: theory and practice. *International Journal of Approximate Reasoning*, 151:101–129, 2022.

[43] Elaine Zanutto, Bo Lu, and Robert Hornik. Using propensity score subclassification for multiple treatment doses to evaluate a national antidrug media campaign. *Journal of Educational and Behavioral Statistics*, 30(1):59–73, 2005.

[44] Yuwei Zhang, Zihan Wang, and Jingbo Shang. Clusterllm: Large language models as a guide for text clustering. *arXiv preprint arXiv:2305.14871*, 2023.

[45] Shandong Zhao, David A van Dyk, and Kosuke Imai. Propensity score-based methods for causal inference in observational studies with non-binary treatments. *Statistical methods in medical research*, 29(3):709–727, 2020.

[46] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

[47] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.

A Visualization of Graph Prior

We use `graph-tool`'s Hierarchical community detection algorithm to plot the visualization of the extracted graph prior, shown in Figure 5.

B Related Work

Multiple Causal Inference Multiple causal inference is a topic of interest, spurred by a plethora of applications that have garnered attention in the machine learning community, inclusive of genome-wide association studies (GWAS), recommender systems, and polypharmacy [39, 32, 29, 31]. Numerous studies have been conducted in the realm of causal inference with multiple versions of treatments, where a single chosen treatment is administered to a subject per observation [3, 43, 10, 45, 22, 21, 29, 35, 17]. Alternatively, some studies focus on combinatorial treatments, where multiple treatments can be administered concurrently to a subject [15, 38, 27, 31]. The latter type of study is generally more complex due to the expanded treatment space and is therefore more challenging to approach. Our work also falls into this category. [31] explores the extension of G-computation, inverse propensity score estimation, and the double robust estimator with respect to four concurrent treatments. [39] proposes a novel algorithm "deconfounder" to tackle the problem of unobserved confounders in multiple causal inference scenarios. [27] seeks to address severe data scarcity by utilizing data augmentation techniques.

Causal Inference Incorporating Graph Structure A substantial body of work on causal inference involves the integration of graph structures or graph data. Some studies consider subject networks and corresponding network effects that potentially violate the basic assumptions of causal inference from observational data [12, 1, 7, 18, 13, 26, 34, 8]. For instance, [12] elucidates the limitations of standard graph machine learning models in estimating causal effects on networked observational data. Other research efforts have focused on estimating the causal effects of graph-structured treatments. As an example, [9] takes into account the graph structure of drug chemicals and employs a Graph Neural Network (GNN) to learn the representation of graph treatments. However, none of the existing work explores the setting where multiple treatments are modeled through a graph structure to estimate causal effects.

LLMs and Graph Creation As large language models emerge these years, they are applied to various downstream tasks, including graph construction in different contexts. One such kind of graphs are the knowledge graphs [14, 23], where LLMs serve as domain experts to develop ontology and build graphs that represent real-world knowledge. In causal inference, previous studies also leverage LLMs to explore causal relationships for directed acyclic graphs (DAG) engineering [19]. For instance, [16] opens the frontier by utilizing LLMs to determine pair-wise causal relationships. Despite the the high accuracy of this approach, it actually has some drawbacks, such as cyclic graph structure [36], the $O(N^2)$ compleity [4] and false information [20, 33]. To mitigate these problems, other works also investigate the role of LLMs in causal discovery [42, 20, 37, 5], which focuses on

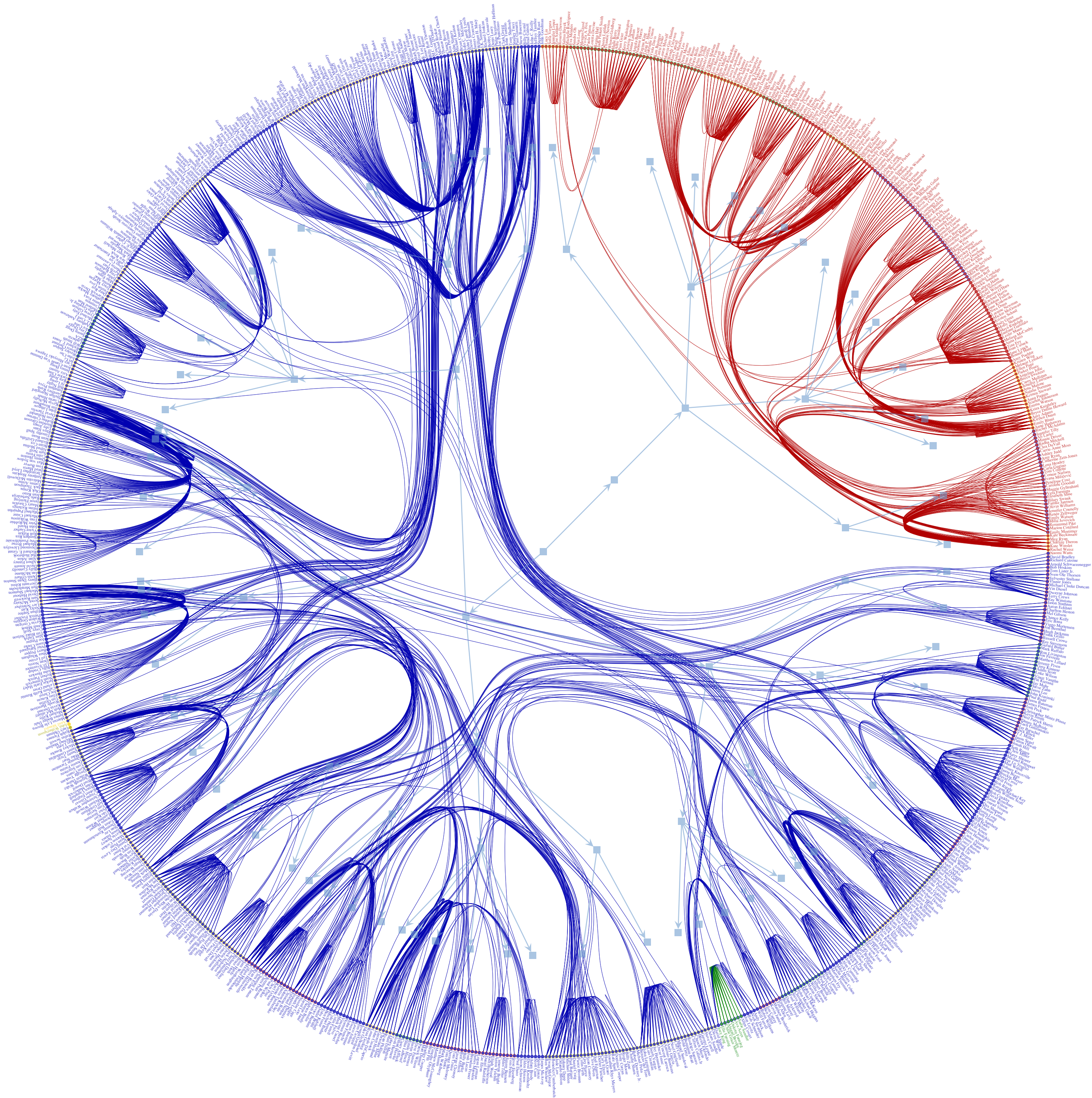

Figure 5: Graph prior that represents actor similarity relationships, generated using prior knowledge from ChatGPT, showing connections among 901 actors. Each edge indicates a similarity between the connected actors. Blue nodes represent male actors, red nodes represent female actors, and green nodes represent child actors. Two isolated nodes are highlighted in yellow and olive.

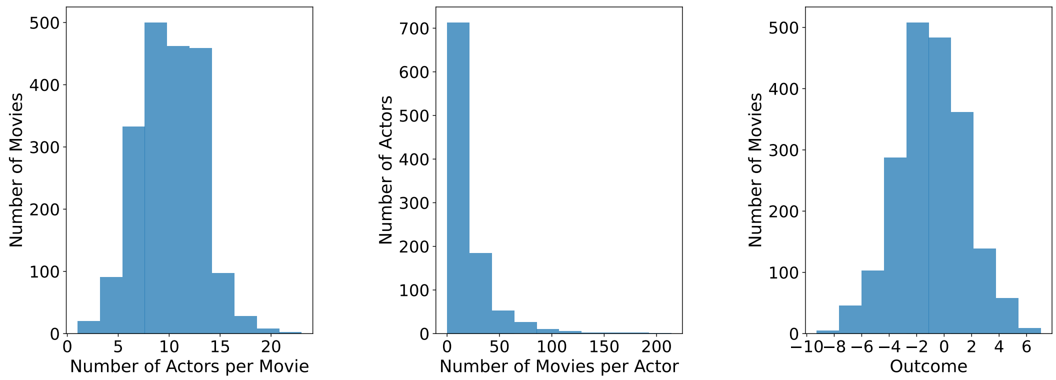

Figure 6: Statistical Distribution Characteristics of the Synthetic Dataset. The dataset mirrors real-world distributions, with both the number of actors and ground truth outcomes adhering to a normal distribution. Furthermore, the distribution of movies per actor indicates that most actors participate in a limited number of movies, paralleling real-world trends.

recovering instead of generating causal graphs. While most of the previous research is focusing on causal discovery, our research is trying to expand the boundaries of LLMs in graph-guided causal estimation, as a reliable foundation model for structural prior retrieval.

C Constructing the Semi-synthetic Dataset

When generating the semi-synthetic dataset, we use the genres of a movie in the original dataset as the confounder, i.e. $x \in \{0,1\}^{18}$. The actors are the causes and the outcome of interest Y is the ROI of a movie. Only the confounders are from the real-world distribution, both treatments and outcomes are generated. We suppose there are $m = 1000$ actors in total so a treatment is a binary vector $v \in \{0,1\}^{1000}$ indicating whether 1000 actors are in the movie or not. The relation of the confounder genre and the cause actors is defined by a preference matrix W:

$$p_v = \alpha \cdot softmax(Wx) \tag{5}$$

where $W \in \mathbf{R}^{n \times m}$ and each element of it follows the normal distribution $\mathcal{N}(0,1)$. The probability of an actor j appeared in the movie is then $P[v_j = 1] = p_{v_j}$. We sample the actors of a movie from the Bernoulli distribution $\mathcal{B}(p_{v_j})$ accordingly. α is the expected number of actors per movie so that given a movie, $E[\sum_{j=1}^{m} v_j] = \alpha$. We set α to be 10 in our experiments to be reasonably close to a real-world setting. The ground truth causal and confounder effect θ_v and θ_x are generated from a uniform distribution $\mathcal{U}(-1,1)$. We then define the linear outcome model:

$$y_i = \theta_v^T v_i + \theta_x^T x_i + \epsilon_y \tag{6}$$

as we assume the linear model and $\epsilon_y \sim \mathcal{N}(0,1)$. With the ground truth causal effect, we use the Gaussian kernel to to measure the similarity between the actors to create a binary graph prior A:

$$A_{ij} = \mathbb{1}\left[\exp \frac{(\tilde{\theta}_{v_i} - \tilde{\theta}_{v_j})^2}{2\sigma_v^2} \geq T\right] \tag{7}$$

where σ_v and T are parameters to control the density of the graph. In particular, the prior is usually expected to be noisy. Therefore, we set $\tilde{\sigma}_v = \sigma_v + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$ and control σ_ϵ to inject noise of different scale in our semi-synthetic setting.

D Prior Information Graph Construction Detail

D.1 Prompts

The following section shows the complete prompts used to generate the prior information graph in this paper.

D.1.1 Intrinsic Feature Extraction

```
You will be provided with an introduction of <ACTOR_NAME> and your
    task is to summarize the following information from the text:

1. gender
2. birth
3. race
4. acting styles
5. other background information that demonstrates the intrinsic
    characteristics.

Each section should be less than a few sentences. Think step by step
    and present your final answer in JSON format.

# Information

The information for <ACTOR_NAME> is provided here:
<DESCRIPTION>

# JSON Format

The JSON format should be as follows:
{
  "name": "<name>",
  "birth": "<birthday>"
  "gender": "<gender>",
  "race": "<race>",
  "styles": "<styles>",
  "background": "<background>"
}

# Example of an output JSON

{
  "name": "Actor A",
  "birth": "October 1, 1970"
  "gender": "male",
  "race": "white",
  "styles": "Method acting, Classical theatre, Comedy",
  "background": "Actor A was born in a small town in the Midwest. He
      discovered his passion for acting in high school theatre and
      went on to study drama at a prestigious performing arts school.
      Known for his versatility, he has performed in a variety of
      roles ranging from Shakespearean plays to modern comedies."
}
```

D.1.2 Candidates Retrieval

```
# Context

You are an expert in the film industry, with a deep understanding of
    actors and actresses, including their characteristics and
    connections within the industry. Your task is to identify which 5
    actors or actresses are most similar to <ACTOR_NAME>, excluding <
    ACTOR_NAME> themselves. This is to suggest that a director could
    potentially replace <ACTOR_NAME> with one of these five
```

```
    individuals without significantly impacting the movie's outcome.
    Consider factors such as their acting style, genres they typically
     work in, gender, age, nationality, ethnicity, and other relevant
    traits. Additionally, you may weigh their professional experiences
     and past collaborations. These suggestions are merely guidelines;
     you may rely on other evidence if you believe it more accurately
    supports your analysis.

We provide the information for <ACTOR_NAME>:
<DESCRIPTION>

Please follow the steps below:
1. Provide a description of <ACTOR_NAME> with at least three sentences
    . Outline your criteria for similarity, explaining what factors
    you believe are most critical for this comparison.
2. From the provided 'Actor List', preliminarily select 10 candidates.
     At this stage, a detailed explanation for each choice is not
    required, and they do not need to be ranked or meticulously
    filtered for relevance.
Verify the presence of your candidates in the 'Actor List'. If any
    selected individuals do not appear on this list, they should be
    excluded from further consideration.
Return the final result in JSON format.

# JSON Format

The JSON format should be as follows:
[{"name": <name>, "id": <id>}, ...]

# Actor List

<ACTOR_LIST>
```

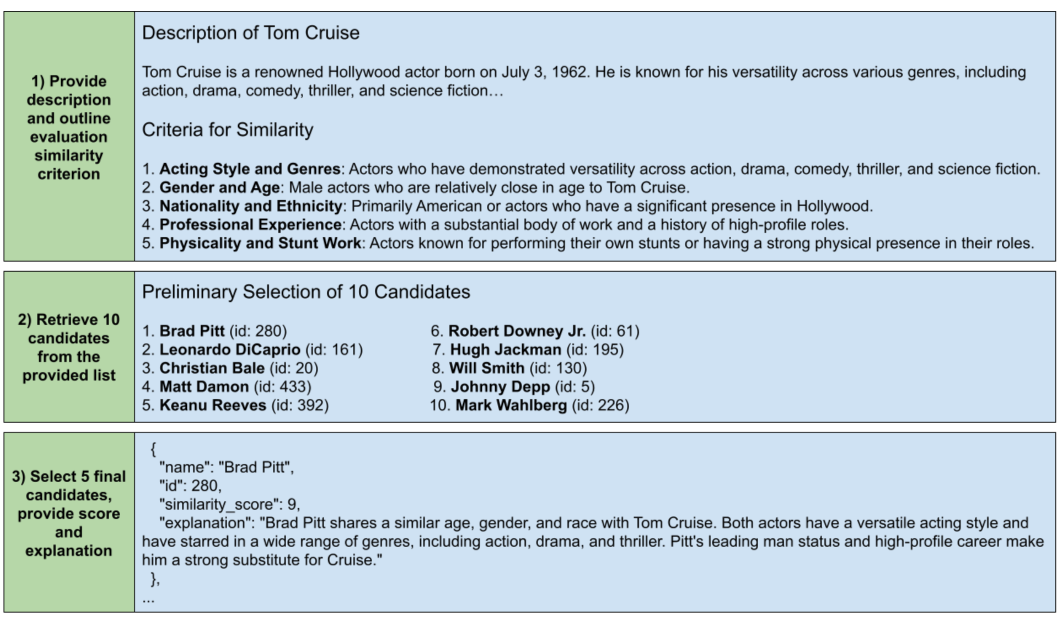

Figure 7: Illustration of leveraging LLM (ChatGPT) to generate a graph prior for actors from TMDB5000. A three-step dialogue is used to identify five actors similar to Tom Cruise. The prompting process are detailed in Appendix D

D.1.3 Candidates Reranking

```
# Context

You are an expert in the film industry, with a deep understanding of
    actors and actresses, including their characteristics and
    connections within the industry. Your task is to identify which 5
    actors or actresses are most similar to <ACTOR_NAME> from the
    candidates list, excluding <ACTOR_NAME> themselves. This is to
    suggest that a director could potentially replace <ACTOR_NAME>
    with one of these five individuals without significantly impacting
     the movie's outcome. Consider factors such as their acting style,
     genres they typically work in, gender, age, nationality,
    ethnicity, and other relevant traits. Additionally, you may weigh
    their professional experiences and past collaborations. These
    suggestions are merely guidelines; you may rely on other evidence
    if you believe it more accurately supports your analysis.

Please follow the steps below:
1. Narrow your selection down to the 5 most suitable candidates based
    on the initial criteria.
2. Present your final selection in JSON format, listing each actor's
    name, ID (as indicated in the 'Actor List'), similarity score (on
    a scale of 1-10), and a brief explanation of why each actor or
    actress is considered similar to <ACTOR_NAME>.

# JSON Format

The JSON format should be as follows:
[
  {
    "name": "<name>",
    "id": <id>,
    "similarity_score": <similarity_score>,
    "explanation": "..."
  },
  ...
]

# Example of Scoring Standard

An example is provided below to indicate how the similarity score
    should be
interpreted. This is only a guide, and you may use your own judgment
    to assign
scores.

[
  {
    "name": "Actor A",
    "id": 123,
    "similarity_score": 3,
    "explanation": "Actor A shares the action genre with <ACTOR_NAME>
        but has a distinctly different acting style, focusing more on
        physical prowess than nuanced emotional expression.
        Additionally, there's a significant age difference, limiting
        role interchangeability."
  },
  {
    "name": "Actor B",
    "id": 456,
    "similarity_score": 5,
    "explanation": "Actor B and <ACTOR_NAME> have both thrived in
        drama films and share a similar acting range. However, Actor B
         has primarily been cast in supporting roles, and their
        physical appearance is quite different, which might affect
```

```
        audience acceptance in leading roles traditionally played by <
        ACTOR_NAME>."
  },
  {
    "name": "Actor C",
    "id": 789,
    "similarity_score": 8,
    "explanation": "Actor C is very similar to <ACTOR_NAME> in age,
        acting style, and the types of characters portrayed,
        particularly in romantic comedies. While there's a slight
        difference in their career trajectories, Actor C could
        seamlessly step into roles typically associated with <
        ACTOR_NAME>."
  },
  {
    "name": "Actor D",
    "id": 1011,
    "similarity_score": 10,
    "explanation": "Actor D is almost indistinguishable from <
        ACTOR_NAME> in terms of physical appearance, acting range, and
         genre specialization. They have even been considered for the
        same roles in the past, making Actor D an ideal substitute for
         <ACTOR_NAME> in virtually any project."
  }
]

# Candidate list
Make your selection based on the list below. The descriptions for the
    actors are also provided for your reference:
<candidates_info>
```

D.2 EXAMPLE RESPONSE

```
{
  "name": "Sam Worthington",
  "birth": "August 2, 1976",
  "gender": "male",
  "race": "white",
  "styles": "Method acting, Action roles, Dramatic roles",
  "background": "Sam Worthington is an Australian actor born to
      English parents in Surrey, UK, and moved to Australia as an
      infant. He studied drama but did not finish high school and
      began his career through construction work before gaining a
      scholarship to the National Institute of Dramatic Art. Known for
       his lead roles in the Avatar franchise and Terminator Salvation
      , Worthington has received critical acclaim in both film and
      television, and is also recognized for his work in video games."
}

### Description of Sam Worthington

Sam Worthington is an Australian actor born on August 2, 1976, known
    for his method acting and roles in action and dramatic films. He
    gained international fame for his lead roles in the "Avatar"
    franchise and "Terminator Salvation." Worthington's background
    includes a mix of English and Australian influences, and he has
    received critical acclaim in both film and television, as well as
    recognition for his work in video games.

### Criteria for Similarity

1. **Acting Style**: Method acting, action roles, dramatic roles.
2. **Genres**: Primarily action and drama.
3. **Gender**: Male.
```

````
4. **Age Range**: Born between 1970 and 1985.
5. **Nationality/Ethnicity**: Preferably Australian or actors with a
   similar Western background.
6. **Professional Experience**: Experience in high-profile action and
   dramatic films, possibly with some involvement in franchises or
   video games.
7. **Past Collaborations**: Actors who have worked with similar
   directors or in similar types of projects.

### Preliminary Selection of 10 Candidates

1. **Orlando Bloom** (ID: 6)
2. **Christian Bale** (ID: 20)
3. **Tom Hardy** (ID: 24)
4. **Liam Neeson** (ID: 29)
5. **Ben Mendelsohn** (ID: 30)
6. **Chris Hemsworth** (ID: 62)
7. **Jeremy Renner** (ID: 66)
8. **Daniel Craig** (ID: 15)
9. **Jake Gyllenhaal** (ID: 298)
10. **Gerard Butler** (ID: 421)

### Verification and Final Result

All selected candidates are present in the provided 'Actor List'. Here
     is the final result in JSON format:

```json
[
    {"name": "Orlando Bloom", "id": 6},
    {"name": "Christian Bale", "id": 20},
    {"name": "Tom Hardy", "id": 24},
    {"name": "Liam Neeson", "id": 29},
    {"name": "Ben Mendelsohn", "id": 30},
    {"name": "Chris Hemsworth", "id": 62},
    {"name": "Jeremy Renner", "id": 66},
    {"name": "Daniel Craig", "id": 15},
    {"name": "Jake Gyllenhaal", "id": 298},
    {"name": "Gerard Butler", "id": 421}
]
```

Based on the provided criteria and the candidate list, here are the
   five actors most similar to Sam Worthington:

```json
[
  {
    "name": "Chris Hemsworth",
    "id": 62,
    "similarity_score": 9,
    "explanation": "Chris Hemsworth and Sam Worthington share a
       similar physical presence and have both starred in action-
       packed roles. Hemsworth's work in the Marvel Cinematic
       Universe as Thor aligns well with Worthington's roles in
       Avatar and Clash of the Titans. Both actors are Australian,
       which adds to their similarity in terms of accent and cultural
        background."
  },
  {
    "name": "Gerard Butler",
    "id": 421,
    "similarity_score": 8,
    "explanation": "Gerard Butler and Sam Worthington have both taken
       on roles in action and historical epics. Butler's performance
````
````

```
        in 300 is comparable to Worthington's in Clash of the Titans.
        Both actors have a rugged, commanding screen presence and have
         worked in similar genres, making Butler a suitable
        replacement."
  },
  {
    "name": "Tom Hardy",
    "id": 24,
    "similarity_score": 7,
    "explanation": "Tom Hardy and Sam Worthington share a penchant for
         intense, physically demanding roles. Hardy's versatility and
        experience in action films like Mad Max: Fury Road align well
        with Worthington's filmography. Although Hardy is known for
        his method acting, their overall acting styles and genre
        preferences are quite similar."
  },
  {
    "name": "Jeremy Renner",
    "id": 66,
    "similarity_score": 7,
    "explanation": "Jeremy Renner and Sam Worthington have both
        excelled in action and thriller genres. Renner's roles in the
        Bourne series and the Marvel Cinematic Universe showcase his
        ability to handle intense, action-oriented characters, similar
         to Worthington's roles in Avatar and Terminator Salvation."
  },
  {
    "name": "Jake Gyllenhaal",
    "id": 298,
    "similarity_score": 6,
    "explanation": "Jake Gyllenhaal and Sam Worthington both have a
        versatile range, though Gyllenhaal leans more towards dramatic
         and psychological roles. However, Gyllenhaal's performances
        in action-thrillers like Source Code and End of Watch
        demonstrate his capability to take on roles similar to those
        of Worthington."
  }
]
‘‘‘

These actors were selected based on their similarities in acting style
    , genre preferences, physical presence, and professional
    experiences, making them suitable replacements for Sam Worthington
    .
```

E FActScore Implementation

The FActScore evaluation is basically performed using the original implementation in https://github.com/shmsw25/FActScore. For each edge in the replacement graph, we retrieve the documents relevant to the two actors involved from wikipedia database, and prompt GPT-4o-mini for groundedness evaluation. We just skip the evaluation for certain edge if the document for either of the involved actor does not exist.

We evaluated 4161 reasonings in total. The average number of atomic facts contained in each edge reasoning is 11.20. The resulting FActScore is 85.4% with a length penalty $\gamma = 10$, which means the generation will be penalized if it is shorter than 2 sentences. Without this penalty, the FActScore is 89.5%.