Discovering Hierarchical Latent Capabilities of Language Models via Causal Representation Learning

Anonymous Authors[1]

Abstract

Faithful evaluation of language model capabilities is crucial for deriving actionable insights that can inform model development. However, rigorous causal evaluations in this domain face significant methodological challenges, including complex confounding effects and prohibitive computational costs associated with extensive retraining. To tackle these challenges, we propose a causal representation learning framework wherein observed benchmark performance is modeled as a linear transformation of a few latent capability factors. Crucially, these latent factors are identified as causally interrelated after appropriately controlling for the base model as a common confounder. Applying this approach to a comprehensive dataset encompassing over 1500 models evaluated across six benchmarks from the Open LLM Leaderboard, we identify a concise three-node linear causal structure that reliably explains the observed performance variations. Further interpretation of this causal structure provides substantial scientific insights beyond simple numerical rankings: specifically, we reveal a clear causal direction starting from general problem-solving capabilities, advancing through instruction-following proficiency, and culminating in mathematical reasoning ability. Our results underscore the essential role of carefully controlling base model variations during evaluation, a step critical to accurately uncovering the underlying causal relationships among latent model capabilities.

1. Introduction

State-of-the-art large language models (LMs) have exhibited exceptional proficiency across a wide spectrum of intricate natural language processing tasks, encompassing text generation, summarization, question answering, and creative language synthesis (BMR+20; AAA+23; GDJ+24; Ant24; AAA+24; YYZ+24; GYZ+25). These billion-parameter models are often pre-trained extensively on diverse web corpora and undergoes various post-training stages including supervised fine-tuning (SFT), reinforcement learning with human feedback (RLHF) (OWJ+22; BJN+22) to enable downstream model deployment. These complicated system engineerings make it hard to evaluate how models acquire capabilities and derive scientific claims thereafter.

In particular, rigorous evaluation of **post-training** presents notable difficulties: (i) **Costs and heterogenity in pre-training**: implementation details such as data mixture, model architecture, etc, are often proprietary and vary greatly across institutions. For example, models might be subject to contamination on benchmark data (GS23); the heterogeneity of base models leads to evidence that the benefits of post-training on reasoning abilities can differ substantially even between models of comparable size (GCS+25; ZMK+25). Even with transparent pre-training recipes, training from scratch to control for these confounders through rigorous controlled studies implies prohibitive costs (CRF+24; QNA+25). (ii) **Intricate interdependencies among distinct capabilities** — such as reasoning, few-shot learning, and instruction-following further complicates evaluation. For example, fine-tuning on instruction data might not improve knowledge-intensive question answering capabilities (GEK+24; GWS+23). Although various capabilities often seem to co-emerge or interact synergistically as model scale increases (OWJ+22; ZCY+25; CSDC25; LBS+25), rigorous theoretical frameworks can aid in understanding which capability to target in post-training.

Our research pioneers a novel approach to these persistent challenges by introducing the first framework for modeling capability factors through an explicit structured representation. The cornerstone of our methodology rests on a crucial insight: the heterogeneity observed across diverse domains (HZZ+20; JS24; ZXNZ24) – rather than being merely an obstacle – actually provides valuable "multi-view" perspectives into the shared latent capability structures across dif-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ferent base models. This lens enables us to identify and characterize these underlying structures with strong guarantees.

To establish the theoretical foundation for our investigation, we propose two fundamental hypotheses:

1. **Capability-Performance Invariance:** A small, distinguishable set of latent capability factors governs benchmark performance, maintaining consistent relationships across diverse base models.

2. **Hierarchical Capability Structure:** Within any individual base model, these capabilities organize themselves into a hierarchical framework representable as a directed acyclic graph (DAG) (Pea95). In this structure, an edge $A \to B$ signifies that interventions targeting capability A can propagate through the model's internal mechanisms to influence capability B, revealing causal pathways of skill development.

While the first hypothesis has been explored in a series of recent studies (RMH24; RBK+25; PSC+24), the second represents a novel contribution to the literature, although the hierarchical structure of *human* capabilities has been an active and influential research area in philosophy (Sim12), cognitive science (Car93; And96; ABB+04; KCB09; TKGG11) and neuroscience (KOK03; BD07). To illustrate this idea, consider a language model fine-tuned on instruction-following data: such tuning may indirectly improve its ability to solve mathematical problems, since successful solutions often require adhering to precise formatting and logical sequencing—skills closely tied to instruction-following. This hypothesis formalizes a common intuition: some capabilities, like instruction-following, serve as foundational building blocks, while others, such as math problem-solving, emerge as higher-level skills that depend on these core abilities. In this context, it is essential to control for the base model, as it influences all downstream capabilities.

We formalize this hierarchical capability structure within Pearl's structural causal model framework (Pea95), treating the base model as a shared latent parent that influences all capability factors and fine-tuning as an intervention on these latent factors, as illustrated in Figure 1. Under this structural hypothesis, existing unstructured factorization approaches (such as PCA) for analyzing latent capabilities (RMH24; RBK+25; PSC+24) may fail to disentangle hierarchical latent factors due to their lack of causal constraints. Probabilistic latent-variable approaches, such as Item Response Theory (IRT) models (TTL+25) and Bayesian latent factor models (PN22), require full likelihood specifications and hand-crafted modeling assumptions. Moreover, all these approaches fail to account for the base model's overarching influence on all latent capabilities. Drawing inspiration from the causal representation learning (CRL) literature, we propose **Hierarchical Component Analysis (HCA)** that exploits heterogeneity across base models to recover hierarchical latent capabilities with provable identifiability guarantees under mild conditions.

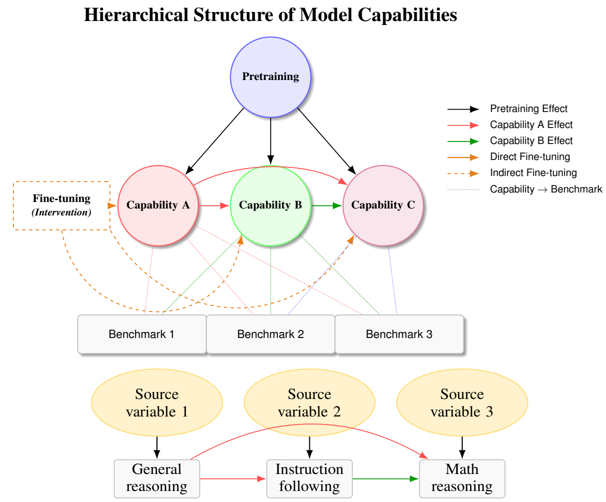


Figure 1: Example of a Hierarchical model of capabilities influencing benchmark performance (top) and hypothesized mechanism (bottom).

We apply HCA to the open LLM leaderboard data[1] and show that models fine-tuned from four base models: Qwen2.5-7B, Qwen2.5-14B, Llama-3-8B, and Llama-3.1-8B can be well-explained by a linear SCM. We further assign meaningful semantic interpretations to these factors, allowing practitioners a clear understanding of which capabilities to target during fine-tuning. Indeed, establishing explicit alignments between learned latent factors and human-interpretable concepts has remained both a significant and underexplored area within the CRL literature. To address this gap, we systematically explore correlations between latent factors, established benchmarks, and the effectiveness of prevalent leaderboard interventions. Moreover, performance on the general-reasoning parent node—encompassing benchmarks such as MMLU (HBB+20) and BIG-Bench-Hard (SSS+22)—correlates more strongly with base-model FLOPs, underscoring the importance of scaling up pre-training compute for downstream problem solving.

1.1. Notation

The eight most frequently-used base models on the leaderboard includes Llama-3-8B, LLama-3.1-8B (GDJ+24), Qwen2.5-14/7/0.5B (YYZ+24), Qwen2-7B (YYZ+24), Mistral-7B (JSM+23) and Gemma-2-9B (TRP+24). Some parts of our analysis also include other base models into study, which we will explicitly describe. We will denote these eight base models by $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_8$.

[1] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard#/

For any LM θ and benchmark B, we use $x_{\theta,\mathsf{B}}$ to denote the accuracy of θ on B, if observed. In our setting, we observe $x_{\theta_i,\mathsf{B}_j}$ for all $i \in [N]$ and $j \in [d]$, and we will simply denote this by x_{ij}. Then $\boldsymbol{x}_i = (x_{ij})_{j=1}^d$ is the observed performance vector for model θ_i. The set of all data, $\{\boldsymbol{x}_i : i \in [N]\}$, is denoted by $\mathcal{D}$. For each $\mathcal{M}_k$, we use $\mathcal{I}_k \subseteq [N]$ to denote the index set of models that use $\mathcal{M}_k$ as a base model. We also define $\mathcal{D}_k = \{\boldsymbol{x}_i : i \in \mathcal{I}_k\}$ to be the set of all data associated with $\mathcal{I}_k$. In what follows, we will sometimes abuse notation and view $\mathcal{D}' \subseteq \mathcal{D}$ as a $|\mathcal{D}'| \times d$ matrix, where each row is a performance vector $\boldsymbol{x}_j \in \mathcal{D}'$.

2. The Latent Capability Hypothesis

Recently, a line of works developed observational scaling laws (RMH24; RBK+25; PSC+24). The key hypothesis that make their analyses possible is that the observed benchmark performance is some linear transformation of low-dimensional latent capability vectors.

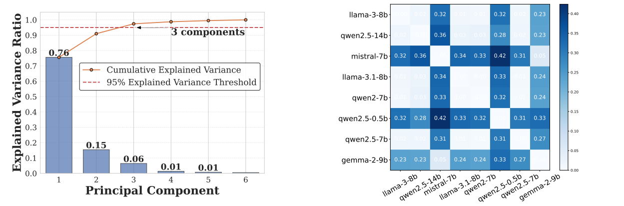
Hypothesis 0. *There exists some latent capability vector $\boldsymbol{z} \in \mathbb{R}^{d_0}, d_0 < d$ and some matrix $\boldsymbol{G} \in \mathbb{R}^{d \times d_0}$, such that $\boldsymbol{x} = \boldsymbol{G}\boldsymbol{z}$.*

Applying PCA to the leaderboard data, we discover that the performance data is approximately rank-3, aligning closely with existing works. However, we find that the low-rank pattern is *not* invariant across different model subgroups. Specifically, for each $k = 1, 2, \cdots, 8$, we apply PCA to the domain data $\mathcal{D}_k$ to obtain the rank-3 principal component subspace, and then measure the cosine distances between these subspaces. The resulting similarity matrix in Figure 2b reveals a striking pattern: five domains (with base models Llama-3-8B, Llama-3.1-8B, Qwen2-7B, Qwen2.5-7B and Qwen-2.5-14B) have roughly the same PC subspaces, whereas the other three lie distinctly apart. We define $S_{\text{inv}} = \{1, 2, 4, 5, 7\}$ to be the index set of these five models and $\mathcal{D}_{\text{inv}} = \cup_{k \in S_{\text{inv}}} \mathcal{D}_k$ to be the corresponding benchmark performance data. Notably, this heterogeneity persists under ICA (HHH+09), another popular factor analysis method, since PCA and ICA span the same component subspace, differing only in how they parametrize the independent sources within it. In what follows, we build on these observations and introduce a novel latent factor model for LM capabilities.

Remark 1. *As a brief detour, we point out that our findings on the discrepancy of the PC subspaces helps us to develop more accurate methods to fill in missing performance data on the leaderboard, as shown in Figure 15.*

2.1. A Refined Hypothesis

In view of the limitation of Hypothesis 0, we propose the following modification, restricting it to domains with identical PC subspaces the we identify in Figure 2b:



(a) Low-rank structure of the leaderboard data.

(b) Pairwise cosine distance between principal component subspaces of different domains.

Figure 2: PCA analysis of leaderboard data that recovers heterogeneous principal component structures across data from different domains.

Hypothesis 1. *The observed benchmark performance $\boldsymbol{x}_i \in \mathcal{D}_{\text{inv}}$ is governed by a set of latent capability factors $\boldsymbol{z}_i \in \mathbb{R}^{d_0}$, where $d_0 \leq d$. Moreover, there exists a linear and injective relationship between $\boldsymbol{z}_i$ and $\boldsymbol{x}_i$, meaning that there exists some matrix $\boldsymbol{G} \in \mathbb{R}^{d \times d_0}$ with full column rank such that $\boldsymbol{x}_i \approx \boldsymbol{G}\boldsymbol{z}_i, \forall i \in \mathcal{D}_{\text{inv}}$.*

In the remainder of this work, we will focus on the base models in S_{inv} and their corresponding benchmark performance data $\mathcal{D}_{\text{inv}}$.

3. Learning Hierarchical Language Model Capabilities

Our second hypothesis captures the hierarchical structure among LM capabilities by leveraging Pearl's structural causal model (SCM) (Pea95).

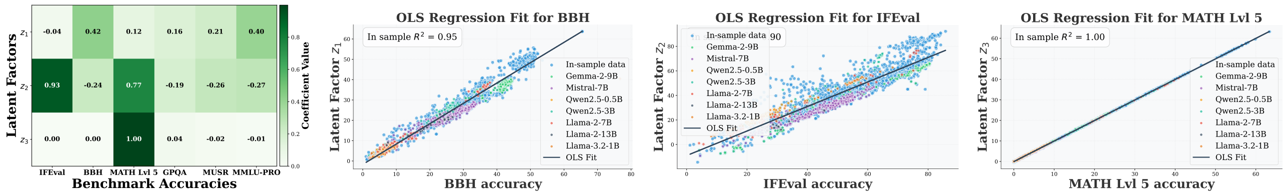
Hypothesis 2. *There exists a subset of domain indices in $S \subseteq S_{\text{inv}}$, such that for all $k \in S$, the capability factors $\boldsymbol{z}_i$ associated with $\boldsymbol{x}_i \in \mathcal{D}_k$ are generated from approximate linear SCMs, and the causal graph $\mathcal{G}$ is invariant across all k's, while the weights and errors can be domain specific.*

Formal definitions can be found in Appendix C.1. Intuitively, latent factors earlier in the topological ordering of the causal graph are primitive, while later factors are progressively less primitive, as they inherits the variability in their ancestors. We pursue two objectives: (1) recover the latent capability factors that drive observed benchmark performance, and (2) characterize precisely how those capabilities map to performance outcomes.

Formally, given observed benchmark performance vector $\boldsymbol{x}$ from K domains $\mathcal{D}_1, \cdots, \mathcal{D}_K$, our goal is to recover the linear causal model

$$\boldsymbol{z} = \boldsymbol{A}_k \boldsymbol{z} + \boldsymbol{\Omega}_k^{1/2} \boldsymbol{\epsilon}^{(k)}, \qquad \boldsymbol{x} = \boldsymbol{G}\boldsymbol{z}, \quad k \in [K], \tag{1}$$

where $(\boldsymbol{A}_k)_{ij} = w_{ij}^{(k)}$ if there exists a direct causal edge $z_j \to z_i$ in the latent graph $\mathcal{G}$ and otherwise it is zero, $\boldsymbol{\Omega}_k$ is a diagonal matrix encoding the variances of source variables.



(a) Adjusted unmixing matrix. (b) Regressing z_1 on BBH. (c) Regressing z_2 on IFEval. (d) Regressing z_3 on MATH.

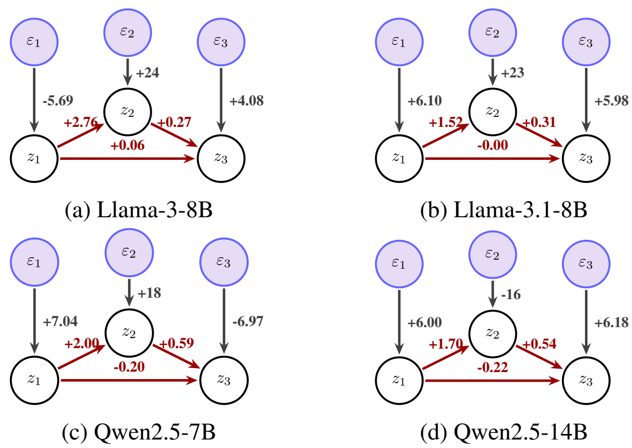
Figure 3: The unmixing matrix and the alignment between benchmarks and capabilities via OLS. We also compare the fitted OLS with the latent factor values of other base models.

$\boldsymbol{G}$ is the shared mixing matrix. For convenience, we assume that the nodes of $\mathcal{G}$ is sorted in a topological order, *i.e.*, $z_j \to z_i$ implies $j < i$. This problem has been widely studied in the causal representation learning literature. Specifically, (JS24) showed that for exact linear SCMs, assuming that the domains E_k satisfy a richness assumption, the latent causal factors are identifiable up to a benign ambiguity set, which for instance implies that one can recover the mixing matrix $\boldsymbol{G}$ up to a left multiplication of lower-triangular matrix for the causal model in Figure 5. We propose a modification of their algorithm, which we call Hierarchical Component Analysis (HCA), that is more robust to the inexactness of the SCM. More details can be found in Appendix C.

4. Experimental Results

Given its theoretical justification in the previous section, we now use HCA to recover a causal model with $d_0 = 3$ nodes that explains the observed benchmark performance of models within domains in S_{inv}. We observe that running our algorithm on the subset of $\{1, 2, 4, 7\}$, with Qwen2-7B excluded, achieves inexactness in the causal structure. This likely indicates that Qwen2-7B may deviate from the shared causal pattern of the other four base models (Llama-3-8B, Llama-3.1-8B, Qwen2.5-7B, Qwen2.5-14B). Moreover, in view of the ambiguity discussed in Appendix C.2, we run an OLS $z_i \approx \sum_{j<i} a_j z_j + \gamma_{\mathsf{B}} x_{\mathsf{B}} + c$ where x_{B} represents the performance on benchmark B. For each i, we pick B that maximizes the R^2 and replace z_i with $z_i - \sum_{j<i} a_j z_j$ to attain best-possible alignment between the recovered latent factors and their most indicative benchmarks.

The causal graphs that we recover are shown in Figure 4. The source factors ϵ_i's are normalized to have unit variance. In Figure 3a, we present the unmixing matrix (*i.e.*, linear mapping from benchmarks to latent capabilities), from which interesting patterns can be observed: z_1 is a mixture of all five benchmarks except IFEval with BBH and MMLU-Pro contributing the most, z_2 is a mixture of IFEval and MATH Lvl 5, and z_3 is almost identical to MATH Lvl 5. We will revisit these observations in the next subsection. Figure 3 further shows the results of OLS, where z_1, z_2, z_3 are observed to correlate strongly with BBH, IFEval and MATH Lvl 5, respectively. It is also important to notice that the causal conclusions we draw only apply to the four base models being considered: Figure 3 shows that the fitted OLS can have poor performance on some other base models.



(a) Llama-3-8B (b) Llama-3.1-8B (c) Qwen2.5-7B (d) Qwen2.5-14B

Figure 4: The causal graphs that we recover for each domain. The numbers represent the weights of each causal edge. For instance, in the Llama-3-8B domain, $z_2 = 2.76 z_1 + 24\epsilon_2$.

We further discuss the role of these capability factors in Appendix D. Furthermore, we conduct supervised fine-tuning on instruction data (intervening on z_2), which significantly improves MATH Lvl 5 performance (Figure 8), supporting the hypothesized causal link from z_2 (instruction-following) to z_3 (mathematical reasoning). This influence is likely because mathematical tasks demand precise adherence to instructions for correct formatting and problem interpretation, where misunderstandings severely impact accuracy.

5. Conclusion

In this work, we initiate the study of using causal-based methods to understand LM evaluation. We show that benchmark performances can be well-explained by a hierarchy of capabilities. We hope this work can help model developers design better post-training strategies, and inspire model evaluators to design leaderboards that incentivize more capable models beyond simply averaging the accuracies. We include in Appendix I a few takeaways and remarks that may be useful to practitioners.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

[AAA+23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[AAA+24] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[ABB+04] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological review*, 111(4):1036, 2004.

[ABGLP19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[And96] John R Anderson. Act: A simple theory of complex cognition. *American psychologist*, 51(4):355, 1996.

[Ant24] Anthropic. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet, 2024.

[ASR+21] Kartik Ahuja, Shiori Sagawa, Harikrishnan Ramaswamy, Edward Kung, Yair Carmon, David Krüger, Amy Zhang, and Percy Liang. Invariant risk minimization games. In *38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 145–159, 2021.

[Bar24] Matthew Barnett. An empirical study of scaling laws for transfer. *arXiv preprint arXiv:2408.16947*, 2024.

[BCE+23] Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[BD07] David Badre and Mark D'Esposito. Functional magnetic resonance imaging evidence for a hierarchical organization of the prefrontal cortex. *Journal of cognitive neuroscience*, 19(12):2082–2099, 2007.

[BdHLC22] Johann Brehmer, Pim de Haan, Phillip Lippe, and Taco S. Cohen. Weakly supervised causal representation learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 17066–17079, 2022.

[BJN+22] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[BMR+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[BR05] Heejung Bang and James M Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

[BRR+23] Simon Buchholz, Goutham Rajendran, Elan Rosenfeld, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. Learning linear causal representations from interventions under general nonlinear mixing. *Advances in Neural Information Processing Systems*, 36:45419–45462, 2023.

[Car93] John Bissell Carroll. *Human cognitive abilities: A survey of factor-analytic studies*. Number 1. Cambridge university press, 1993.

[CCZ16] Tianxi Cai, T Tony Cai, and Anru Zhang. Structured matrix completion with applications to genomic data integration. *Journal of the American Statistical Association*, 111(514):621–633, 2016.

[CHL+24] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[CPY+23] Jiaao Chen, Xiaoman Pan, Dian Yu, Kaiqiang Song, Xiaoyang Wang, Dong Yu, and Jianshu Chen. Skills-in-context prompting: Unlocking compositionality in large language models. *arXiv preprint arXiv:2308.00304*, 2023.

[CRF+24] Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, and David Owen. The rising costs of training frontier ai models, 2024.

[CSDC25] Pin-Yu Chen, Han Shen, Payel Das, and Tianyi Chen. Fundamental safety-capability trade-offs in fine-tuning large language models. *arXiv preprint arXiv:2503.20807*, 2025.

[DODH25] Ricardo Dominguez-Olmedo, Florian E. Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. In *The Thirteenth International Conference on Learning Representations*, 2025.

[EK04] Jan Eriksson and Visa Koivunen. Identifiability, separability, and uniqueness of linear ica models. *IEEE signal processing letters*, 11(7):601–604, 2004.

[FR02] Constantine E Frangakis and Donald B Rubin. Principal stratification in causal inference. *Biometrics*, 58(1):21–29, 2002.

[GCS+25] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

[GDJ+24] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[GEK+24] Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, Dinesh Manocha, et al. A closer look at the limitations of instruction tuning. *arXiv preprint arXiv:2402.05119*, 2024.

[GS23] Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*, 2023.

[GWS+23] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

[GYZ+25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[HBB+20] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[HBM+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[HDN+24] Zhenyu Hou, Pengfan Du, Yilin Niu, Zhengxiao Du, Aohan Zeng, Xiao Liu, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. Does rlhf scale? exploring the impacts from data, model, and method. *arXiv preprint arXiv:2412.06000*, 2024.

[HHH+09] Aapo Hyvärinen, Jarmo Hurri, Patrik O Hoyer, Aapo Hyvärinen, Jarmo Hurri, and Patrik O Hoyer. *Independent component analysis*. Springer, 2009.

[HZZ+20] Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21(89):1–53, 2020.

[JS24] Jikai Jin and Vasilis Syrgkanis. Learning linear causal representations from general environments: Identifiability and intrinsic ambiguity. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[JSM+23] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[KCB09] James C Kaufman, Jason C Cole, and John Baer. The construct of creativity: Structural model for self-reported creativity ratings. *The Journal of Creative Behavior*, 43(2):119–134, 2009.

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[KOK03] Etienne Koechlin, Chrystele Ody, and Frédérique Kouneiher. The architecture of cognitive control in the human prefrontal cortex. *Science*, 302(5648):1181–1185, 2003.

[LAD+22] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[LBL+] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

[LBS+25] Emmy Liu, Amanda Bertsch, Lintang Sutawika, Lindia Tjuatja, Patrick Fernandes, Lara Marinov, Michael Chen, Shreya Singhal, Carolin Lawrence, Aditi Raghunathan, et al. Not-just-scaling laws: Towards a better understanding of the downstream impact of language model design decisions. *arXiv preprint arXiv:2503.03862*, 2025.

[LPR+20] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 6348–6359, 2020.

[MLH+22] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[MLP+23] Ian R. McKenzie, Alexander Lyzhov, Michael Martin Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Xudong Shen, Joe Cavanagh, Andrew George Gritsevskiy, Derik Kauffman, Aaron T. Kirtland, Zhengping Zhou, Yuhui Zhang, Sicong Huang, Daniel Wurgaft, Max Weiss, Alexis Ross, Gabriel Recchia, Alisa Liu, Jiacheng Liu, Tom Tseng, Tomasz Korbak, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*, 2023.

[OWJ+22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[Pea95] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[PN22] Panagiotis Papastamoulis and Ioannis Ntzoufras. On the identifiability of bayesian factor analytic models. *Statistics and Computing*, 32(2):23, 2022.

[PSC+24] Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. Sloth: scaling laws for llm skills to predict multi-benchmark performance across families. *arXiv preprint arXiv:2412.06540*, 2024.

[QNA+25] Zhenting Qi, Fan Nie, Alexandre Alahi, James Zou, Himabindu Lakkaraju, Yilun Du, Eric Xing, Sham Kakade, and Hanlin Zhang. In search of lost language model training dynamics, 2025. Manuscript.

[RBK+25] Richard Ren, Steven Basart, Adam Khoja, Alice Gatti, Long Phan, Xuwang Yin, Mantas Mazeika, Alexander Pan, Gabriel Mukobi, Ryan Kim, et al. Safetywashing: Do ai safety benchmarks actually measure safety progress? *Advances in Neural Information Processing Systems*, 37:68559–68594, 2025.

[Rec11] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(12), 2011.

[RMH24] Yangjun Ruan, Chris J Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. *arXiv preprint arXiv:2405.10938*, 2024.

[Sim12] Herbert A Simon. The architecture of complexity. In *The Roots of Logistics*, pages 335–361. Springer, 2012.

[SLB+21] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[SMK23] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36:55565–55581, 2023.

[SSBU23] Chandler Squires, Anna Seigal, Salil S Bhate, and Caroline Uhler. Linear causal disentanglement via interventions. In *International conference on machine learning*, pages 32540–32560. PMLR, 2023.

[SSS+22] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[Stu10] Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

[SWR+22] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

[TKGG11] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

[TRP+24] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[TTL+25] Sang Truong, Yuheng Tu, Percy Liang, Bo Li, and Sanmi Koyejo. Reliable and efficient amortized model-based evaluation. *arXiv preprint arXiv:2503.13335*, 2025.

[vKBW+23] Julius von Kügelgen, Michel Besserve, Liang Wendong, Luigi Gresele, Armin Kekić, Elias Bareinboim, David Blei, and Bernhard Schölkopf. Nonparametric identifiability of causal representations from unknown interventions. *Advances in Neural Information Processing Systems*, 36:48603–48638, 2023.

[WJ22] Yixin Wang and Michael I. Jordan. Desiderata for representation learning: A causal perspective. *arXiv*, 2022.

[WTB+22] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[YYZ+24] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[ZCY+25] Xinlu Zhang, Zhiyu Zoey Chen, Xi Ye, Xianjun Yang, Lichang Chen, William Yang Wang, and Linda Ruth Petzold. Unveiling the impact of coding data instruction fine-tuning on large language models reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25949–25957, 2025.

[ZGS+23] Jiaqi Zhang, Kristjan Greenewald, Chandler Squires, Akash Srivastava, Karthikeyan Shanmugam, and Caroline Uhler. Identifiability guarantees for causal disentanglement from soft interventions. *Advances in Neural Information Processing Systems*, 36:50254–50292, 2023.

[ZLCF] Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. In *The Twelfth International Conference on Learning Representations*.

[ZMK+25] Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. *arXiv preprint arXiv:2504.07912*, 2025.

[ZTL+23] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*, 2023.

[ZXNZ24] Kun Zhang, Shaoan Xie, Ignavier Ng, and Yujia Zheng. Causal representation learning from multiple distributions: A general setting. In *Forty-first International Conference on Machine Learning*, 2024.

A. Related Work

Benchmark-driven LM capability studies. Benchmarks give researchers a shared scoreboard, letting everyone check claims about better language models instead of relying on hype. Early scaling-law studies showed that test loss falls in a smooth power curve as model size, data, and compute grow, setting a baseline for how capability should rise (KMH+20). Later work found many frontier models were under-trained for their size and mapped out a compute-efficient path that the Chinchilla model follows (HBM+22). Instead of running new model sweeps, (RMH24) proposed observational scaling laws involving *latent capability factors*, that depend on the model family and are obtained by PCA. They showed that benchmark performances are inherently low-rank and 3 principal components are sufficient to obtain good fitting performance. This approach is also adopted by some follow-up works on new tasks (RBK+25) and larger sets of models (PSC+24). (DODH25) further proposed an adjustment of the scaling law based on the model release time, given the fact that later models are more likely to be ”trained on test tasks”.

While pretrained LMs exhibit predictable scaling laws post-training presents a more complex picture regarding such predictive capabilities. For fine-tuning, performance generally scales with model size and fine-tuning data (as suggested by (ZLCF)), but the ”transfer gap” between pre-training and downstream tasks is a key variable (Bar24), and pre-training metrics aren't always reliable predictors of post-tuning success. Instruction tuning demonstrates clear benefits from scaling model size and the number/diversity of instructional tasks, as shown by work on FLAN (WTB+22), T0 (SWR+22), and FLAN-PaLM (CHL+24). RLHF, crucial for aligning models with human preferences (OWJ+22), shows performance gains with larger models and more feedback. However, recent work (HDN+24) indicates RLHF might scale less efficiently than pre-training, with potential diminishing returns from increased data or reward model size under fixed conditions.

Connections between LM capabilities. Research increasingly shows that LM capabilities are not isolated but form a complex, interconnected system. Studies reveal strong synergies, such as the bidirectional enhancement between coding and reasoning abilities (ZCY+25; BCE+23), and how strong reasoning underpins mathematical problem-solving (LAD+22). Complex skills often arise from compositionality, where LMs combine simpler, foundational skills in novel ways (CPY+23).

Evidence also points towards latent abilities or general factors influencing performance across diverse tasks (LBL+; PSC+24). The nature of emergent abilities – skills appearing in larger models – is debated, with some questioning if they are genuinely novel or byproducts of other mechanisms (WTB+22; SMK23).

Also, there are significant trade-offs: efforts to enhance safety can sometimes reduce raw capability (CSDC25), and fine-tuning for one skill can lead to catastrophic forgetting of others (ZTL+23). Phenomena like inverse scaling further highlight these complex interactions (MLP+23). Finally, successful task transfer and in-context learning demonstrate that LMs leverage shared underlying mechanisms and representations across different tasks (MLH+22; BMR+20), underscoring the deep interrelations among their varied skills.

Causal representation learning. Causal representation learning (CRL) aims to recover latent variables and mechanisms that remain stable under interventions and distribution shifts, thereby enabling robust prediction, reasoning, and control. Foundational position papers argue that learning disentangled *causal* factors is essential for machine intelligence rather than merely desirable for interpretability (SLB+21; WJ22). Most existing works are devoted to establishing identifiability of causal representations in realistic scenarios. Weakly supervised disentanglement shows that paired samples before/after unknown interventions are sufficient to identify factors without compromising downstream utility (LPR+20; BdHLC22). (vKBW+23) showed that a pair of single-node *hard* interventions on each latent factor is sufficient for full identifiability of the latent causal factors. Subsequent works generalize this to the case of single-node soft interventions (ZGS+23; SSBU23; BRR+23). Recently, there has been a surge of interest in studying identifiability under multi-node interventions, which is much more practical (JS24; ZXNZ24). Closely related to CRL, invariant Risk Minimization (IRM) and its game-theoretic variants formalize how multiple training environments can pin down causal predictors (ABGLP19; ASR+21).

To conclude this section, we summarize how CRL is related to our problem setting in Table 1.

B. Implementation Details

Supervised Fine-tuning. We use lm-eval-hardness to evaluate models before and after fine-tuning. We first test base model performance and observe that it can match the performance in Open LM Leaderboard. We train all models with standard hyper-parameters for SFT - 3 epochs, learning rate 2e-5. Moreover, noticing that the IFEval dataset lacks ground truth responses followed by the instructions, we query GPT-4 to generate responses with the prompt *”You are a helpful assistant*

Causal representation learning	Our context: learning latent LM capabilities
Observed data $\boldsymbol{X}$	$\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the accuracies of the i-th LM (denoted by θ_i) on the d benchmarks.
domain set $\mathfrak{E}$	Each domain $E_k \in \mathfrak{E}$ is defined by a base model $\mathcal{M}_k$ and the observed dataset $\mathcal{D}_k$ that contains the performance of all LM $\{\theta_i\}_{i=1}^N$ that use $\mathcal{M}_k$ as base model.
Latent causal factors $\boldsymbol{Z}$	$\boldsymbol{Z} = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N\}$ where $\boldsymbol{z}_i$ is the unobserved d_0-dimensional capability vector of θ_i that possesses some causal structure. We assume that $d_0 \leq d$.
Mixing matrix $\boldsymbol{G}$ (invariant across different domains)	The observed benchmark performance is a linear transformation of the underlying capability factors. This linear dependency does not change no matter what base model is chosen.
Identification of exact causal models	We define the notion of inexact causal models, and the objective is to minimize the inexactness.

Table 1: A comparison between linear CRL and some key elements in our context.

evaluating instruction-following ability. For each prompt, provide ONLY a direct response to the specific instruction, prefixed with 'Response: '. Keep your response concise, clear, and strictly follow the instruction without adding explanations or unnecessary information. Your response (excluding the 'Response: ' prefix) should strictly satisfy the length requirement." Moreover, we also SFT on z_1 BBH. But we observe a marginal improvement over the same BBH test sets. We hypothesize that parent nodes like z_1 are more dependent on base model FLOPs thus maybe hard to improve through fine-tuning alone.

Matching models on the leaderboard with the base models. Our algorithm for mapping LLMs to their pretraining token counts implements a hierarchical, multi-layered identification strategy with progressively decreasing confidence levels. The approach consists of four distinct identification layers:

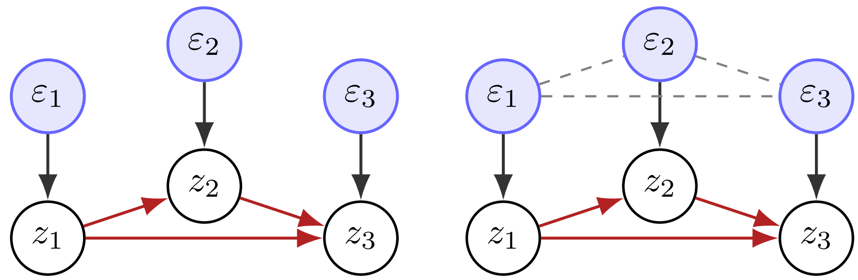
1. **Explicit Base Model Detection:** We first parse the model name for explicit references to base models with size specifications (e.g., Llama-3.1-8B). This is implemented through specialized regular expression patterns tailored to each model family's naming conventions. For instance, Gemma-2-9B is unambiguously matched to the Gemma-2-9B model trained on 8 trillion tokens.

2. **Model Name Pattern Inference:** For models lacking explicit base references, we perform broader pattern matching on model names, scanning for family indicators (e.g., "mistral", "qwen2.5") and version numbers. This layer identifies the model family but may not precisely determine the variant, necessitating parameter-based disambiguation in some cases. For example, detecting "llama-3" in the name identifies the family but requires parameter count verification to distinguish between 8B and 70B variants.

3. **Architecture-based Attribution:** Lastly, we leverage architecture information combined with parameter counts. This approach varies by model family:

   - For Llama models, we employ stringent parameter matching (e.g., 7.8-8.3B for Llama-3-8B) to prevent false positives, as many models adopt the Llama architecture without using Llama weights.
   - For other architectures (e.g., Mistral, Qwen), we implement more generous parameter ranges and higher confidence attribution, as architecture adoption typically indicates weight inheritance.
   - Size-variant mapping is crucial for families like Gemma-2, where pretraining compute differs by size (2B: 2T tokens, 9B: 8T tokens, 27B: 13T tokens).

The algorithm traverses these layers sequentially, defaulting to the highest-confidence identification available. When all layers fail to produce a sufficient confidence match, the algorithm returns null rather than making low-confidence attributions. This ensures precision over recall, maintaining the reliability of identified mappings. Upon successful model identification, we retrieve the corresponding pretraining token count from our comprehensive knowledge base, which consolidates information from research papers, technical reports, and official documentation. This multi-layered approach balances completeness with accuracy, addressing the inherent ambiguity in model naming and metadata across the diverse landscape of contemporary LLMs.



(a) A standard SCM with z_i's being the causal factors.

(b) An inexact SCM where the ϵ_i's can be dependent.

Figure 5: Illustration of Definition 1.

C. Details of HCA and Its Theoretical Guarantee

C.1. Formal definitions of linear causal models

To formally describe this latent structure, we introduce the following definition of linear structural causal models (SCMs) (Pea95).

Definition 1. *Given a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = [d_0]$ and edge set $\mathcal{E}$, a linear SCM is a data-generating process of d_0 random variables $z_1, z_2, \cdots, z_{d_0}$ with $z_i = \sum_{j \in \mathrm{pa}_{\mathcal{G}}(i)} w_{ji} z_j + \sigma_i^{1/2} \epsilon_i, i \in [d]$ with independent source variables ϵ_i wuth unit variance, where $w_{ij} \in \mathbb{R}$ are weights and $\mathrm{pa}_{\mathcal{G}}(i)$ is the parent set of i in $\mathcal{G}$.*

Intuitively, latent factors earlier in the topological ordering of the DAG are primitive, while later factors are progressively less primitive, as they inherits the variability in their ancestors.

In practice, assuming exact SCMs is often too restrictive. We define inexact SCMs below, which allows the source variables to be entangled with each other:

Definition 2. *A linear α-inexact SCM is a data generating process of $z_1, z_2, \cdots, z_{d_0}$ with $\hat{\boldsymbol{\epsilon}} = \boldsymbol{U}\boldsymbol{\epsilon}$, $z_i = \sum_{j \in \mathrm{pa}_{\mathcal{G}}(i)} w_{ji} z_j + \sigma_i^{1/2} \hat{\epsilon}_i, i \in [d_0]$ for some independent source variables ϵ_i wuth unit variance and some matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{d_0}]^\top \in \mathbb{R}^{d_0 \times d_0}$ with $\|\boldsymbol{u}_i\|_2 = 1$ and $\frac{1}{d_0} \sum_{i \neq j}^{d_0} u_{ij}^2 \leq \alpha$. Finally, for a collection $\mathcal{C}$ of α_i-inexact linear SCMs sharing the same causal graph $\mathcal{G}$, we define $\alpha = \max_i \alpha_i$ to be the maximum inexactness coefficient (MIC) of $\mathcal{C}$.*

When $\alpha = 0$, an α-inexact SCM becomes an exact SCM. Hence, the MIC measures the extent of violating the independence assumption on the source variables. Given this definition, we are ready to state our second hypothesis. A graphical illustration of exact and inexact SCMs is given in Figure 5.

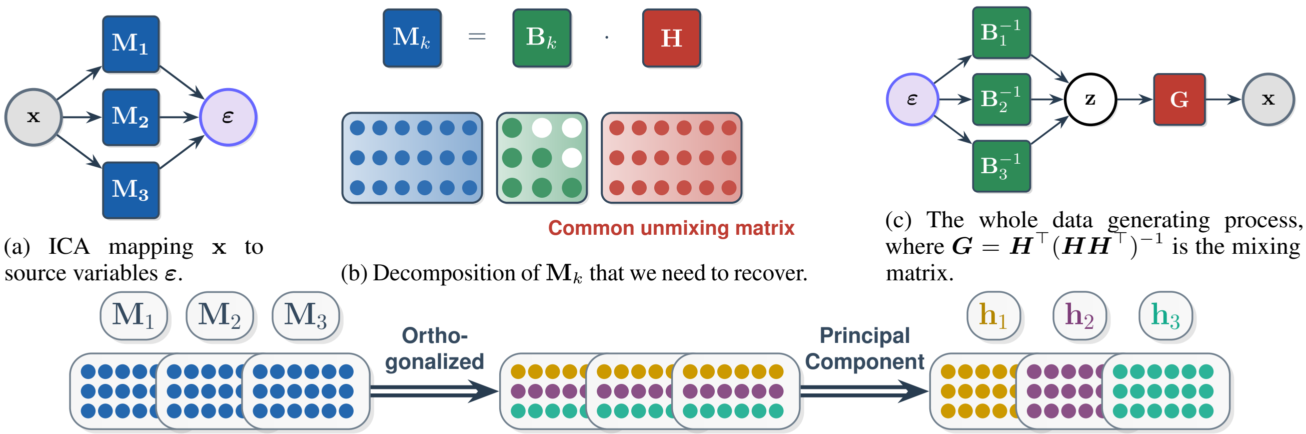
Different from all existing works that are restricted to correlation-based analysis, Hypothesis 2 characterizes a *causal* generative mechanism underlying an LM's capabilities. Specifically, given a base model B_k, each independent factor ϵ_i directly influences exactly one capability z_i, while other capabilities are either unaffected by ϵ_i or affected only *indirectly* through z_i.

C.2. Hierarchical Component Analysis (HCA)

In this section, we introduce the main ideas behind HCA, an algorithm for recovering hierarchical latent factors.

1. ICA-based unmixing. As a first step, we apply Independent Component Analysis (ICA, (HHH+09)) separately to each domain $k \in [K]$ to obtain an unmixing matrix $\boldsymbol{M}_k$ that maps independent source variables to observed benchmark data $\boldsymbol{x}$, as shown in Figure 6a. Under standard non-Gaussian assumptions in the ICA literature, these source varisbles are uniquely identified as $\boldsymbol{\epsilon}^{(k)}$ up to permutations, implying that $\boldsymbol{M}_k = \boldsymbol{P}_k \boldsymbol{B}_k \boldsymbol{H}$, where $\boldsymbol{P}_k$ is an unknown permutation matrix, $\boldsymbol{B}_k = \boldsymbol{\Omega}_k^{-1/2}(\boldsymbol{I} - \boldsymbol{A}_k)$ is lower-triangular, and $\boldsymbol{H} = (\boldsymbol{G}^\top \boldsymbol{G})^{-1} \boldsymbol{G}^\top$ is the right inverse of $\boldsymbol{G}$.

Our goal is to recover the matrices $\boldsymbol{B}_k$ and $\boldsymbol{H}$ from $\boldsymbol{M}_k$, as they allow us to recover the whole DGP as shown in Figure 6c. In partcular, the latent factors are recovered via $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}$.



(a) ICA mapping $\mathbf{x}$ to source variables $\boldsymbol{\varepsilon}$.

(b) Decomposition of $\mathbf{M}_k$ that we need to recover.

(c) The whole data generating process, where $\boldsymbol{G} = \boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{H}^\top)^{-1}$ is the mixing matrix.

(d) Finding orthogonalized matrices and extracting principal components.

Figure 6: Illustration of our setting and the key row-residual extraction step in our algorithm

2. Row-residual extraction. For any matrices $\boldsymbol{M}_k, k \in [K]$, we derive a testable equivalent condition for admitting the decomposition $\boldsymbol{M}_k = \boldsymbol{B}_k\boldsymbol{H}$. specifically, for each component index $i \in [d_0]$, we can compute the residual $\boldsymbol{r}_{k,i}$ of projecting the i-th row of $\boldsymbol{M}_k^*$ onto the span of its first $(i-1)$ rows. Then such decomposition exists if and only if $[\boldsymbol{r}_{k,i}]_{k=1}^K$ is rank 1 for all i, and $\boldsymbol{h}_i$ can be recovered (up to scale) as its principal singular vector. This process is visualized in Figure 6d.

3. Permutation alignment and factor refinement. Since each $\boldsymbol{M}_k^*$ is known only up to row permutation, we search over all permutations of the rows of $\boldsymbol{M}_k$. For each case, we apply the previous step to obtain an estimate of $\boldsymbol{H}$, and then refine each domain's weight matrix by solving $\min_{\boldsymbol{B}_k \text{ lower-triangular}} \|\boldsymbol{M}_k - \boldsymbol{B}_k\boldsymbol{H}\|_F^2$, thereby fitting the best hierarchical structure to the observed unmixing matrices. Finally, we choose the set of permutations that induces minimal MIC.

The full description HCA appears in Algorithm 2, and Appendix C.4 proves that, under an exact SCM, HCA is guaranteed to identify the underlying causal factors up to some benign ambiguities. Specifically, for the causal graph in Figure 5, $\boldsymbol{H}$ is recovered up to a left multiplication of lower-triangular matrix. Equivalently, each identified latent causal factor for z_i is a mixture of $z_j, 1 \le j \le i$. As shown in (JS24), this ambiguity is not a limitation but rather an intrinsic property reflecting equivalent models that generate identical distributional outcomes.

When the SCM is inexact, HCA recovers a data generating process

$$\boldsymbol{z} = \hat{\boldsymbol{B}}_k^{-1}\hat{\boldsymbol{\epsilon}}^{(k)}, \quad \boldsymbol{x} = \hat{\boldsymbol{G}}\boldsymbol{z}, \quad k \in [K],$$

so that $\hat{\boldsymbol{\epsilon}}^{(k)} = \hat{\boldsymbol{B}}_k\hat{\boldsymbol{H}}\boldsymbol{x}$. On the other hand, the ICA recovers $\boldsymbol{\epsilon}^{(k)} = \boldsymbol{M}_k\boldsymbol{x}$ with independent source components, so one can see that $\hat{\boldsymbol{\epsilon}}^{(k)} = \boldsymbol{J}_k\boldsymbol{\epsilon}^{(k)}$ where $\boldsymbol{J}_k = \hat{\boldsymbol{B}}_k\hat{\boldsymbol{H}}\boldsymbol{M}_k^\top(\boldsymbol{M}_k\boldsymbol{M}_k^\top)^{-1}$. This provides a guarantee on the MIC (introduced in in Definition 2):

Proposition 1. *Suppose that the ICA step is exact, then HCA recovers a linear α_k-inexact SCM for the k-th domain, where $\alpha_k = \frac{1}{d_0}\sum_{i\neq j}(\tilde{\boldsymbol{J}}_k)_{ij}^2$, $(\tilde{\boldsymbol{J}}_k)_i = (\boldsymbol{J}_k)_i/\|(\boldsymbol{J}_k)_i\|_2$. It follows that $\alpha = \max_{k\in[K]}\alpha_k$ is a valid MIC.*

Proposition 1 provides a quantitative measure of how well the recovered causal model can explain the variations in the observed benchmark data $\boldsymbol{X}^{(k)}, k \in [K]$.

C.3. HCA: Hierarchical Component Analysis

In (JS24), the authors introduced the LiNGCReL algorithm identfiability guarantees in Theorem 1 for *exact* SCMs. Here we introduce hierarchical Component Analysis (HCA) that is equivalent to LiNGCReL in the exact setting, but with several modifications to make it fit into our context.

The first step, same as (JS24), is to apply linear ICA to each individual domain. Recall that ICA's goal is to the independent signals; in our setting, it recovers the *ICA unmixing matrix* $\boldsymbol{M}_k$ that maps observed $\boldsymbol{x}$ to the source variables $\boldsymbol{\epsilon}^{(k)}$ defined in

Equation (1). This shall be carefully distincted from $\boldsymbol{H} = (\boldsymbol{G}^\top \boldsymbol{G})^{-1}\boldsymbol{G}^\top$ which is the unmixing matrix for CRL. When the SCM is exact, we would have $\boldsymbol{P}_k \boldsymbol{M}_k = \boldsymbol{B}_k \boldsymbol{H}$, where $\boldsymbol{P}_k$ is some permutation matrix. The main challenge of CRL is that we only know $\boldsymbol{M}_k = \boldsymbol{B}_k \boldsymbol{H}, k \in [K]$ but each $\boldsymbol{P}_k$ is unknown.

The second and main part of our algorithm is presented in Algorithm 2. The algorithm is motivated by the observation that, since the unmixing matrix $\boldsymbol{H}$ is the same across all domains, the structure of any row spaces of $\boldsymbol{B}_k, k \in [K]$, which are unknown, is captured by the row structures of the known ICA unmixing matrices $\boldsymbol{B}_k \boldsymbol{H}$. Moreover, given an already-recovered subgraph $\mathcal{G}_1$ of $\mathcal{G}$, one can discover some $v \notin \mathcal{G}_1$ such that $\mathrm{pa}_{\mathcal{G}}(v) \subseteq \mathcal{G}_1$, if the corresponding rows in each $\boldsymbol{B}_k$, after projecting onto the row spaces corresponding to the the orthogonal complement of the row space of already-recovered nodes, is rank-1. This is because this rank captures the "remaining degree of freedom" of v conditioned on $\mathcal{G}_1$, which equals one if and only if all its parents are in $\mathcal{G}_1$.

While this idea is close to the original LiNGCReL, some key differences are worth-noticing:

1. Compared with LiNGCReL, HCA only recovers a transitive closure $\bar{\mathcal{G}}$ of the true graph $\mathcal{G}$[2]. It is still possible to infer whether each edge in $\bar{\mathcal{G}}$ indeed exists in $\mathcal{G}$ (see appendix). For simplicity and due to the fundamental inexactness of our model, we do not perform this step here. Equivalently, we are only imposing the constraint that each $\boldsymbol{B}_k$ is upper-triangular, without assuming that any other entries are also zero.

2. The identifiability guarantee of LiNGCReL makes the restrictive assumption that the distribution $\boldsymbol{\epsilon}^{(k)}$ does not depend on k. This assumption is indeed unnecessary; the price to pay is a more complicated approach to identify the "correct matching" between the components of $\boldsymbol{\epsilon}^{(k)}$. This step could be computationally expensive, but works well in our context where d is small.

3. We determine the matrices $\boldsymbol{B}_k, k \in [K]$ by explicitly optimizing the distance between the recovered unmixing matrix and the target unmixing matrix. Compared with LiNGCReL that sets the entries of $\boldsymbol{B}_k$'s as the projection coefficients in Algorithm 1, which is theoretically equivalent for exact causal models, this extra step provides additional flexibility that optimizes the fitting quality in the presence of inexactness.

Algorithm 1 `Ortho-proj`$(S, \{\boldsymbol{A}_k\}_{k=1}^K)$

Input: Ordered set $S = \{s_1, s_2, \ldots, s_m\} \subseteq [d]$, index $i \notin S$, $\boldsymbol{A}_k \in \mathbb{R}^{d \times n}$ for $k \in [K]$.
Output: Residual matrices $\{\boldsymbol{R}_k\}_{k=1}^K$.
for $k \leftarrow 1$ **to** K **do**
  $\boldsymbol{W} \leftarrow \mathrm{span}\{(\boldsymbol{A}_k)_s : s \in S\}$; // $(\boldsymbol{A}_k)_s$ is the s-th row of $\boldsymbol{A}_k$
  $\boldsymbol{R}_k \leftarrow \mathrm{proj}_{\boldsymbol{W}^\perp}(\boldsymbol{A}_k)$;// Row-wise orthogonal projection
end

[2]The transitive closure of a directed acyclic graph (DAG) $\mathcal{G}$ is obtained by drawing an edge $i \to j$ for any i and j such that i is an ancestor in j, *i.e.*, there is a path $i = i_0 \to i_1 \to \cdots \to i_k = j$ in $\mathcal{G}$.

Algorithm 2 `Hierarchical component analysis`

Input: Matrices $\boldsymbol{M}_k \in \mathbb{R}^{d\times n}$, $k \in [K]$.
Output: The optimal unmixing matrix $\hat{\boldsymbol{H}}^*$ and weight matrices $\{\hat{\boldsymbol{B}}_k^*\}_{k=1}^K$.
Let S_d be the set of all permutations of $\{1, 2, \dots, d\}$. min_mic_score $\leftarrow \infty$ $\hat{\boldsymbol{H}}^* \leftarrow$ null; $\{\hat{\boldsymbol{B}}_k^*\}_{k=1}^K \leftarrow$ null
for *each permutation combination* $\pi = (\pi_1, \dots, \pi_K) \in (S_d)^K$ **do**
  `// 1. Apply the current permutation to each matrix`
  Let $\boldsymbol{M}'_k$ be the matrix $\boldsymbol{M}_k$ with rows permuted according to π_k, for $k = 1, \dots, K$.
  `// 2. Generate candidate` $\hat{\boldsymbol{H}}_\pi$ `based on permuted matrices`
  for $j \leftarrow 0$ **to** $d-1$ **do**
    $S_{ortho} \leftarrow \{j+1, j+2, \dots, d\}$ $\{\boldsymbol{R}'_k\}_{k=1}^K \leftarrow$ `Ortho-proj`$(S_{ortho}, \{\boldsymbol{M}'_k\}_{k=1}^K)$;
    `// Extract principal direction from the (j+1)-th rows of residuals`
    $\tilde{\boldsymbol{R}} \leftarrow [(\boldsymbol{R}'_1)_{j+1}^\top, \dots, (\boldsymbol{R}'_K)_{j+1}^\top]^\top$; `// Stack the (j+1)-th rows`
    $\boldsymbol{h}'_{j+1} \leftarrow \boldsymbol{v}_1(\tilde{\boldsymbol{R}})$; `// Top right singular vector`
  end
  $\hat{\boldsymbol{H}}_\pi \leftarrow [\boldsymbol{h}'_1, \dots, \boldsymbol{h}'_d]^\top$; `// Construct candidate H. Optionally: Gram-Schmidt`$(\boldsymbol{h}'_1, \dots, \boldsymbol{h}'_d)$
  `// 3. Compute optimal upper-triangular` $\hat{\boldsymbol{B}}_{k,\pi}$
  Let $\mathcal{T}(d)$ be the set of $d \times d$ upper-triangular matrices. **for** $k \leftarrow 1$ **to** K **do**
    $\hat{\boldsymbol{B}}_{k,\pi} \leftarrow \arg\min_{\boldsymbol{B}\in\mathcal{T}(d)} \|\boldsymbol{M}'_k - \boldsymbol{B}\hat{\boldsymbol{H}}_\pi\|_F^2$; `// Best upper-triangular estimate`
  end
  `// 4. Compute the MIC score for this permutation using Proposition 1`
  current_mic_score $\leftarrow$ ComputeMIC$(\{\boldsymbol{M}'_k\}_{k=1}^K, \{\hat{\boldsymbol{B}}_{k,\pi}\}_{k=1}^K, \hat{\boldsymbol{H}}_\pi)$
  `// 5. Update if this is the best score found so far`
  if *current_mic_score* $<$ *min_mic_score* **then**
    min_mic_score $\leftarrow$ current_mic_score $\hat{\boldsymbol{H}}^*, \{\hat{\boldsymbol{B}}_k^*\}_{k=1}^K \leftarrow \hat{\boldsymbol{H}}_\pi, \{\hat{\boldsymbol{B}}_{k,\pi}\}_{k=1}^K$
  end
end
return $\hat{\boldsymbol{H}}^*, \{\hat{\boldsymbol{B}}_k^*\}_{k=1}^K$

C.4. Identifiability Guarantee for HCA

In this subsection, we provide our main identifiability result for HCA in the special case when the graph $\mathcal{G}$ is known to be a complate DAG with $i \to j$ for all $i < j$. Equivalently, this means that each $\boldsymbol{A}_k$ is lower-triangular.

Assumption 1. *(Node-level non-degeneracy, adapted from (JS24), Assumption 5) We assume that the matrices $\{\boldsymbol{B}_k\}_{k=1}^K$ are node-level non-degenerate,* i.e.*, for all node $i \in [d]$, we have $\dim \operatorname{span}\langle(\boldsymbol{B}_k)_i : k \in [K]\rangle = \left|\mathrm{pa}_{\mathcal{G}}(i)\right| + 1$, where $(\boldsymbol{B}_k)_i$ is the i-th row of $\boldsymbol{B}_k$.*

As shown in (JS24), this assumption holds as long as the K weight vectors at node i across K domains do not lie in a low-dimensional vector space, which generally holds. To ensure identifiability, we also require that the components of noise variables are non-Gaussian and have different distributions.

Assumption 2. *For all $k \in [K]$, each component of $\boldsymbol{\epsilon}^{(k)}$ follows a different distribution, and all of them are non-Gaussian.*

Remark 2. *With a more involved procedure, (JS24) showed that one can identify $z_i, i \in [d]$ up to a "surrounding node ambiguity" in the case when $\mathcal{G}$ is unknown. Specifically, this means that the $\mathcal{G}$ can be fully recovered and the identified factor z'_i is some linear combination of z_j's with $j \in \mathrm{sur}_{\mathcal{G}}(i) := \{i\} \cup \{i' \in \mathrm{pa}_{\mathcal{G}}(i) : \mathrm{ch}_{\mathcal{G}}(i) \subseteq \mathrm{ch}_{\mathcal{G}}(i')\}$. Moreover, this ambiguity is intrinsic in this setting.*

Our main result is stated below:

Theorem 1. *Suppose that $K \geq d$, then if the ICA step is exact, one can recover the mixing matrix $\boldsymbol{G}$ up to a left multiplication of lower-triangular matrix. Equivalently, it recovers latent factors $z'_1, \cdots, z'_d$ where z'_i is a linear mixture of the true latent factors $z_j, j < i$.*

The remaining part of this subsection is devoted to proving Theorem 1.

By our assumption of the causal model, we know that in the k-th domain, the observations and the noise variables are related via $\boldsymbol{\epsilon}^{(k)} = \boldsymbol{B}_k\boldsymbol{H}\boldsymbol{x}$. Since we assume that the ICA is exact, the uniqueness of ICA in the non-Gaussian setting (EK04) implies that the umixing matrix that leads to independent source variables must be unique up to row permutations. In other words, there exists some permutation matrix $\boldsymbol{P}_k^*$, such that $\boldsymbol{P}_k^*\boldsymbol{M}_k = \boldsymbol{B}_k\boldsymbol{H}, \forall k \in [K]$. Without loss of generality, we also assume that $\boldsymbol{H}$ is orthonormal, since otherwise one can always consider a QR factorization $\boldsymbol{H} = \boldsymbol{U}\tilde{\boldsymbol{H}}$ where $\boldsymbol{U}$ is lower-triangular and $\tilde{\boldsymbol{H}}$ is orthonormal, and one can treat $\boldsymbol{B}_k\boldsymbol{U}$ as the new $\boldsymbol{B}_k$.

Recall that our algorithm goes through all possibilities of permutations $\boldsymbol{P}_k, k \in [K]$ and pick one with the smallest MIC. To begin with, it is not hard to see the following fact:

Proposition 2. *Suppose that $\boldsymbol{M}'_k = \boldsymbol{P}_k^*\boldsymbol{M}_k$, then running the subroutine in Algorithm 2 on $\boldsymbol{M}'_k, k \in [K]$ would give a zero MIC.*

Proof. Recall that $\boldsymbol{M}'_k = \boldsymbol{B}_k\boldsymbol{H}$. We will prove by induction that each row $\boldsymbol{h}'_i$ of the recovered matrix $\hat{\boldsymbol{H}}_\pi$ in Algorithm 2 is parallel to $\boldsymbol{h}_i$ (*).

For $i = 1$, since $\boldsymbol{B}_k$ is lower-triangular, and its diagonal entries $\boldsymbol{\Omega}_k^{-1/2}$ are nonzero, so the last rows of $\boldsymbol{B}_k\boldsymbol{H}, k \in [K]$ is a nonzero multiple of $\boldsymbol{h}_1$. By definition, $\boldsymbol{h}'_1$ is the principal component of these rows, which is obviously parallel to $\boldsymbol{h}_1$.

Suppose the conclusion holds for all $j < j_0$, we now prove it for $j = j_0$. Since $\boldsymbol{B}_k$ is lower-triangular, the induction hypothesis implies that for each $k \in [K]$, $\mathrm{span}\langle(\boldsymbol{M}'_k)_i : i < j\rangle = \mathrm{span}\langle\boldsymbol{h}_i : i < j\rangle$. By definition, $(\boldsymbol{R}_k)_j$ is the orthogonal projection of $(\boldsymbol{M}_k)_j$ onto this subspace. Notice that $(\boldsymbol{M}_k)_j \in \mathrm{span}\langle\boldsymbol{h}_i : i \le j\rangle$, is then easy to see that this projection is nothing but a constant multiple of $\boldsymbol{h}_j$, since this is the unique direction in $\mathrm{span}\langle\boldsymbol{h}_i : i \le j\rangle$ that is orthogonal to $\mathrm{span}\langle\boldsymbol{h}_i : i \le j\rangle$. Hence by definition, we have $\boldsymbol{h}'_j = \alpha_j\boldsymbol{h}_j$ for some scalar α_j. This concludes the proof of (*).

From (*), it is easy to see that the best lower-triangular estimate $\hat{\boldsymbol{B}}_{k,\pi}$ is equal to $\boldsymbol{B}$ up to some row-wise scaling, and that $\hat{\boldsymbol{B}}_{k,\pi}\hat{\boldsymbol{H}}_\pi = \boldsymbol{M}'_k$. Hence the MIC is zero by definition. □

To complete the proof of Theorem 1, we need to show that any permutation that achieves a zero MIC successfully recovers the causal graph up to transitive closure. Specifically, suppose that some permutation matrices $\boldsymbol{P}_k, k \in [K]$ leads to a zero MIC, let $\boldsymbol{Q}_k = \boldsymbol{P}_k\boldsymbol{P}_k^*$, then $\boldsymbol{M}_k = \boldsymbol{Q}_k\boldsymbol{B}_k\boldsymbol{H}$. We show that

1. $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \cdots = \boldsymbol{Q}_d$, and

2. Suppose that the j-th row of $\boldsymbol{Q}_1$ is $\boldsymbol{e}_{i_j}$, then $i_1, i_2, \cdots, i_d$ is a topological ordering of the graph $\mathcal{G}$, meaning that $\mathrm{pa}_{\mathcal{G}}(i_j) \subseteq \{i_1, \cdots, i_{j-1}\}$.

We say that a row index j is "good" if the j-th row of $\boldsymbol{Q}_k, k \in [K]$ are equal and the second condition above is satisfied up to j (*i.e.* $i_1, \cdots, i_j$ is an ancestral set of $\mathcal{G}$), and is "bad" otherwise. Then it suffices to show that all $j \in [d]$ are good.

Suppose the contrary holds, let $j = j_0$ be the smallest bad index. A zero MIC implies that $k \in [K]$,

$$\left[\sum_{i=1}^{j}(\hat{\boldsymbol{B}}_k)_{ji}\hat{\boldsymbol{h}}_i\right]\boldsymbol{M}_k^\top(\boldsymbol{M}_k\boldsymbol{M}_k^\top)^{-1} = \lambda_{kj}\boldsymbol{e}_j.$$

Hence,

$$\left[\sum_{i=1}^{j}(\hat{\boldsymbol{B}}_k)_{ji}\hat{\boldsymbol{h}}_i - \lambda_{kj}(\boldsymbol{M}_k)_j\right]\boldsymbol{M}_k^\top(\boldsymbol{M}_k\boldsymbol{M}_k^\top)^{-1} = 0 \quad \Rightarrow \quad \sum_{i=1}^{j}(\hat{\boldsymbol{B}}_k)_{ji}\hat{\boldsymbol{h}}_i - \lambda_{kj}(\boldsymbol{M}_k)_j \in V^\perp, \tag{2}$$

where V is the row space of $\boldsymbol{H}$. The last step holds since the row space of $\boldsymbol{M}_k$ is also V. However, by induction hypothesis, the first $(j-1)$ rows of $\boldsymbol{M}_k$ are exactly the $i_1, i_2, \cdots, i_{j-1}$-th rows of $\boldsymbol{B}_k\boldsymbol{H}$. The construction of $\hat{\boldsymbol{H}}$ and $\hat{\boldsymbol{B}}_k$ imply that the s-th ($s < j$) row of $\hat{\boldsymbol{H}}$ is equal to the i_s-th row of $\boldsymbol{H}$, and the first $(j-1)$ rows of $\hat{\boldsymbol{B}}_k$ are equal to the $i_1, i_2, \cdots, i_{j-1}$-th rows of $\boldsymbol{B}_k$. Let $V_i = \mathrm{span}\langle\hat{\boldsymbol{h}}_1, \cdots, \hat{\boldsymbol{h}}_i\rangle$, then we have that

$$\begin{aligned}\mathrm{proj}_{V_{j-1}^\perp}\left(\sum_{i=1}^{j}(\hat{\boldsymbol{B}}_k)_{ji}\hat{\boldsymbol{h}}_i - \lambda_{kj}(\boldsymbol{M}_k)_j\right) &= \mathrm{proj}_{V_{j-1}^\perp}\left((\hat{\boldsymbol{B}}_k)_{jj}\hat{\boldsymbol{h}}_j - \lambda_{kj}(\boldsymbol{M}_k)_j\right)\\ &= (\hat{\boldsymbol{B}}_k)_{jj}\mathrm{proj}_{V_{j-1}^\perp}(\hat{\boldsymbol{h}}_j) - \lambda_{kj}\mathrm{proj}_{V_{j-1}^\perp}(\boldsymbol{M}_k)_j.\end{aligned} \tag{3}$$

However, equation 2 implies that this quantity is equal to zero. As a result, we have

$$\operatorname{rank}\left\langle \operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{M}_k)_j : k \in [K] \right\rangle = 1. \tag{4}$$

Let the j-th row of $\boldsymbol{Q}_k$ be $\boldsymbol{e}_{j_k}, k \in [K]$. Then the above equation becomes

$$\operatorname{rank}\left\langle \operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{B}_k \boldsymbol{H})_{j_k} : k \in [K] \right\rangle = 1. \tag{5}$$

In the following, we show that this property can only hold when j is good. Note that

$$\operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{B}_k \boldsymbol{H})_{j_k} = \sum_{i \in \bar{\operatorname{pa}}_{\mathcal{G}}(j_k) \setminus \{i_1, \cdots, i_{j-1}\}} (\boldsymbol{B}_k)_{j_k, i} \operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{h}_i), \tag{6}$$

where $\bar{\operatorname{pa}}_{\mathcal{G}}(i) = \operatorname{pa}_{\mathcal{G}}(i) \cup \{i\}$. For $k \neq l$, equation 5 implies that $\operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{B}_k \boldsymbol{H})_{j_l}$ and $\operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{B}_k \boldsymbol{H})_{j_l}$ are colinear, but since $\boldsymbol{H}$ has full row rank, $\operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{h}_i), i \in [d] \setminus \{i_1, \cdots, i_{j-1}\}$ are independent, so we must have $\bar{\operatorname{pa}}_{\mathcal{G}}(j_k) \setminus \{i_1, \cdots, i_{j-1}\} = \bar{\operatorname{pa}}_{\mathcal{G}}(j_l) \setminus \{i_1, \cdots, i_{j-1}\}$. In particular, $j_k \in \operatorname{pa}_{\mathcal{G}}(j_l)$ and $j_l \in \operatorname{pa}_{\mathcal{G}}(j_k)$, so we must have $j_k = j_l$. Thus $j_1 = j_2 = \cdots = j_K$.

By Assumption 2,

$$\operatorname{rank}\langle (\boldsymbol{B}_k \boldsymbol{H})_{j_1} : k \in [K] \rangle = \left| \operatorname{pa}_{\mathcal{G}}(j_1) \right| + 1,$$

so that

$$\operatorname{rank}\left\langle \operatorname{proj}_{V_{j-1}^{\perp}}(\boldsymbol{B}_k \boldsymbol{H})_{j_k} : k \in [K] \right\rangle \geq \left| \operatorname{pa}_{\mathcal{G}}(j_1) \right| + 1 - \left| \operatorname{pa}_{\mathcal{G}}(j_1) \cap \{i_1, \cdots, i_{j-1}\} \right|.$$

Hence it mus be the case that $\operatorname{pa}_{\mathcal{G}}(j_1) \subseteq \cap\{i_1, \cdots, i_{j-1}\}$, concluding the proof.

D. Towards a Causal Hierarchy of Interpretable Capabilities

In Appendix I, we explored the correlation between benchmark performance and the inferred latent factors. However, practically interpreting what a causal intervention entails within this framework remains unclear. The broader challenge of interpreting and intervening on latent factors is a longstanding and unresolved issue within causal representation learning, with no universal methodology currently available. By further analyzing the models included in leaderboards, we propose hypotheses regarding these latent capabilities, supported by reliable empirical evidence.

Interpreting z_1 (Foundational General Capability). As a root node in the causal graph, z_1 a root node in our causal graph, likely represents a foundational, generalized capability. This interpretation is supported by its positive influence across nearly all benchmarks (see mixing matrix in Figure 3a), consistent with the expectation that enhancing a general capability should broadly improve downstream task performance. Interestingly, we find that model performances on benchmarks well-aligned with general capabilities, such as BBH, roughly follows a sigmoid scaling law described by: $Y \approx L/(1 + \exp(-k(\log C - \log C_0)) + \tau T + b$, where L, k, C_0, b, τ are unknown parameters, C is the pretraining compute and T is a binary variable distinguishing fine-tuned models ($T = 1$) from from pretrained-only models ($T = 0$) as shown in Appendix D. This relationship suggests that LMs' general capabilities are predominantly determined by pretraining compute resources and experience comparatively modest enhancements during subsequent post-training procedures.

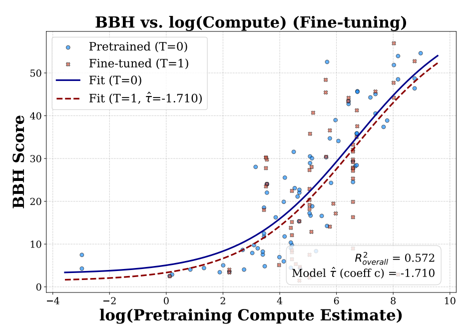
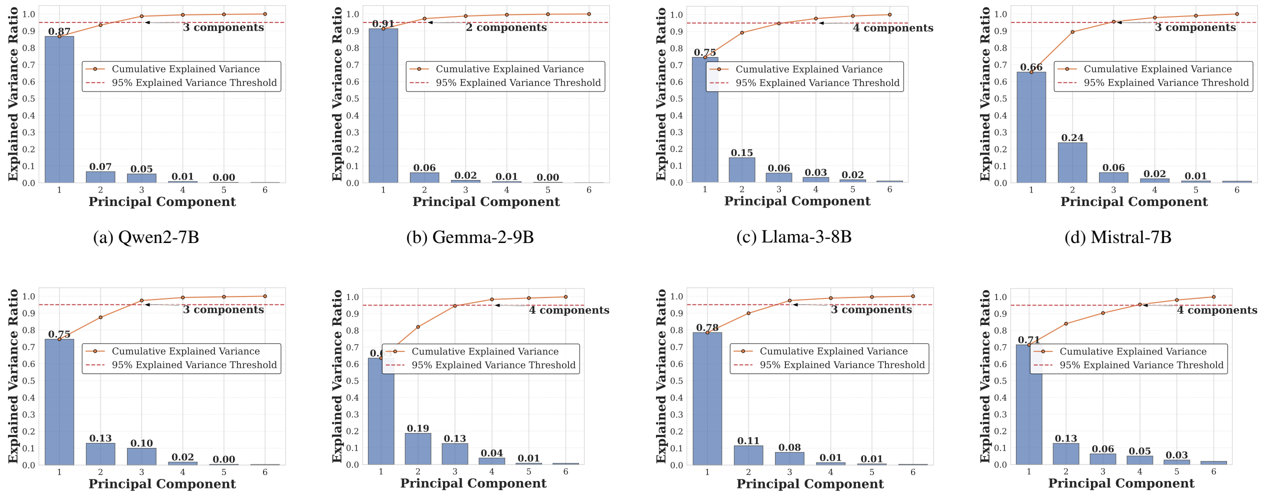


Figure 7: Sigmoid scaling law for BBH performance.

Interpreting z_2 (Instruction Following). z_2 strongly correlates with IFEval, suggesting it embodies instruction-following capability. Official instruct-tuned models (Figure 8), acting as proxies for interventions on z_2, show minimal impact on BBH, GPQA, MUSR, and MMLU-Pro for the first three base models, aligning with its mixing matrix pattern (Figure 3a).

Interpreting z_3 (Advanced Mathematical Reasoning). z_3 highly correlates with the MATH Lvl 5 benchmark, suggesting it represents an advanced mathematical reasoning capability. Isolating this capability through fine-tuning is difficult; targeted



(a) Qwen2-7B (b) Gemma-2-9B (c) Llama-3-8B (d) Mistral-7B

(e) Qwen2.5-7B (f) Qwen2.5-14B (g) Llama-3.1-8B (h) Qwen2.5-0.5B

Figure 9: Results for running PCA on individual domains.

mathematical fine-tuning often causes catastrophic forgetting (ZTL+23), reducing performance on other tasks and thus likely affecting z_1 and z_2. Identifying fine-tuning strategies that selectively enhance this specific mathematical reasoning (z_3) without negatively impacting other core capabilities (z_1, z_2) remains a critical open question.

E. Additional PCA Analyses

In this section, we examine different ways to choose the domains from the open LM leaderboard and discuss our findings.

Model	Config	BBH	IFEval	MATH	GPQA	MUSR	MMLU-PRO
Llama-3-8B	Base	0.46	0.12	0.05	0.33	0.37	0.33
	IFEval SFT	0.49	**0.50**	0.05	0.32	0.38	0.33
	Official Instruct	0.50	**0.74**	**0.09**	0.26	0.36	0.37
Qwen2.5-7B	Base	0.54	0.33	0.23	0.32	0.44	0.44
	IFEval SFT	0.55	**0.50**	**0.28**	0.33	0.43	0.44
	Official Instruct	0.54	**0.76**	**0.50**	0.29	0.40	0.43
Qwen2.5-14B	Base	0.61	0.36	0.29	0.40	0.45	0.53
	IFEval SFT	0.63	**0.55**	**0.32**	0.36	0.43	0.52
	Official Instruct	0.64	**0.82**	**0.55**	0.32	0.41	0.49

Figure 8: Performance of different models before and after fine-tuning (IFEval SFT). Observational data of Base and Official Instruct models are extracted from the open LM leaderboard.

Domain-specific PCA. While Figure 2a indicates that the complete leaderboard dataset is approximately low-rank, this global characteristic does not inherently imply a similar low-rank structure for benchmark performance data within individual domains. It is plausible that some domains possess full-rank data, but these higher-rank properties are obscured or averaged out when the entire leaderboard is considered. To investigate this, we performed PCA on the eight domains containing the largest number of model entries. As illustrated by the analysis of their leading principal components in Figure 9, all examined domains are effectively rank-3, with the exception of Gemma-2-9B, which exhibits an approximate rank of 2.

PCA for more base models. We first provide an extended version of Figure 2b for 20 most frequently used base models of the open LM leaderboard.

Mixture of Experts (MoE). We investigate the MoE architecture, which is used by Mixtral, and more recently, by Deepseek. The information of whether a model uses the MoE architecture is directly available from our leaderboard. In Figure 11a, we plot the principal component subspace distances between MoE models and non-MoE models. We also include two architectures upon which a vast majority of MoE models are built. We can see that there is little difference in the principal

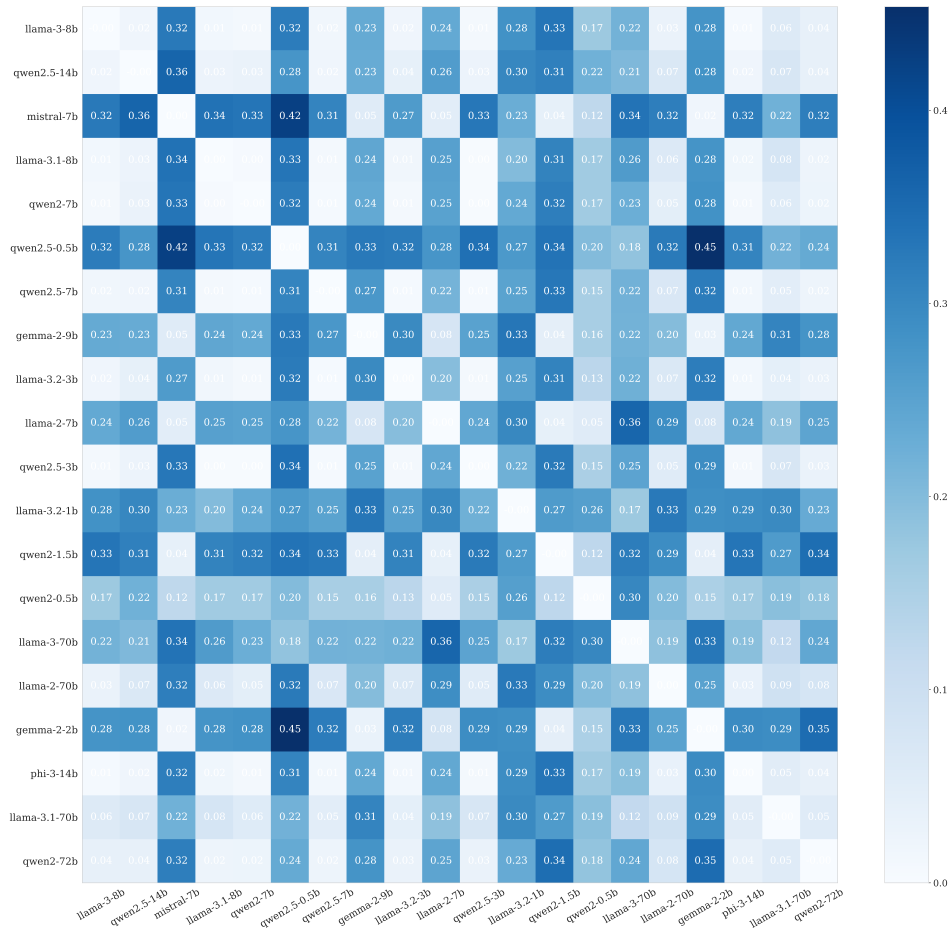
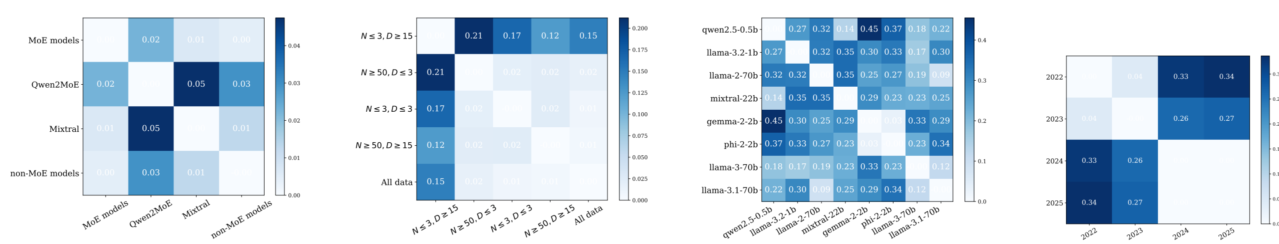


Figure 10: Pairwise cosine distance matrix for 20 base models.

component subspaces.

Different relative sizes of N and D. While the pretraining compute $C \approx 6ND$ is well-known to directly affect the model performance, the precise roles of N and D remain unclear. Our PCA results in Figure 11b considers four domains of data that contain models with small N and large D, large N and small D, small N and small D, large N and large D respectively. The finding is intriguing – it shows that the small N, large D domain has a principal component subspace that is quite different from the other domains. Further investigation by controlling for base models show that this is just a coincidence, as shown in Figure 11c. In this figure, we consider domains corresponding to the two most frequent base models for each domain used in Figure 11b. We find that while three principal component subspaces in Figure 11b look similar, they are actually the mixture of domains with very different principal component subspaces. This further hightlights the importance of controlling for the base model in causal analysis, as in the approach of our main work.



(a) MoE v.s. non-MoE. (b) Different N's and D's. (c) Different model types. (d) Model uploaded time.

Figure 11: PCA Results comparing principal component subspaces for different criteria.

Different uploaded time. Lastly, we define domains according to which year the model is uploaded. In Figure 11d, we find that there is a clear watershed between 2023 and 2024. Similar findings are also made in (DODH25), where the authors argue that after November 2023, ”training on test task” becomes more prevalent.

It should be noticed that similarity of PC subspaces is a necessary but not sufficient conditions for our causal analysis. Domains with similar PC subspaces may not be explained by a linear causal model. Moreover, we apply causal analysis to domains defined by base models primarily because this helps us remove all confounders related to the pretraining stage. On this other hand, difference in PC subspaces likely indicate some heterogeneous causal patterns. We leave the analyses of these patterns to future work.

F. Scaling Laws and The Effect of Fine-tuning

Existing predictive models for language model performances are typically restricted to pretrained models. This is not unexpected, since it is hard to characterize the performance gains in post training in terms of the relevant factors. In this section, we point out some of the key challenges in understanding the effect of fine-tuning.

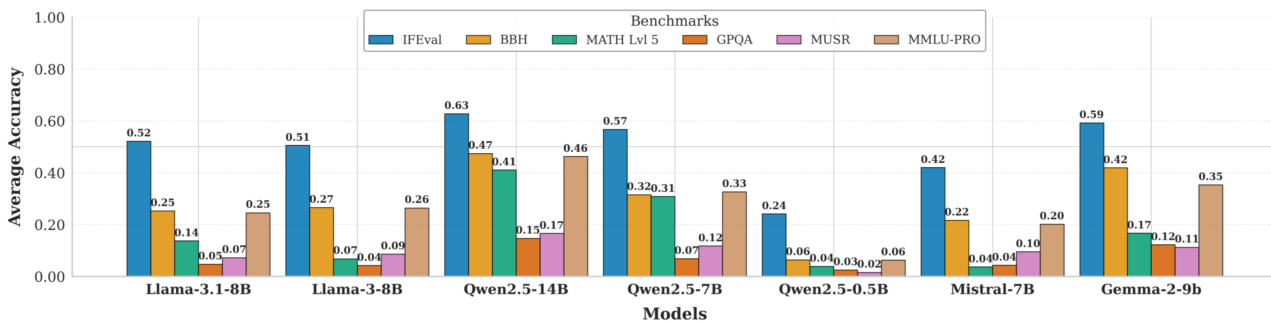


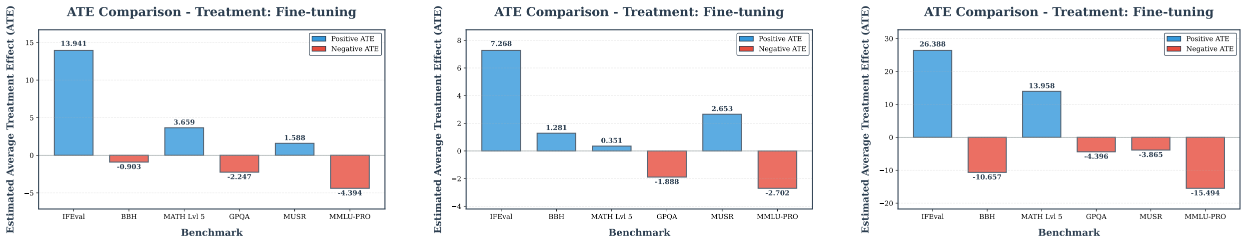
Figure 12: The average benchmark performance of fine-tuned models on the open LM leaderboard with three base models in different sizes.

As illustrated in Figure 12, models fine-tuned on more powerful base models tend to exhibit uniformly better performance across all benchmarks. In other words, base model is a common confounder of all benchmark performances. We observe that base model also confounds the amount of improvment one can achieve on all benchmarks. To illustrate this point, we estimate the average treatment effect (ATE) of T on all six benchmarks of the open LM leaderboard using the backdoor adjustment formula $\mathbb{E}[Y \mid do(T)] = \int \mathbb{E}[Y \mid T, X = x] p_X(x) \mathrm{d}x$, where $X = \log(C)$ is the log pretraining compute and $p_X(\cdot)$ its density. As illustrated in Figure 13a, fine-tuning yields substantial gains on math reasoning and instruction-following benchmarks, while producing little to negative change on general reasoning and QA-based tasks. Examining Llama- versus Qwen-based variants separately, we observe that Qwen models gain more from fine-tuning on math reasoning and instruction-following, yet incur larger drops on general reasoning.

Remark 3. *Caution is warranted when interpreting the causal implications of the estimates presented in Figure 13. These values represent true Average Treatment Effects (ATEs) only when the conditional ignorability assumption—fundamental to causal inference—is satisfied. In our context, this assumption requires that different base models experience equivalent ”distributions of interventions” across tasks. For example, if Qwen demonstrates superior performance gains compared to Llama on mathematics-related tasks, conditional ignorability would be violated if researchers strategically selected Qwen models more frequently for mathematical applications to maximize performance outcomes. We contend that this assumption is difficult to substantiate in practice. An important open research question remains: how might we circumvent this methodological challenge when limited to observational performance data? Developing robust approaches that account for such selection biases represents a significant opportunity for future work in this domain.*

F.1. Heterogeneity of Fine-tuning Effects

Scaling law (KMH+20) has been widely adopted to predict the benchmark performance from pretraining compute. Later, (RMH24; RBK+25) used sigmoid scaling laws to fit the principal components of performance data from multiple benchmarks. Could scaling law alone explain the leaderboard data? To investigate this question, we let $C \approx 6 \cdot N \cdot D$ be the



(a) Estimated ATE across all models.

(b) Estimated ATE for Llama-based models.

(c) Estimated ATE for Qwen-based models.

Figure 13: Estimates of the average effect of fine-tuning.

pretraining compute, and fit a sigmoid regression equation $Y \approx \frac{L}{1+\exp(-k(\log C - \log C_0)} + \tau T + b$, where L, k, C_0, b, τ are unknown parameters, and T is a binary treatment variable indicating whether a model is fine-tuned or pretrained.

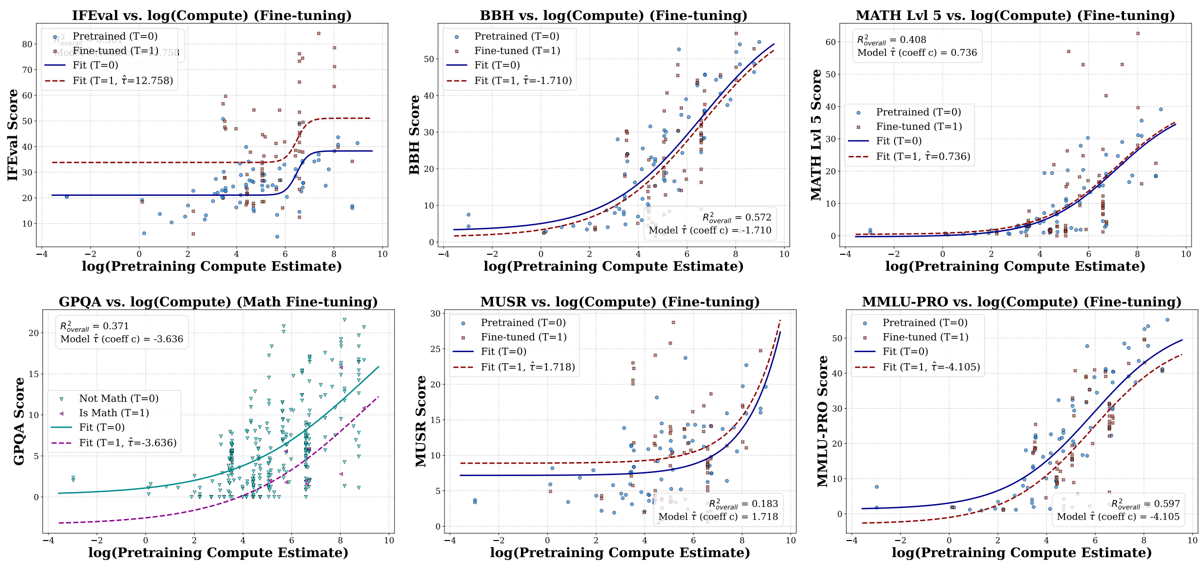


Figure 14: Sigmoid scaling laws of benchmark accuracies for pretrained and fine-tuned models. Top row: all pretrained and fine-tuned models. Middle row: Llama-based models only. Bottom row: Qwen-based models only.

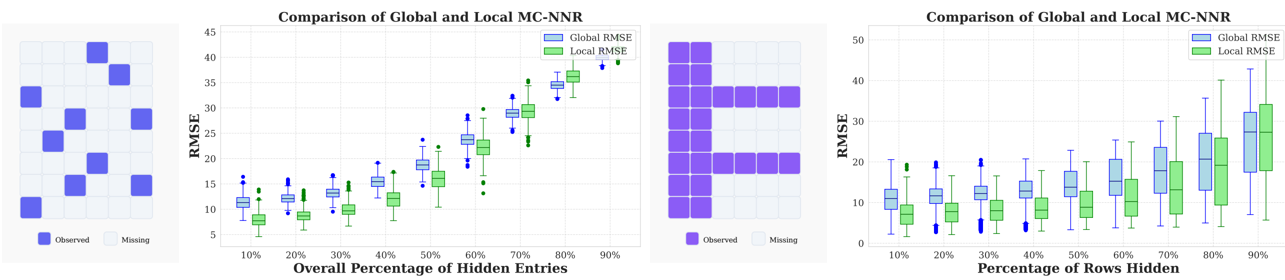
We fitted a sigmoid curve to the benchmark results of all officially released models on the leaderboard (see Figure 14). Our findings indicate that scaling laws more faithfully describe trends on BBH, MMLU-Pro and GPQA—than the remaining benchmarks. Our conjecture is that the former three benchmarks are more "knowledge-driven", in the sense that many questions in these benchmarks merely test whether the model possesses cetain knowledge. As a result, fine-tuning, mainly focusing on reasoning and alignment, can being negligible effect. By contrast, performances on tasks requiring other proficiencies (e.g. instruction following in IFEval, mathematical reasoning in MATH or multi-step soft reasoning in MUSR) are much easier to improve by fine-tuning.

G. Details for Matrix Completion

In this subsection, we provide a detailed description of the experimental setup in Remark 1. Specifically, our goal is to show how to accurately impute missing leaderboard data when the benchmark performances of LMs are only partially observed. Indeed, this task can be naturally viewed as an instance of matrix completion, where $\boldsymbol{X} \in \mathbb{R}^{N \times d}$ is the performance matrix for N models and $d = 6$ benchmarks, with missing entries.

Restricting ourselves to the missing entries in one particular domain – the group of models fine-tuned on Qwen2.5-14B – we consider a "global" and a "local" approach to perform matrix completion. In the global approach, we apply nuclear norm regularization (NNR, (Rec11)) to the whole leaderboard data $\mathcal{D} \in \mathbb{R}^{N \times d}$, while the local approach only runs NNR on the submatrix $\mathcal{D}_2$ that only contains rows in $\mathcal{I}_2$, following the notation in Section 1.1.

We conduct synthetic experiments to simulate two different scenarios. First, for the case when the benchmark accuracies are missing at random, we remove each entry of $\mathcal{X}$ independently with probability $p = 0.8$, as visualized in Figure 15 (a). Second, we consider a "block" missing pattern as visualized in Figure 15 (b), where performance on two benchmarks are fully observed, while for the ramaining four benchmarks, the performance data for a $p = 0.1, 0.2, \cdots, 0.9$ fraction of models is missing. We repeat the experiment 1000 times for the first case, and for all $\binom{6}{3} = 20$ possible sets of fully observed benchmarks of size 3 for the second case. Since standard NNR does not perform well on block missing entries, we use structured matrix completion that is designed specifically for handling this case (CCZ16). The RMSEs of the global and local approaches for these two cases are plotted in Figure 15. We can see that for the first case, the local approach is significantly more accurate than the global approach, despite the fact that it relies on fewer rows. For the second case, the local approach also performs better on average.



(a). Random missing pattern. (b). Block missing pattern.

Figure 15: RMSE of global and local matrix completion approaches for two types of missing patterns.

In Figure 16 we further plot the RMSEs for all 20 possible choices of fully observed columns. The remaining $6 - 3 = 3$ columns have rows that are missing with $p = 0.5$ probability. We observe that the local approach is always no worse than the global one, and in most cases, the local aproach leads to significant improvements.

H. Additional Experiment Results

H.1. Details for the HCA Recovery in Section 4

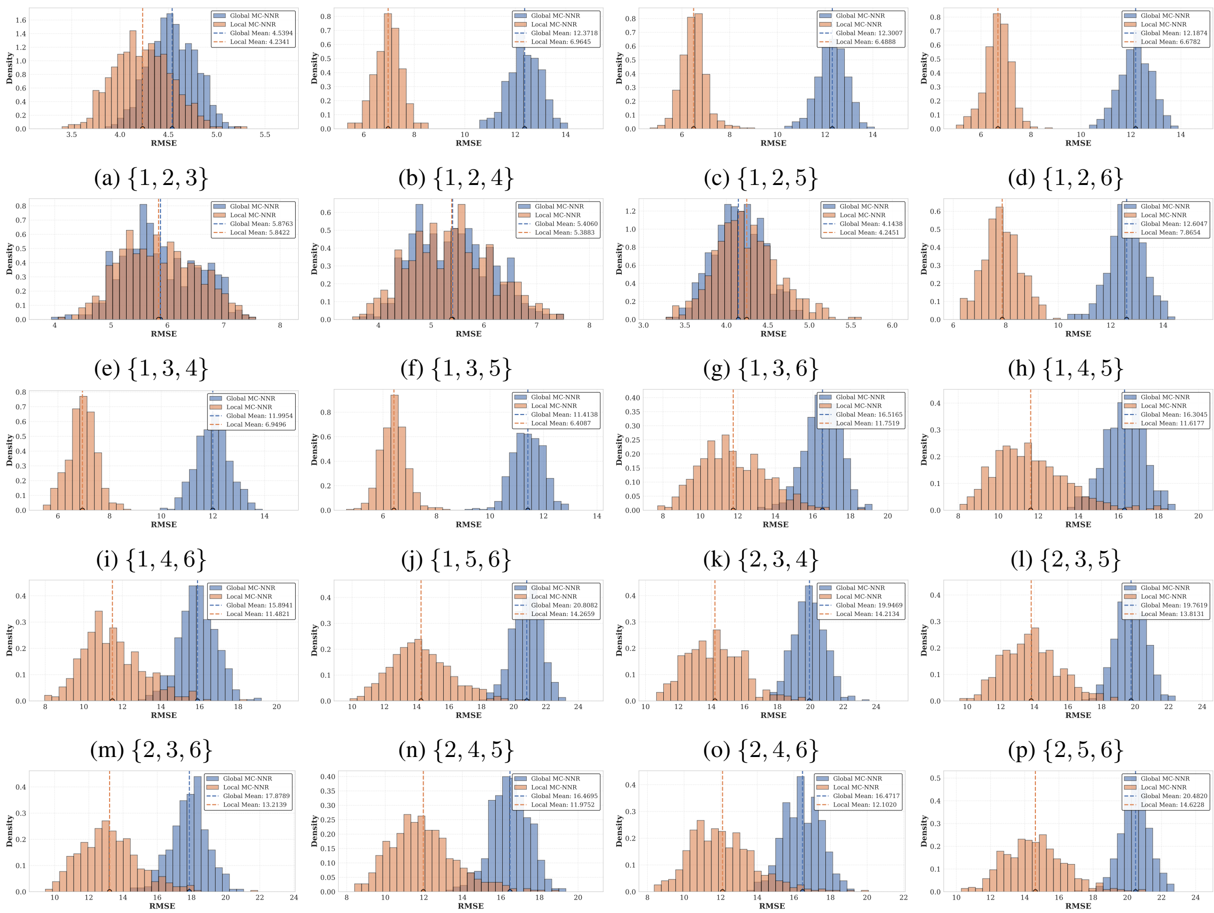
In this subsection, we provide more details and results for the recovery of the causal model in Section 4. First, we provide the visualization of the full DGP recovered by HCA *before* the OLS adjustment in Figure 17.

The OLS adjustment essentially operates on the columns of the mixing matrix in Figure 17 by subtracting from the i-th column some linear combination of the $j, j < i$ columns. We report the R^2 for aligning all six benchmarks with the three capability factors in Table 2a. The findings are particularly interesting if we consider what each benchmark is supposed to measure. Specifically, BBH and MMLU-PRO both contain tasks across different domains and are both related to language understanding and general reasoning, which, intuitively, are more fundamental capabilities. IFEval tests a model's ability of answering questions in correct formats, which is built on top of the language understanding ability. Finally, the MATH Lvl 5 benchmark requires models to answer math questions correctly *and* in the correct format, which is the most ad-hoc capability built on all the previous ones. These intuitions precisely align with the hierarchical structure of capability factors that we recover. More discussions can be found in Appendix D. A caveat is that this causal structure is only guaranteed to hold for the four base models we consider. As shown in Table 2b, the fitted OLS model can have poor performance on other base models.

H.2. Complementary Results for Section 4

MIC for all other domain subsets. In Figure 18, we report the corresponding MIC for all possible choices of domains in S_{inv}. We observe that the four subsets with smallest MIC are achieved by excluding Qwen2-7B.

Additional metrics for the recovery results. Note that one potential limitation of MIC is that it is insensitive to the



(a) $\{1, 2, 3\}$ (b) $\{1, 2, 4\}$ (c) $\{1, 2, 5\}$ (d) $\{1, 2, 6\}$

(e) $\{1, 3, 4\}$ (f) $\{1, 3, 5\}$ (g) $\{1, 3, 6\}$ (h) $\{1, 4, 5\}$

(i) $\{1, 4, 6\}$ (j) $\{1, 5, 6\}$ (k) $\{2, 3, 4\}$ (l) $\{2, 3, 5\}$

(m) $\{2, 3, 6\}$ (n) $\{2, 4, 5\}$ (o) $\{2, 4, 6\}$ (p) $\{2, 5, 6\}$

(q) $\{3, 4, 5\}$ (r) $\{3, 4, 6\}$ (s) $\{3, 5, 6\}$ (t) $\{4, 5, 6\}$

Figure 16: RMSEs of global v.s. local matrix completion for each possible set of fully observed columns. The caption below each figure indicates the columns that are fully observed. Each number representing a benchmark on the leaderboard, with $1, 2, \cdots, 6$ standing for IFEval, BBH, MATH Lvl 5, GPQA, MUSR and MMLU-Pro respectively.

	IFEval	BBH	MATH	GPQA	MUSR	MMLU-PRO
z_1	0.36	**0.96**	0.56	0.73	0.57	**0.96**
z_2	**0.92**	0.53	0.66	0.57	0.58	0.54
z_3	**1.00**	0.43	**1.00**	0.18	0.14	0.16

(a) The R^2 of running OLS on z_i using $z_j, j > i$ and the benchmark performance as controls.

	In Sample	Gemma-2-9B	Mistral-7B	Qwen2.5-0.5B	Qwen2.5-3B	Llama-2-7B	Llama-2-13B	Llama-3.2-1B
z_1	0.96	0.76	0.71	-0.2	0.89	0.97	0.66	-0.01
z_2	0.92	0.94	0.74	-1.18	0.73	0.98	0.45	0.9
z_3	1	1	1	0.99	1	1	1	1

(b) The R^2 of the fitted OLS on out-of-sample performance data with different base models.

Table 2: The precise alignment of underlying factors with established benchmarks, coupled with their ability to extend effectively across diverse model domains.

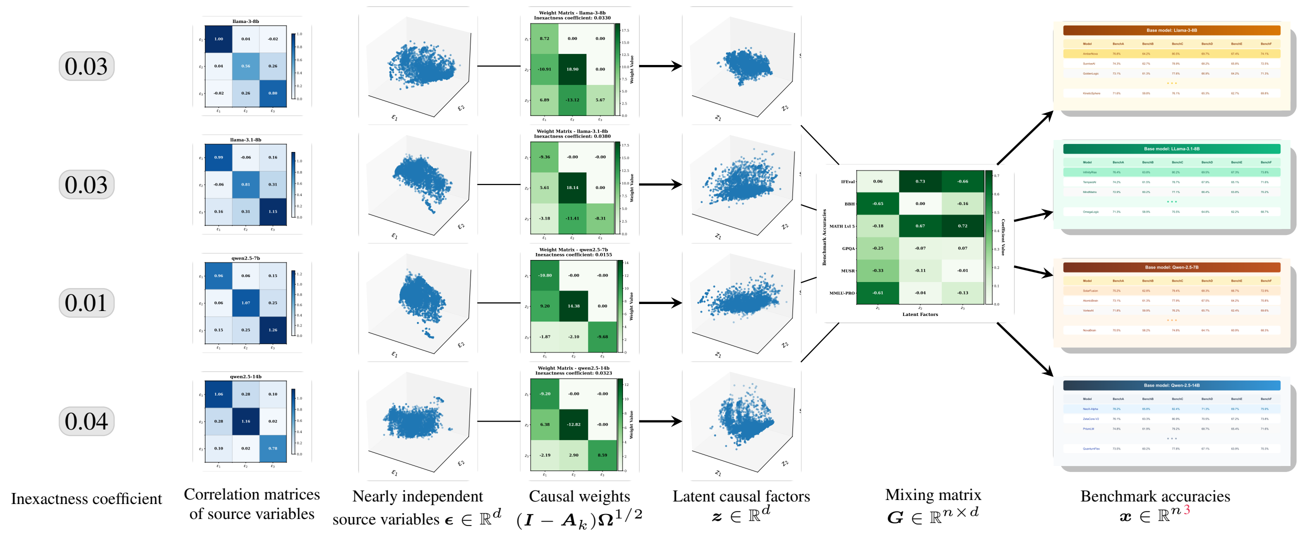


Figure 17: HCA's recovery of the DGP, including the linear SCM (second column) and mixing matrix (fourth column) on four domains (base models): Llama-3-8B, Llama-3.1-8B, Qwen2.5-7B and Qwen2.5-14B. Here, we have $n = 6$ benchmarks, $d = 3$ latent factors and $K = 6$ domains.

orthogonal complement component of each row in $\hat{B} - \hat{H}$ relative to M_k. Therefore, we present two additional metrics indicating how well our causal model fits the observed data, as shown in Table 3. We introduce these metrics since they directly measure how close $\hat{B}_k\hat{H}$ is to the true ICA mixing matrix M_k.

Node	z_1	z_2	z_3
Rank-1 error	0.05	0.13	0.02

(a) The amount of variation in $\tilde{R}$ defined in Algorithm 2 uncaptured by a rank-1 matrix in each iteration.

Domain	Llama-3-8B	Llama-3.1-8B	Qwen2.5-7B	Qwen2.5-14B
Unmixing error	0.17	0.20	0.16	0.23

(b) The relative recovery error of the unmixing matrix of ICA for each domain, calculated from $\|M_k - B_k H\|_F / \|M_k\|_F$, where M_k is the ICA unmixing matrix of the k-th domain, B_k is the inverse of the recovered weight matrix, and H is the unmixing matrix of CRL.

Table 3: Additional metrics on how good our causal model explains the observed data.

Low-rank approximation error of our causal model. Recall that we hypothesize that the data is generated from a linear causal model with 3 nodes. This necessarily requires that the performance data across all 6 benchmarks is a matrix with rank at most 3. While we have seen in Figure 2a that this is approximately the case, here we revisit this assumption and see investigate the error induced by this assumption.

In Figure 19, we plot the approximation errors of the subspace spanned by the values of three latent factors $z_i, i = 1, 2, 3$ learned via our algorithm. One can see that the low rank subspace approximates 4 out of 6 benchmarks nearly perfectly. The relatively poor fitting for the remaining two, namely GPQA and MUSR, is partially due to the fact that the model's accuracies on them are systematically lower than the remaining ones. As a result, they would be ignored to some extent when picking the principle components. In terms of the MSE, the error of fitting GPQA is comparable to the remaining four, while that of MUSR is significantly higher.

This highlights a limitation in our current methodology: although we introduce the notion of inexact causal graph for more flexibility, the assumption that each two latent factors have a causal relationship is still restrictive. For instance, it is possible

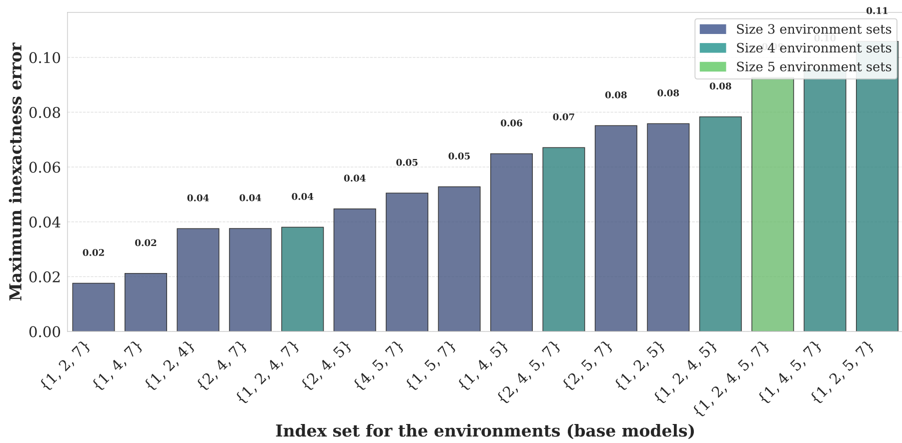
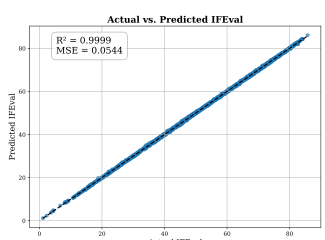
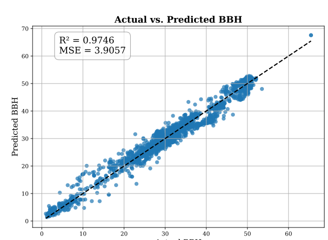


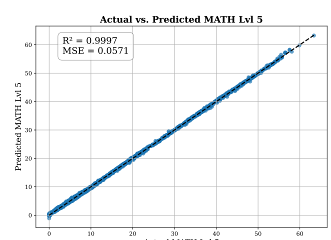
Figure 18: Overview of the MIC obtained by difference choices of domain indices. Here, as we indicated in Section 1.1, indices $1, 2, 4, 5, 7$ correspond to base models Llama-3-8B, Qwen2.5-14B, Llama-3.1-8B, Qwen2-7B and Qwen2.5-7B respectively.



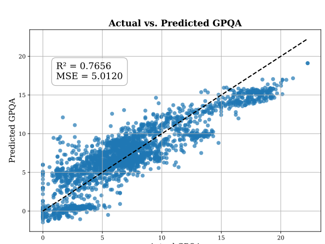
(a) IFEval



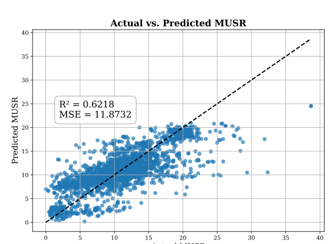
(b) BBH



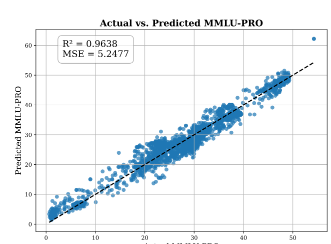
(c) MATH Lvl 5



(d) GPQA



(e) MUSR



(f) MMLU-Pro

Figure 19: Approximation error of the low-rank latent factor space for the observed benchmark performances.

that z_1, z_2 are correlated but there exists no causal relationship between them, and both of them causally affect z_3. It will be an interesting future direction to investigate how to identify the latent factors in these cases.

H.3. Filtering Out "badly" Fine-tuned models

We notice that some models on the leaderboard are badly fine-tuned, so that their benhmark performances are even worse than the pretrained model. In this subsection, we provide results of our causal analysis with these bad models removed. Removing the bad models allow us to characterize the hierarchical relationship between capabilties that is restricted to "good" fine-tuning strategies. The recovered DGP is shown in Figure 20. After adjusting for the ambiguity as we did in Section 4, we obtain the causal graphs shown in Figure 21. Finally, in Figure 22, we plot the unmixing matrix after adjustment and the relationship between each latent factor and the most indicative benchmark.

The overall pattern that our algorithm discovers is the same as the unfiltered approach. However, we notice that in the filtered case, the MIC is much larger, indicating that the causal model is less well-fitted. This is likely due to the fact that after filtering, the variance of performances on BBH and MMLU-Pro becomes significantly smaller, so that the weights of GPQA and MUSR in z_1 are larger compared with the unfiltered case (see Figure 3a). These two benchmarks are relatively not well-explained by our linear causal model, as we discussed in Appendix H.2.

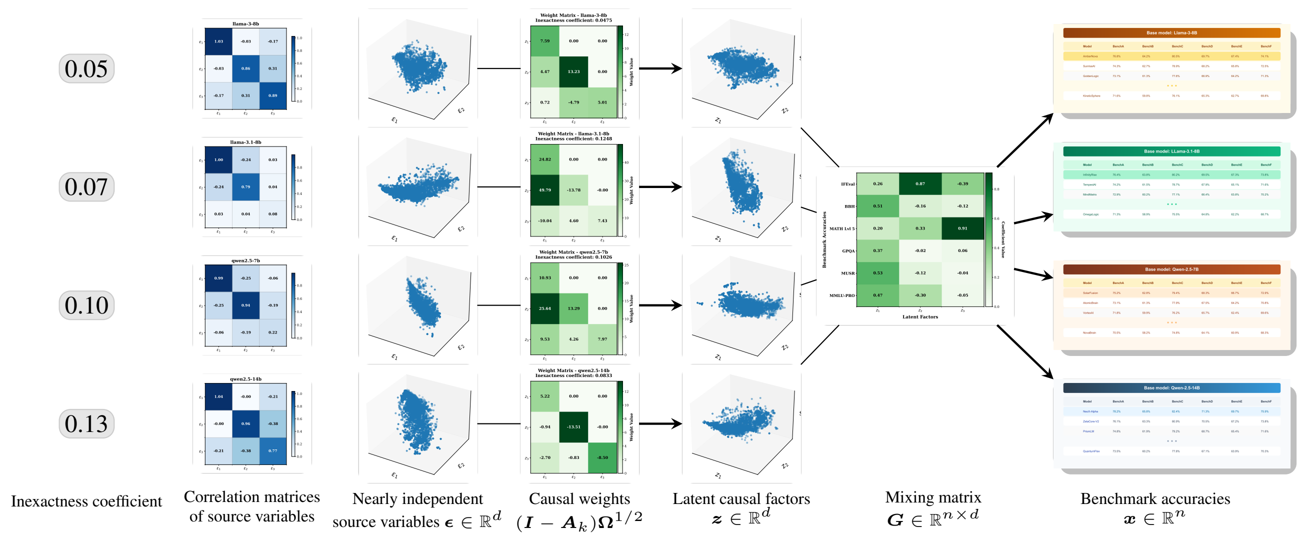
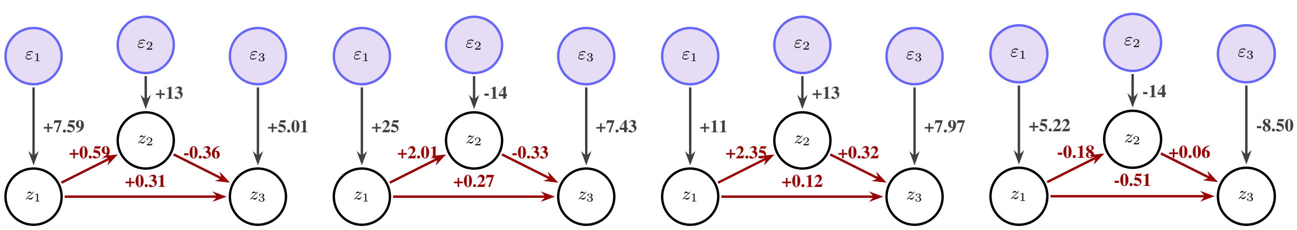


Figure 20: HCA's recovery of the DGP after removing badly fine-tuned models that have average performance lower than the pretrained model.

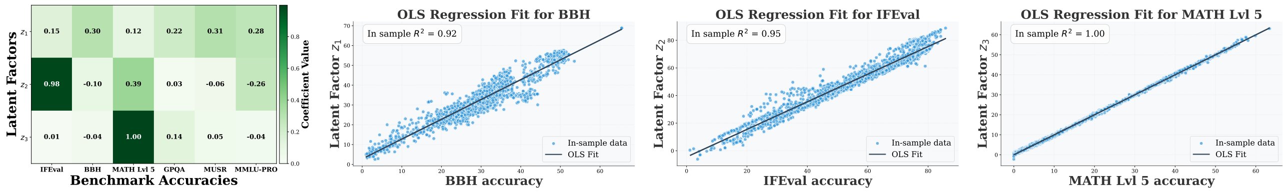


(a) Llama-3-8B (b) Llama-3.1-8B (c) Qwen2.5-7B (d) Qwen2.5-14B

Figure 21: The causal graphs recovered for different models. The numbers represent the weights of each causal edge. For instance, in the Llama-3-8B model, $z_2 = 0.59z_1 + 13\varepsilon_2$ (representing direct influences shown).

H.4. Using Open LM Leaderboard v1

We also apply our method to analyze the hierarchical structure underlying the six benchmarks used in the old version of open LM leaderboard. We choose the following six base models that are most commonly used there: Mistral-7B, Llama-2-13B, Llama-3-8B, Llama-2-7B, Llama-2-70B and Mixtral-8x7B. Similar to our previous case, we plot the pairwise



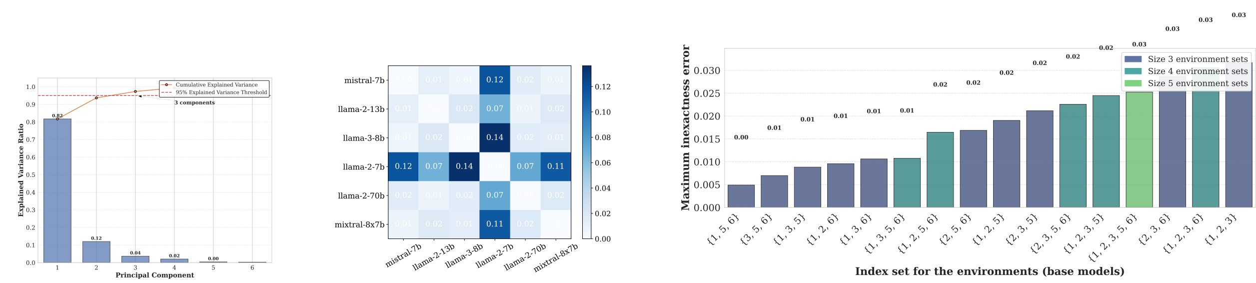
(a) Adjusted unmixing matrix. (b) Regressing z_1 on BBH. (c) Regressing z_2 on IFEval. (d) Regressing z_3 on MATH.

Figure 22: The unmixing matrix and the alignment between benchmarks and capabilities via OLS.

cosine distance between domains in Figure 23b. We denote these models by $\mathcal{M}_1, \cdots, \mathcal{M}_6$. We observe that except for Llama-2-7B, the principle component subspaces of all remaining domains are pretty close to each other, so that the invariant domain is $S_{\text{inv}} = \{1, 2, 3, 5, 6\}$. This is quite interesting, since Mixtral-8x7B uses MoE architecture, which is a fundamental difference compared with the other base models.

We then run HCA on all subsets of S_{inv} of size ≥ 3 and plot the corresponding MIC in in Figure 23c. We observe that choosing all domains in S_{inv} would still lead to a small error. The corresponding recovered DGP is presented in Figure 24. We further adjust for the ambiguity as in Section 4, and obtain the causal graphs shown in Figure 25. Finally, the adjusted unmixing matrix and the alignment between latent factors and benchmarks are presented in Figure 26.

From Figure 26, we can see a hierarchical relationship from truthfulness to general reasoning capability, and math reasoning capability. The hierarchical relationship between the latter two is consistent with our findings on the Open LLM Leaderboard v2. By looking at the causal graphs, one can observe that the weight of the edge $z_2 \to z_3$ for the Llama-3-8B domain is much larger that that of the remaining ones, which indicates that models fine-tuned on Llama-3-8B could have more performance gains on math problem solving when fine-tuned to enhance general reasoning.



(a) Low-rankness of the old Open LM leaderboard data. (b) The PCA similarity matrix for Open LM leaderboard (old version). (c) domains defined via pretraining compute.

Figure 23: figures for our analysis of Open LLM leaderboard v1.

H.5. MMLU by Task Leaderboard

The MMLU benchmark has a total of 57 subtasks, each corresponding to a distinct subject. It therefore makes sense to apply our methodology to these subjects and investigate their latent causal structure. To begin with, we first investigate the correlation between the performance of different tasks in MMLU, which is plotted in Figure 27. We observe that a majority of tasks have highly-correlated performance, although they seemingly focus on unrelated fields. This is likely due to the fact that MMLU primarily contains knowledge-based tasks, and crucially depends on the quality of the training dataset. Larger datasets likely contain more data in all disciplines and can hence lead to improvement on all tasks. In terms of causality, this means that there exists a single "knowledge" node for the MMLU benchmark as a whole.

Math-related subjects. We first select subjects that correspond to mathematics, including: MMLU_college_mathematics, MMLU_elementary_mathematics, MMLU_high_school_mathematics. In this setting, we choose the set of base models to be Mistral-7B, Llama-2-7B and Llama-2-70B, which induces a minimal MIC of 0.02. The result of HCA is summarized in Figure 28c. Counterintuitively, it shows that z_1 is close to college math, while z_2, z_3 likely represent elementary and high-school math.

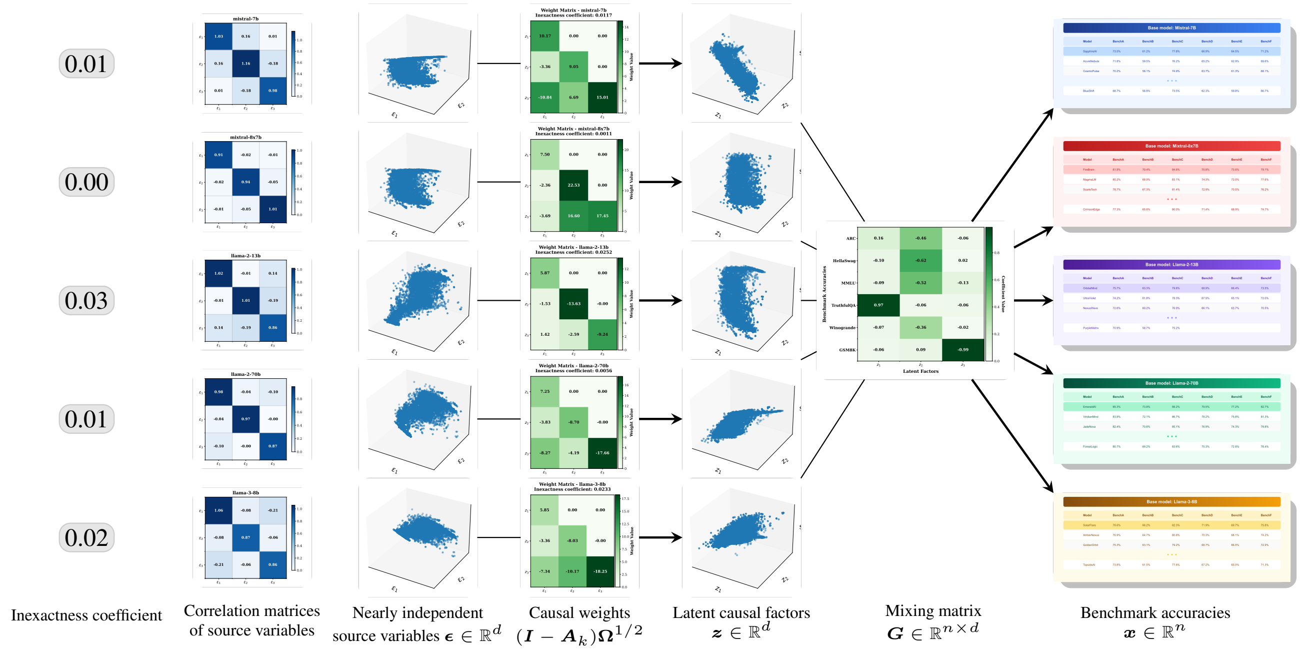
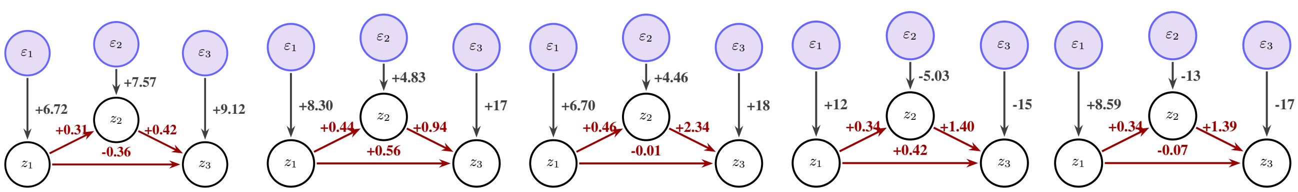
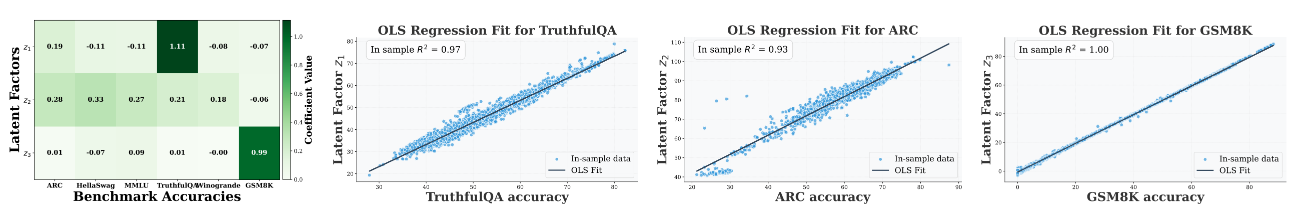


Figure 24: Results for applying our method to open LM leaderboard v1.



(a) Llama-2-13B (b) Llama-2-70B (c) Llama-3-8B (d) Mistral-7B (e) Mixtral-8×7B

Figure 25: The causal graphs recovered for different models. The numbers represent the weights of each causal edge. For instance, in the Llama-2-13B model, $z_2 = 0.31z_1 + 7.57\varepsilon_2$ (representing direct influences shown).



(a) Adjusted unmixing matrix. (b) Regressing z_1 on TruthfulQA. (c) Regressing z_2 on ARC. (d) Regressing z_3 on GSM8K.

Figure 26: The unmixing matrix and the alignment between benchmarks and capabilities via OLS.

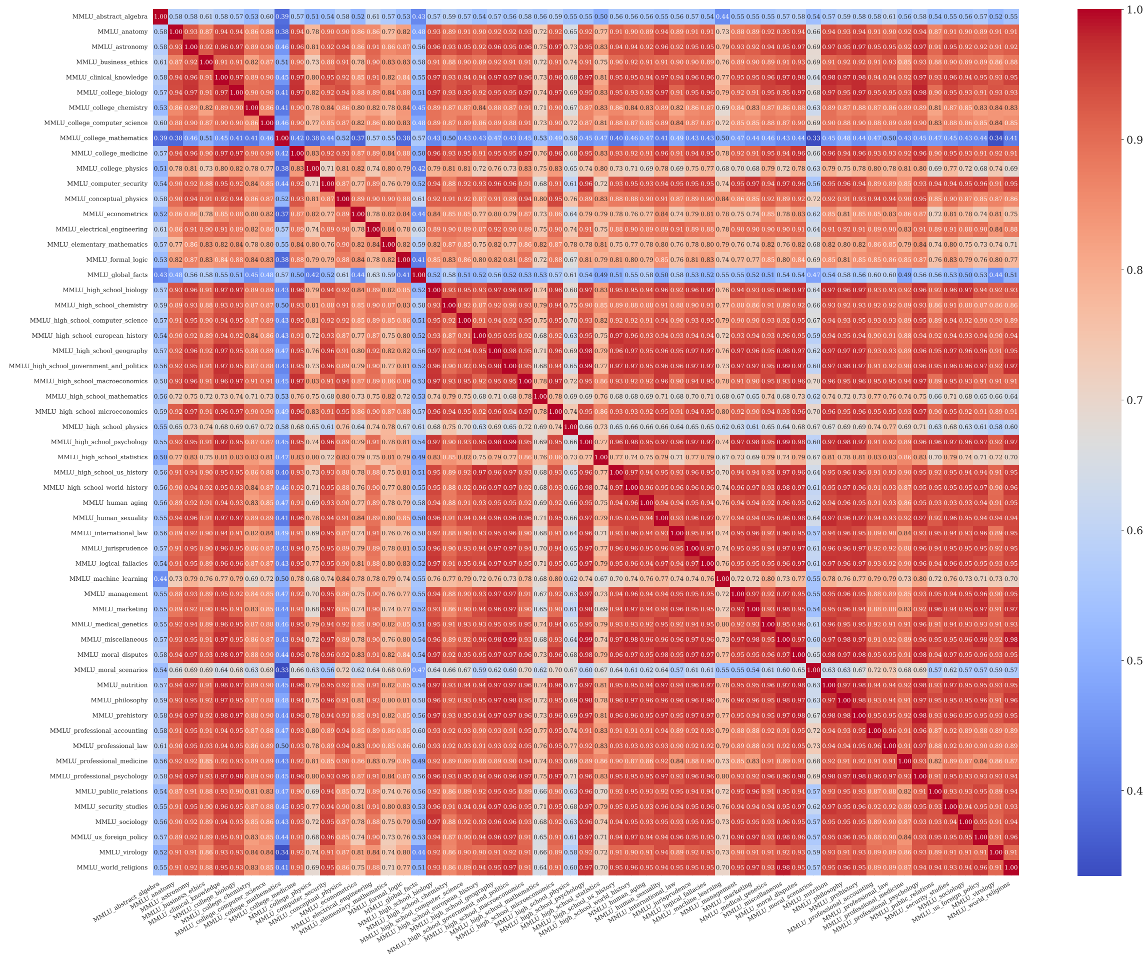
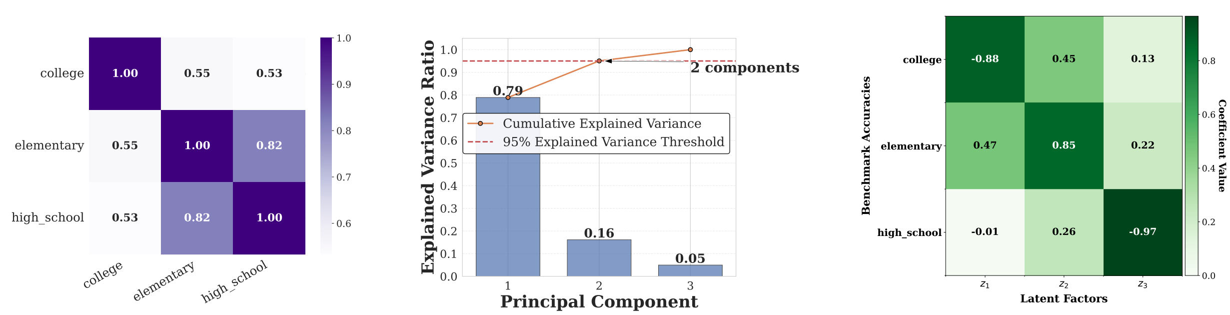


Figure 27: Correlation matrix for the tasks in the MMLU benchmark.



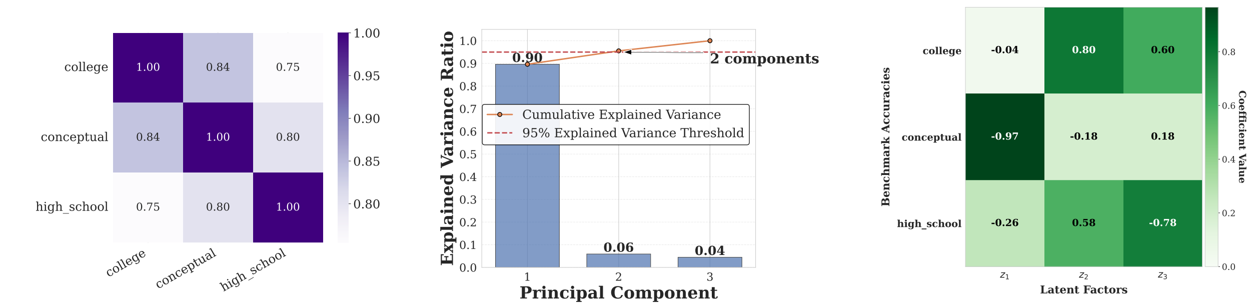
(a) Correlations of the performance of math-related subtasks.

(b) Low-rank structure of the subtasks performance data.

(c) The learned mixing matrix.

Figure 28: HCA analysis of the MMLU by Task Leaderboard data of math-related subjects.

Physics-related subjects. We conduct a similar analysis for Physics-related subjects. We choose the set of base models to be Mistral-7B, Mistral-8x7B, Llama-2-13B, Llama-2-70B, which induce a minimal MIC of 0.05 among all domain subsets with size ≥ 4. The result of HCA is summarized in Figure 29c. We can see that z_1 is conceptual physics while z_2 and z_3 are both linear cominations of high-school and college physics.



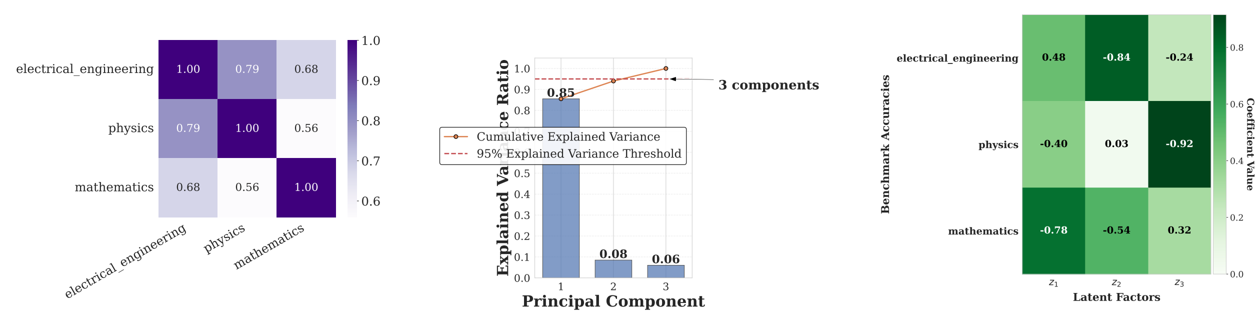
(a) Correlations of the performance of physics-related subtasks.

(b) Low-rank structure of the subtasks performance data.

(c) The learned mixing matrix.

Figure 29: HCA analysis of the MMLU by Task Leaderboard data of physics-related subjects.

Cross-subject domains. It would also be interesting to explore how different subjects are related. We choose MMLU_college_mathematics, MMLU_college_physics and MMLU_college_electrical_engineering and run HCA on these subjects. We found the a hierarchical relationship exists in the order of math, electrical engineering and physics.



(a) Correlations of the performance of tasks under different subjects.

(b) Low-rank structure of the subtasks performance data.

(c) The learned mixing matrix.

Figure 30: HCA analysis of the MMLU by Task Leaderboard data of three different subjects

I. Discussions

For post-training evaluation: Our results demonstrate that the impact of any fine-tuning intervention can differ substantially across base models. Evaluation studies therefore are expected to specify exactly which pre-trained checkpoints their methodology applies to. To quantify these heterogeneous effects, one can employ standard causal-inference tools – such as estimating the conditional average treatment effect (CATE) – to measure, with statistical rigor, how fine-tuning impacts performance on each base models.

For model developers: The directed, hierarchical structure of capabilities we uncover suggests a clear development priority: given sufficient compute budget, one can focus on scaling up pre-training FLOPs which are more correlated with upstream parent node z_1 performance and can, and gains there cascade to more specialized abilities. That said, not every capability is equally malleable. Some – like instruction-following – correlate less with model scale (i.e., FLOPs) and exhibit large noise-factor variances (the z_2 node in Figure 4), indicating they respond more readily to post-training. On the other hand, given limited budgets or for small models, our noise-weight estimates suggest that we may need other interventions like instruction tuning to further improve downstream performance.

For model evaluators: Due to the inherent hierarchical structure of evaluation suites, it is important to examine fine-grained performance beyond aggregate numerical scores. For example, gains on the MATH benchmark may partly stem from improved instruction-following, which, while related to, is not equivalent to the mathematical reasoning the benchmark aims to evaluate. Secondly, as specialized abilities are causally affected by upstream ones, evaluators can therefore prioritize designing benchmarks that evaluate general, foundational capabilities, such as BBH and MMLU-Pro. These benchmarks reflect more substantive improvements rather than artifacts of limited domain adaptation.

We leave as future work the use of a tapestry of tools in causal inference, such as matching (Stu10), stratification (FR02), doubly robust estimation (BR05), etc, to derive more scientific insights from observational language model benchmark data.