

Discovering Hierarchical Latent Capabilities of Language Models via Causal Representation Learning

Jikai Jin

Stanford University
jkjin@stanford.edu

Vasilis Syrgkanis

Stanford University
vsyrgk@stanford.edu

Sham M. Kakade

Harvard University
sham@seas.harvard.edu

Hanlin Zhang

Harvard University
hanlinzhang@g.harvard.edu

Abstract

Public LLM leaderboards contain rich observational evidence about post-training, but this evidence is strongly confounded by base-model family. We study *capability transfer*: whether improving one capability tends to improve another, and whether that carryover differs across related families. Our goal is to study this question mainly from cheap observational evaluation data rather than from large collections of expensive training runs. We propose Hierarchical Component Analysis (HCA), a family-aware latent-factor method for studying capability transfer from observational data alone. Applied to Llama and Qwen families containing 1500+ models from the Open LLM Leaderboard, HCA identifies a three-factor structure whose most stable pattern is a positive link from an instruction-aligned factor to a math-aligned factor. We then verify this finding via real fine-tuning with external dataset, showing that it is not merely a benchmark artifact. Overall, this work shows that observational data can already suggest concrete post-training predictions.

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 (Brown et al., 2020; Achiam et al., 2023; Grattafiori et al., 2024; Anthropic, 2024; Abdin et al., 2024; Yang et al., 2024; Guo et al., 2025). 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) (Ouyang et al., 2022; Bai et al., 2022) to enable downstream model deployment. These complicated system engineering make it hard to evaluate how models acquire capabilities and derive scientific claims thereafter.

In this paper, we study *capability transfer*, a common intuition among practitioners: capabilities are not independent targets for fine-tuning. Improving a foundational skill might be a prerequisite for, and causally influence, a more specialized skill.

Despite being intuitive, rigorous quantification of such transfer has two concrete obstacles. First, pre-training heterogeneity is substantial: implementation details such as data mixture, architecture, and inherited base-model representations vary widely across institutions and are often proprietary. Consequently, the benefits of post-training on reasoning abilities can differ substantially even between models of comparable size (Gandhi et al., 2025; Zhao et al., 2025; Hochlehnert et al., 2025). Even with transparent pre-training recipes, controlling these factors through from-scratch studies is prohibitively expensive (Cottier et al., 2024; Qi et al., 2025). Second, in many training recipes, various capabilities often seem to co-emerge as

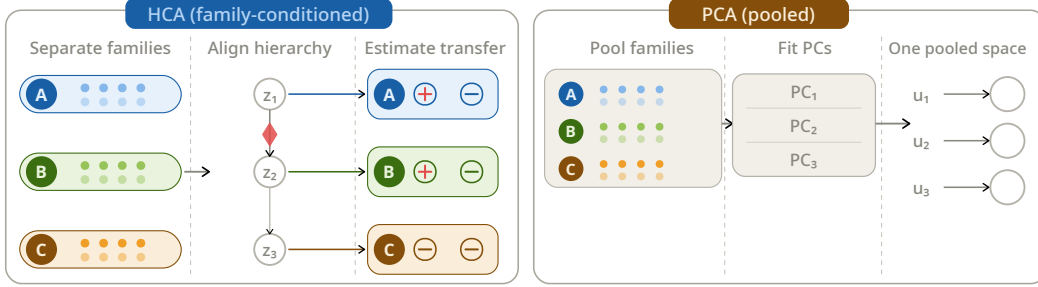


Figure 1: PCA does not preserve the family-specific structure needed for understanding heterogeneous capability transfer. HCA instead fits family-specific latent spaces and aligns them under a shared ordering, allowing both the direction and the size of capability transfer to vary by family.

model scale increases (Ouyang et al., 2022; Zhang et al., 2025; Chen et al., 2025; Liu et al., 2025), systematic understanding of their *interactions* is still lacking.

Existing factor-modelling approaches (Ruan et al., 2024; Truong et al., 2025) leverage cheap, observational evaluation data to discover latent capabilities of LMs. Their main finding is that model’s performances across diverse benchmarks are governed by a few latent factors. However, these approaches offer limited insights into whether and how one capability affects another, primarily because of two reasons:

- (i) they typically pool all models together and therefore average away the heterogeneity of the post-training behaviors across different model families. We will discuss this issue in details in Section 2, where we will see that a simple PCA yields distinct latent capability structures across model families.
- (ii) correlation alone is also insufficient for the intriguing empirical finding (as we show in Section 4.5, Fig. 9, and Table 4) that the transfer is often *directional*: fine-tuning on A improving B does not imply that fine-tuning on B will improve A.

In this paper, we propose **Hierarchical Component Analysis (HCA)**, a method that aligns low-dimensional latent capabilities across related base-model families and estimates how much improvement in one capability carries over to another in each family. In our setting, this gives us a structured way to study capability transfer using only evaluation data while preserving family heterogeneity. As in prior observational factor-analysis work, the input is only evaluation data; what changes is that HCA preserves family heterogeneity rather than pooling it away. A high-level comparison of HCA and PCA is illustrated in Figure 1. To the best of our knowledge, we are among the first to directly produce concrete predictions about post-training transfer from observational benchmark data.

We will introduce the HCA algorithm in Section 3.1. We then move on to address the following two key questions:

- (i) after conditioning on related base-model families, can multi-benchmark evaluation data support a useful approximate hierarchical latent-factor fit, and
- (ii) if so, does the most stable fitted downstream relation help predict the direction and strength of post-training transfer in clean interventions?

We address the first question through PCA screening, error-based subset selection, and fit diagnostics in Figures 2 and 6 and Table 2. Figure 5 summarizes the procedure. This process selects four related base-model families—Qwen2.5-7B, Qwen2.5-14B, Llama-3-8B, and Llama-3.1-8B—for which HCA fits the benchmark data well. Within the selected family cluster, the most stable recovered pattern is that an instruction-aligned target consistently points toward a math-aligned target (Figure 7 and Section 4.3). We then test whether this

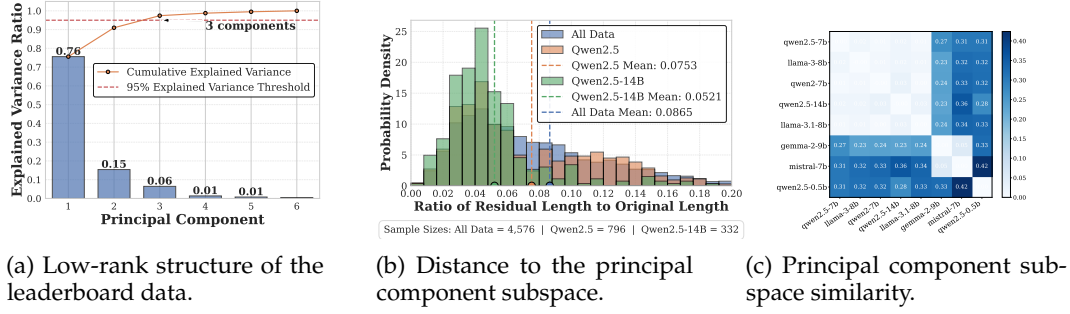


Figure 2: PCA diagnostics for the leaderboard data. Domain-specific PCA reveals heterogeneity across base-model families.

observational HCA pattern predicts real post-training behavior using external datasets that are not used as inputs to HCA (Section 4.5, Fig. 9, and Tables 4 and 6).

1.1 Notation

Most of our analysis uses the Open LLM Leaderboard, which reports scores for $N_0 = 4576$ models on $d = 6$ benchmarks. Because the leaderboard’s base-model field often names an intermediate tuned checkpoint rather than the original pretrained model,¹ we infer base-model family from the remaining metadata. This yields a subset of $N = 3360$ models with confident base-model assignments, denoted by Θ . The eight most common families are Llama-3-8B, Llama-3.1-8B (Grattafiori et al., 2024), Qwen2.5-0.5B/7B/14B and Qwen2-7B (Yang et al., 2024), Mistral-7B (Jiang et al., 2023), and Gemma-2-9B (Team et al., 2024). Some later analyses also include additional families, which we introduce where needed. We denote these eight base-model families by $\theta_1^*, \theta_2^*, \dots, \theta_8^*$ and let $\Theta_k = \{\theta_{1,k}, \dots, \theta_{N_k,k}\} \subseteq \Theta$ be the set of models using θ_k^* as base model.

For any LM θ and benchmark B, we use $x_{\theta,B}$ to denote the accuracy of θ on B, if observed. In our setting, we observe $x_{\theta_{i,k},B_j}$ for all $k \in [K], i \in [N_k]$ and $j \in [d]$, and we will simply denote this by $x_j^{(k,i)}$. Then $\mathbf{x}^{(k,i)} = (x_j^{(k,i)})_{j=1}^d$ is the observed performance vector for model $\theta_{i,k}$. The set of all data, $\{\mathbf{x}^{(k,i)} : k \in [K], i \in [N_k]\}$, is denoted by X . We also define $X^{(k)} = \{\mathbf{x}^{(k,i)} : i \in [N_k]\}$. In what follows, we will sometimes abuse notation and view $X' \subseteq X$ as a $|X'| \times d$ matrix, where each row is one performance vector $\mathbf{x} \in X'$.

2 From benchmark heterogeneity to a domain-conditioned latent-factor model

We begin with pooled benchmark geometry and then show why it is insufficient for the transfer question we study. Pooled analyses suggest an approximate low-rank structure, but they also mask substantial differences across base-model families.

Figure 2 provides three observations. First, the full leaderboard matrix is approximately rank-3, consistent with prior observational factor analyses (Ruan et al., 2024; Ren et al., 2025; Polo et al., 2024). Second, distances to the rank-3 subspace vary materially across families, so a one-size-fits-all factor analysis can blur domain-specific structure. Third, among the eight common base-model families in Section 1.1, five domains—Llama-3-8B, Llama-3.1-8B, Qwen2-7B, Qwen2.5-7B, and Qwen2.5-14B—have closely aligned rank-3 PC subspaces. We denote this candidate set by S_{cand} .

¹There is a column named ‘Base model’ in the leaderboard, but it often refers to an intermediate tuned checkpoint rather than the original pretrained model.

FINDING 1: FAMILY HETEROGENEITY

Base model family materially shapes benchmark geometry, and only a subset of families supports a shared low-rank structure.

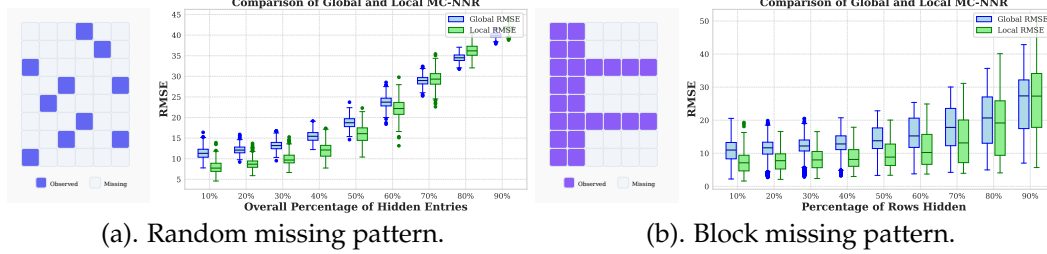


Figure 3: RMSE of global and local matrix completion approaches.

Remark 1. As a side application, the same family-conditioning idea also improves missing-score imputation on the leaderboard. We focus on two distinct missing patterns, illustrated in Figure 3, and study the missing performances of models fine-tuned on Qwen2.5-14B. Motivated by the heterogeneity seen in the PC subspaces, we apply matrix completion with nuclear norm regularization (MC-NNR) both within the Qwen2.5-14B domain and across the entire leaderboard. The local approach yields notably lower reconstruction error. Additional details are provided in Figure 27.

3 Hierarchical Component Analysis

We now introduce HCA as a cross-family latent-factor method. The operational goal is simple: starting from benchmark vectors in several related base-model families, we look for a small set of latent capabilities whose linear combinations explain benchmark scores within each family, align those capabilities across families, and then estimate how strongly movement along one aligned capability tends to propagate to a later one in the fitted ordering. In this way, HCA treats family heterogeneity not only as a confounder to control away, but also as the variation that lets us compare transfer effects across related families. Section 3.1 gives this high-level description, while Section 3.2 explains how it relates to an idealized linear-SCM view in the exact case. The full algorithm is presented in Appendix C.

3.1 Algorithm Description

In our context, benchmark scores are treated as noisy linear mixtures of a few latent factors that later line up with the benchmarks. These factors are aligned only across related base-model families rather than across all LMs. Within each family, later factors can depend on earlier ones, and the fitted coefficients measure how strong that dependence is. Comparing these fitted weights across families then shows how much the carryover differs by family when the benchmark patterns are otherwise similar. With the notations introduced in Section 1.1, HCA has three steps:

- 1. Recover family-wise latent capabilities (Figure 4 (a)).** For each domain k , we apply ICA to the observed benchmark vectors x and obtain a family-wise unmixing matrix M_k , whose rows define independent source directions. Intuitively, these source directions are the ingredients from which the aligned latent capabilities are later reconstructed.
- 2. Align latent capabilities across families (Figure 4 (b) and (d)).** Across domains, HCA searches for row orderings of $M_k, k \in [K]$ under which, after removing what earlier factors already explain, the remaining rows still point in the same direction across families. Intuitively, we remove the part of each row that can already be explained by earlier aligned capabilities and then ask whether the remaining projection residual points in a common direction across families. This row-residual step is designed to isolate what is not yet explained by the already-aligned earlier capabilities; when the residual still points in a shared direction across families, it supplies evidence for one additional aligned capability

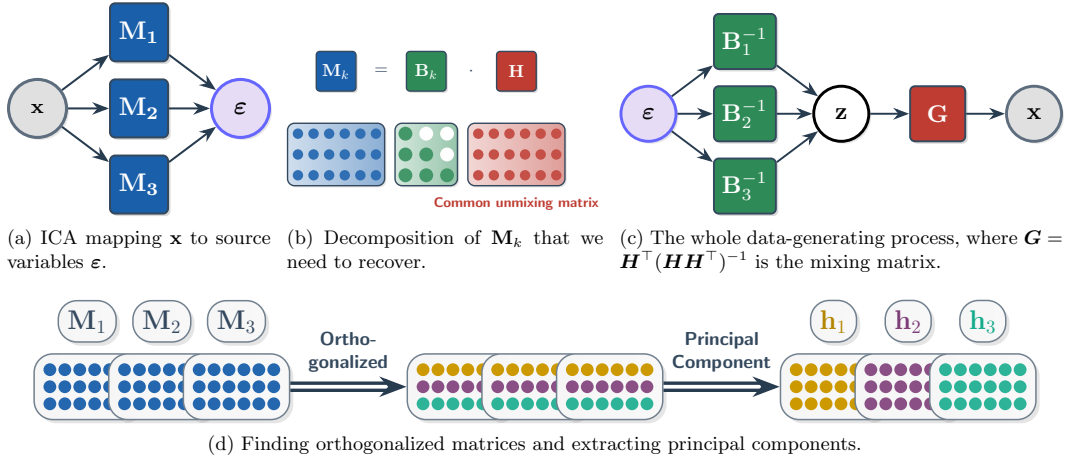


Figure 2: Illustration of our setting and the key row-residual extraction step in our algorithm.

Figure 4: Illustration of our setting and the key row-residual extraction step in our algorithm.

rather than an arbitrary rotation. When this happens, the leading singular vector of the stacked residual recovers one new capability that is directly built on top of the already-aligned ones. Algebraically, this means that we are searching for row-permuted M_k , denoted by M'_k , that can approximately be decomposed as $B_k H$, where B_k is lower-triangular and H is shared across all $k \in [K]$.

3. Estimate family-specific transfer effects. In the finite-sample regime, we find it helpful to re-compute the best fitted B_k after H is obtained. Namely we minimize $B_k \mapsto \|M'_k - B_k H\|_F^2$ over all lower-triangular matrices. The entries of B_k summarize how strongly one capability depends on earlier ones; Section 3.2 and Fig. 7 makes this interpretation concrete. A larger coefficient indicates a stronger transfer link.

3.2 Theoretical foundation

This subsection explains the exact linear-SCM idealization behind the procedure above. The connection to causal representation learning (Jin & Syrgkanis, 2024; Zhang et al., 2024) provides one principled way to understand why cross-family alignment and family-specific transfer effects are well-defined quantities in the exact model. In the main text, the empirical evidence comes from approximate fit, MIC, and stability across domain choices (Figure 6, Table 2, and Section 4.3). In that setting, one assumes K domains $\mathfrak{E} = \{\mathcal{E}_k : k \in [K]\}$ and a dataset $\mathbf{X}^{(k)} = \{\mathbf{x}^{(k,i)}\}_{i=1}^{N_k}$ associated with the k -th domain generated from the structural equations

$$\mathbf{z}^{(k,i)} = A_k \mathbf{z}^{(k,i)} + \Omega_k^{1/2} \epsilon^{(k,i)}, \quad \mathbf{x}^{(k,i)} = G \mathbf{z}^{(k,i)}, \quad k \in [K], \quad (1)$$

where $(A_k)_{ij} = w_{ij}^{(k)}$ if there exists a direct causal edge $z_j \rightarrow z_i$ in the latent graph $\mathcal{G}$ and otherwise it is zero, Ω_k is a diagonal matrix encoding the variances of source variables. G is the shared mixing matrix. For convenience, we assume that the nodes of $\mathcal{G}$ are sorted in a topological order, i.e., $z_j \rightarrow z_i$ implies $j < i$. CRL seeks to uncover both the causal graph $\mathcal{G}$ and the mixing map G . Table 1 summarizes the parallels and distinctions between this CRL framework and our LM capability model.

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

Causal representation learning	Our context: learning latent LM capabilities
Domain set $\mathfrak{E}$	Each domain $\mathcal{E}_k \in \mathfrak{E}$ is indexed by a base-model family θ_k^* and the observed dataset $\mathbf{X}^{(k)} = \{\mathbf{x}^{(k,i)}\}_{i=1}^{N_k}$ that contains the performance of all LM $\{\theta_{i,k}\}_{i=1}^{N_k}$ that use θ_k^* as base model.
Observed $\mathbf{X}^{(k)} = \{\mathbf{x}^{(k,i)}\}_{i=1}^{N_k}, k \in [K]$	$\mathbf{x}^{(k,i)} \in \mathbb{R}^d$ contains the known benchmark accuracies of $\theta_{i,k}$.
Causal factors $\mathbf{Z}^{(k)} = \{\mathbf{z}^{(k,i)}\}_{i=1}^{N_k}, k \in [K]$	$\mathbf{z}^{(k,i)}$ is the unobserved d_0 -dimensional capability vector of $\theta_{i,k}$ that possesses some causal structure. We assume that $d_0 \leq d$.
Mixing matrix $\mathbf{G}$ (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.

Section C.2 shows that under an exact SCM, HCA recovers the unmixing matrix $\mathbf{H} = (\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top$ up to left multiplication by a lower-triangular matrix. Equivalently, each identified latent factor for z_i is a mixture of $z_j, 1 \leq j \leq i$. Following [Jin & Syrgkanis \(2024\)](#), this lower-triangular ambiguity is unavoidable: multiple parameterizations can induce the same observed distribution.

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 $z_1, z_2, \dots, z_{d_0}$ with $\hat{\mathbf{e}} = \mathbf{U}\boldsymbol{\epsilon}, z_i = \sum_{j \in \text{pa}_{\mathcal{G}}(i)} w_{ji} z_j + \sigma_i^{1/2} \hat{\mathbf{e}}_i, i \in [d_0]$ for some independent source variables ϵ_i with unit variance and some matrix $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{d_0}]^\top \in \mathbb{R}^{d_0 \times d_0}$ with $\|\mathbf{u}_i\|_2 = 1$ and $\frac{1}{d_0} \sum_{i \neq j} u_{ij}^2 \leq \alpha$. 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 the SCM is inexact, HCA recovers a data generating process

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

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

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

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. Our empirical use of HCA relies instead on approximate fit diagnostics, especially MIC and stability across domain choices. In the results, MIC is used both to select the primary analysis set ([Figure 6](#)) and to define the low-MIC subsets whose stability we compare in [Section 4.3](#).

4 Empirical Results

In this section, we demonstrate the results of HCA when applied to the evaluation data from the Open LLM Leaderboard, discuss their implications and compare the fitted transfer pattern with real fine-tuning outcomes.

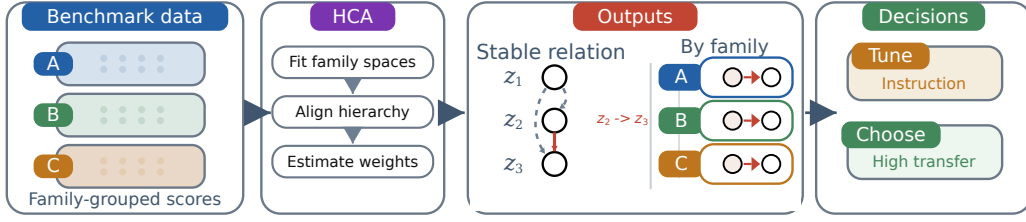


Figure 5: HCA takes multi-family benchmark results, fits a shared latent space across related base-model families, and estimates how strongly one factor depends on another in each family. The main robust output is a stable downstream $z_2 \rightarrow z_3$ relation together with family differences in the fitted transfer pattern, which suggest different post-training choices across families.

Primary fit	MIC	Rank-1 residuals (z_1, z_2, z_3)	Unmixing error range
S^*	0.04	(0.05, 0.13, 0.02)	0.16–0.23

Table 2: Compact fit diagnostics for the primary analysis set. Rank-1 residuals summarize the row-extraction step in HCA, and unmixing error reports $\|\mathbf{M}_k - \mathbf{B}_k \mathbf{H}\|_F / \|\mathbf{M}_k\|_F$ across the four domains.

4.1 Selecting the primary analysis set

We first apply HCA to candidate subsets screened by PCA. Among the five-family candidate pool $S_{\text{cand}} = \{\text{Llama-3-8B}, \text{Llama-3.1-8B}, \text{Qwen2-7B}, \text{Qwen2.5-7B}, \text{Qwen2.5-14B}\}$, the lowest-MIC fit is achieved by the four-family subset $S^* = \{\text{Llama-3-8B}, \text{Llama-3.1-8B}, \text{Qwen2.5-7B}, \text{Qwen2.5-14B}\}$. All lowest-MIC subsets exclude Qwen2-7B. We therefore use S^* as the primary analysis set. This exclusion highlights that even closely related families need not share the same hierarchical fit.

FINDING 2: FAMILY-CONDITIONAL FIT

Among the PCA-screened candidates, the best hierarchical fit is family-conditional: every lowest-MIC subset excludes Qwen2-7B.

4.2 Approximate hierarchical fit on the primary analysis set

On the primary analysis set S^* selected in Figure 6, HCA with $d_0 = 3$ yields the lowest MIC among the screened candidates. MIC is used to compare candidate fits for the approximate hierarchical model.

In these per-family sketches, ε_i denote source variables and z_i denote the fitted latent capabilities.

Across the primary analysis set, the strongest fitted structure is the positive $z_2 \rightarrow z_3$ coefficient in Figure 7, which appears in all four domains of S^* . By contrast, the z_1 -related coefficients are less stable once the domain choice is varied (Section 4.3), so the main structural evidence lies in the downstream z_2/z_3 pair. Additional raw factorization views and fit diagnostics are reported in Figure 15 and Table 5.

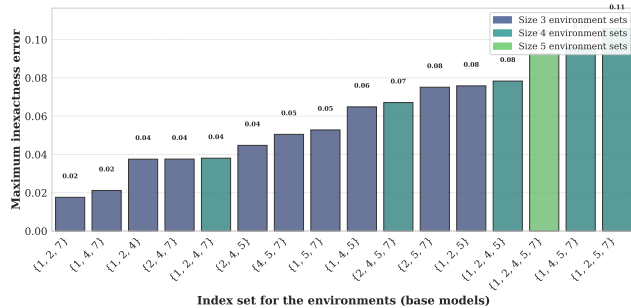


Figure 6: MIC across candidate subsets screened by PCA. The four-family subset S^* attains the lowest MIC among the screened candidates.

4.3 Stability across domain choices

We check robustness by asking whether the recovered structure survives changes to the domain set. Across low-MIC subsets, three features recur: after the post-hoc benchmark alignment shown in Figure 8, z_2 remains most strongly aligned with instruction-following behavior, z_3 remains most strongly aligned with mathematical reasoning, and the fitted $z_2 \rightarrow z_3$ coefficient remains positive. By contrast, the interpretation of z_1 is materially less stable and can rotate toward a broader tradeoff between general reasoning and more specialized benchmarks.

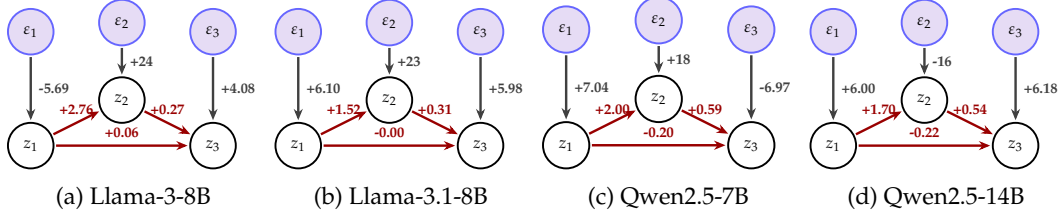


Figure 7: Estimated weights A_k for the four domains in S^* . The most stable qualitative feature is the positive fitted $z_2 \rightarrow z_3$ coefficient across all four families.

The same downstream pattern persists when two additional related domains are introduced (Figure 14). This sensitivity check again supports the relation from the instruction-aligned factor to the math-aligned factor within low-MIC related families, while the upstream factor changes more under alternative domain choices. Accordingly, our robust claim concerns the $z_2 \rightarrow z_3$ relation; the full three-factor interpretation is more tentative.

4.4 Benchmark alignment and scope of interpretation

Because the exact model is identifiable only up to lower-triangular ambiguity, after fitting the latent factors, we rotate them within the allowed triangular family so they line up more clearly with the benchmarks. Concretely, for each latent factor z_i , we regress it on upstream factors and one observed benchmark, then select the benchmark that yields the strongest alignment and subtract upstream mixtures accordingly. This step chooses an interpretable coordinate system within the equivalent lower-triangular family.

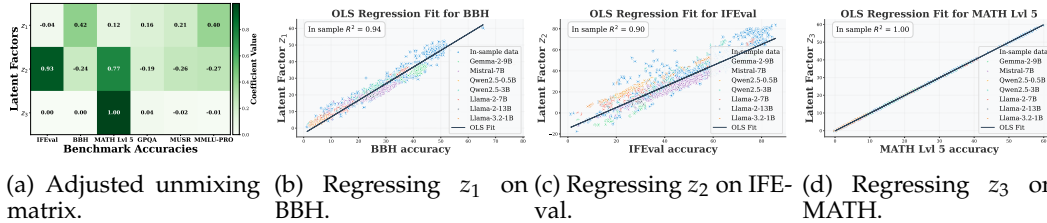


Figure 8: Post-hoc benchmark alignment of the recovered latent factors. The OLS step fixes an interpretable coordinate system within the equivalent lower-triangular family. Within the primary analysis set, the benchmark alignment is clearest for a broad earlier factor z_1 and a more stable downstream pair (z_2, z_3) (Figure 8). We interpret these factors within the selected low-MIC families. z_1 is a broad upstream factor with strongest alignment to BBH and MMLU-Pro, but it mixes multiple benchmarks and becomes less stable under alternative domain choices (Section 4.3), so it is best understood as a broad reasoning/knowledge factor. z_2 aligns most closely with IFEval and remains stable across low-MIC subsets, so we interpret it as an instruction-aligned factor. z_3 aligns most closely with MATH Lvl 5 and is the most semantically stable factor across our analyses.

These alignments only hold within the selected families. The fitted OLS extrapolates poorly to several out-of-family base models, especially for z_1 and z_2 : for example, on Qwen2.5-0.5B the out-of-family R^2 drops to -0.20 for z_1 and -1.18 for z_2 , while z_3 remains near-perfect at 0.99 (Table 3). This asymmetry is part of the main result: the aligned factors are informative within related families and extrapolate unevenly outside them.

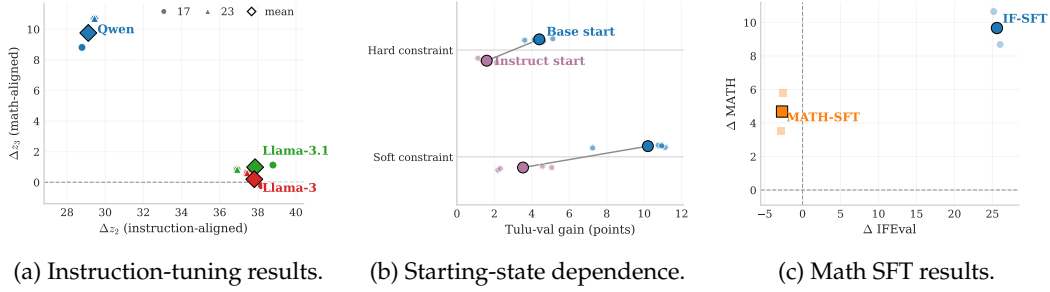


Figure 9: Results of SFT experiments that are aligned with the predictions of our observational analysis.

FINDING 3: STABLE DOWNSTREAM RELATION

Across well-fitted families, the stable recovered relation is a positive link from the instruction-aligned factor to the math-aligned factor.

4.5 Matched SFT interventions

The observational analysis identifies one stable downstream relation: within the selected family cluster, the instruction-aligned factor z_2 points toward the math-aligned factor z_3 (Figures 7 and 8 and Section 4.3). We test that directional prediction with matched post-training interventions built from data screened against the benchmark prompts.

We compare three matched intervention settings: forward instruction SFT from base checkpoints, the same recipe from already-instruct checkpoints, and a matched math-focused control. Across Qwen2.5-7B, Llama-3-8B, Llama-3.1-8B, forward TuluIF-Clean-v1 SFT strongly improves instruction following in raw benchmark performance (Table 6). In our framework, this means every completed base-family run moves sharply in the positive z_2 direction, but the downstream response on z_3 differs by family. Qwen shows a clear upward shift on the math-aligned axis, whereas both Llama families remain near zero on z_3 despite large instruction-side movement (Figure 9a and Table 6). This pattern suggests that the carryover differs by family in the fitted $z_2 \rightarrow z_3$ relation. The same TuluIF-Clean-v1 recipe improves instruction following from already-instruct checkpoints, but the gains are much smaller than from base checkpoints (Figure 9b and Table 7).

Remark 2. Notably, Figure 8 already predicts that Qwen-based models has a stronger transfer effect than Llama-based models, via the weights of the edge $z_2 \mapsto z_3$. Also note that these weights are generally higher than the actual performance transfer ratio we observe in training (Figure 9a and Table 6). A possible reason is that z_2 is by definition a mixture of instruction tuning and math problem solving, as shown in Figure 8a. However, this does not undermine the real pattern that HCA successfully predicts.

Is this transfer effect directional? We observe that math-focused SFT improves MATH-Lvl-5 but does not improve IFEval, and it does not improve instruction-following on Tulu as well, as shown in Figure 9c and Table 4.

FINDING 4: FAMILY-VARYING AND DIRECTIONAL TRANSFER

Instruction-tuning reliably improve math performance, but the effect is strong in Qwen and weak in the two Llama families. Conversely, math SFT brings no improvements on instruction following.

5 Scope, limitations, and conclusion

HCA demonstrates that observational benchmark data contains actionable signals about capability transfer. The core finding is that fine-tuning targets are fundamentally entangled:

instruction-side improvements directly drive mathematical reasoning in certain model lineages, while yielding marginal carryover in others.

For post-training evaluation. The same benchmark gain should not be interpreted in the same way across lineages. A gain on a downstream benchmark can reflect both direct improvement and carryover from an upstream capability, so evaluations should report base-model family and how those other benchmarks’ performances change, rather than a single aggregated delta.

For model development and benchmark design. As our analysis reveals, when math problem-solving is the goal, instruction tuning can be a useful indirect path in some families, but not in all of them. More generally, a recipe that works in one family should not be assumed to transfer unchanged to another. Furthermore, capability transfer implies that gains on a specific dataset rarely represent an isolated capability improvement.

Limits. HCA does not replace controlled post-training experiments, and it only yields meaningful conclusions for families where the latent-factor fit is stable. An important future direction is to find the correct causal modeling tool that are suitable for characterizing heterogeneous causal dependencies between capabilities, applicable to arbitrary model families.

References

- 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.
- Emre Acartürk, Burak Varıcı, Karthikeyan Shanmugam, and Ali Tajer. Sample complexity of interventional causal representation learning. *Advances in Neural Information Processing Systems*, 37:39350–39385, 2024.
- 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.
- Anthropic. Claude 3.5 sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>, 2024.
- 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.
- 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.
- Ryan Burnell, Wout Schellaert, John Burden, Tomer D Ullman, Fernando Martinez-Plumed, Joshua B Tenenbaum, Danaja Rutar, Lucy G Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, et al. Rethink reporting of evaluation results in ai. *Science*, 380(6641):136–138, 2023.
- 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.
- Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, and David Owen. The rising costs of training frontier ai models, 2024.

- Jan Eriksson and Visa Koivunen. Identifiability, separability, and uniqueness of linear ica models. *IEEE signal processing letters*, 11(7):601–604, 2004.
- 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.
- 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.
- 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.
- Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandaraao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *arXiv preprint arXiv:2504.07086*, 2025.
- 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.
- 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. URL <https://openreview.net/forum?id=dB99jjwx3h>.
- 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.
- Emmy Liu, Amanda Bertsch, Lintang Sutawika, Lindia Tjauatja, 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.
- 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.
- 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.
- Zhenting Qi, Fan Nie, Alexandre Alahi, James Zou, Himabindu Lakkaraju, Yilun Du, Eric Xing, Sham Kakade, and Hanlin Zhang. Evolm: In search of lost language model training dynamics, 2025.
- 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.
- Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=On5WIN7xyD>.

- 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.
- 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.
- Sang Truong, Yuheng Tu, Percy Liang, Bo Li, and Sanmi Koyejo. Reliable and efficient amortized model-based evaluation. *arXiv preprint arXiv:2503.13335*, 2025.
- Yixin Wang and Michael I. Jordan. Desiderata for representation learning: A causal perspective. *arXiv*, 2022. URL <https://arxiv.org/abs/2109.03795>.
- 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.
- 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. URL <https://openreview.net/forum?id=Pte6iiXvpf>.
- 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, pp. 25949–25957, 2025.
- 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.

LLM Usage

We used LLMs only for light writing assistance. No LLM-generated responses were used as training targets in the experiments reported in the main paper or in this appendix.

A Related Work

Observational benchmark geometry. Recent work shows that multi-benchmark LLM evaluation data often have strong low-rank structure and can support observational scaling or factor analyses (Ruan et al., 2024; Ren et al., 2025; Polo et al., 2024). Our paper builds on that observation, but departs from pooled analyses by conditioning explicitly on related base-model families rather than treating all leaderboard models as one homogeneous population.

Post-training transfer and confounding. A growing literature shows that post-training effects are heterogeneous across model families and difficult to interpret from benchmark gains alone because contamination, lineage differences, and evaluator choices can dominate apparent improvements (Burnell et al., 2023; Hochlehnert et al., 2025). Accordingly, the intervention section focuses on a clean, family-conditional test of a single edge-specific transfer hypothesis.

Causal representation learning. CRL studies when latent structure can be identified from heterogeneous environments (Schölkopf et al., 2021; Wang & Jordan, 2022; Jin & Syrgkanis, 2024; Zhang et al., 2024). We draw on this literature for idealized motivation and for the lower-triangular ambiguity story. In this paper, the empirical evidence comes from approximate fit, MIC, stability across domain choices, and a clean edge-specific intervention test.

B Reproducibility and Experimental Details

B.1 Leaderboard preprocessing and matched subset

We begin with the Open LLM Leaderboard v2 and retain models with complete scores on the six benchmarks used in the main paper. We then construct a matched subset with inferred base-model assignments and standardized metadata needed for the analysis within related families. The same matched subset underlies both PCA screening and HCA fitting.

B.2 Base-model matching and audit

We infer base-model family from model names, metadata, architecture tags, and parameter counts using a high-precision matching procedure. The matching rules use three levels of evidence: explicit base-model mentions, family-pattern matches with size disambiguation, and architecture-plus-parameter attribution when weight inheritance is sufficiently clear. When none of these rules gives a high-confidence match, we discard the model.

B.3 Candidate-family screening and primary-set selection

For each common base-model family, we run domainwise rank-3 PCA and compare principal-component subspaces. PCA is used only as a screening tool for families with similar low-rank benchmark patterns. HCA is then fit to candidate subsets, and the primary analysis set S^* is selected by comparative MIC. We first screen families with similar benchmark patterns and then fit HCA within that subset.

B.4 HCA implementation details

For each retained domain, HCA first applies ICA to estimate a domainwise unmixing matrix. It then searches for row orderings that make orthogonalized residual rows approximately rank-1 when stacked across domains, and finally fits the best lower-triangular factors in Frobenius norm to compute MIC. In the empirical sections, MIC serves as a fit statistic used to compare candidate fits.

B.5 Clean intervention protocol

The clean intervention evidence reported in the main paper combines completed TuluIF-Clean-v1 runs from three base-model families (Qwen2.5-7B, Meta-Llama-3-8B, and Llama-3.1-8B), start-state extensions from two already-instruct checkpoints (Qwen2.5-7B-Instruct and Meta-Llama-3-8B-Instruct), and a matched math-focused control dataset screened against benchmark prompts on Qwen2.5-7B. TuluIF-Clean-v1 is assembled from filtered instruction-following data plus deterministic synthetic augmentation under explicit overlap checks. We check exact prompt matches, normalized prompt matches, high-overlap 8-grams, and nearest-neighbor text similarity.

The final TuluIF-Clean-v1 split sizes are 12,000 training examples and 1,000 validation examples, with the training split combining 9,600 filtered Tulu rows and 2,400 deterministic synthetic rows, and the validation split combining 800 filtered Tulu rows and 200 synthetic rows. The synthetic augmentation uses eight controlled instruction families: `json_only`, `titled_paragraphs`, `required_and_forbidden_keywords`, `exact_end_phrase`, `repeat_prompt_then_answer`, `lowercase_no_comma`, `placeholder_thread`, and `section_separator_postscript`. Across the final audited 13,000 instruction rows, the exact-match, normalized-match, purity, and cross-split overlap checks all pass with zero unresolved collisions.

The common forward-training recipe, verified from the preserved Qwen and Meta-Llama configuration files, is QLoRA with rank 64, $\alpha = 128$, dropout 0.05, and target modules `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. The optimizer is `paged_adamw_8bit` with learning rate 2×10^{-4} , warmup ratio 0.03, cosine schedule, one epoch, maximum sequence length 2048, train/eval batch size 1/1, gradient accumulation

16, eval/save steps 100/100, early stopping patience 3, bf16=true, and sdpa attention. The exact local training YAML for the earlier Llama-3.1-8B runs is unavailable, but the evaluation manifests and experiment namespace place those runs under the same TuluIF-Clean-v1 protocol.

B.6 Evaluation and uncertainty

We evaluate the forward checkpoints against the same six-benchmark panel used in the observational analysis, together with held-out instruction-following evaluations and the filtered Tulu validation set. Section 4.5 combines six completed base-model forward runs, four instruct-start extensions, and one matched Qwen2.5-7B math-focused control. The Tulu-style validation re-evaluation uses a reduced nine-run comparison because one legacy Qwen2.5-7B tuned merged checkpoint is unavailable. The main reproducibility files used in this paper are the synced result pack under `outputs/result_packs/important_results.20260329_fullsync`, the Tulu validation re-evaluation under `outputs/day2_tulu3_recipe/reval_tuluif_on_tulu_val_available`, and the math-control files under `outputs/day2_asymmetry_math`. These files preserve the held-out clean instruction suite, the benchmark manifest hashes, the benchmark-to-latent mapping used in the analysis, the per-seed evaluation summaries, and the configuration files used to verify the common TuluIF-Clean-v1 recipe.

C Details of HCA and Its Theoretical Guarantee

C.1 HCA: Hierarchical Component Analysis

In [Jin & Syrgkanis \(2024\)](#), the authors introduced the LiNGCReL algorithm together with identifiability guarantees in [Theorem 1](#) for *exact* SCMs. Here we introduce HCA, which matches LiNGCReL in the exact setting but is adapted to our evaluation problem.

The first step, as in [Jin & Syrgkanis \(2024\)](#), is to apply linear ICA to each individual domain. Recall that ICA’s goal is to recover the independent signals; in our setting, it recovers the ICA unmixing matrix M_k that maps observed x to the source variables $\epsilon^{(k)}$ defined in [Equation \(1\)](#). This should be clearly distinguished from $H = (G^\top G)^{-1} G^\top$, which is the unmixing matrix for CRL. When the SCM is exact, we would have $P_k M_k = B_k H$, where P_k is some permutation matrix. The main challenge of CRL is that we only know $M_k = B_k H, k \in [K]$ but each 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 H is the same across all domains, the structure of any row spaces of $B_k, k \in [K]$, which are unknown, is captured by the row structures of the known ICA unmixing matrices $B_k H$. Moreover, given an already-recovered subgraph $\mathcal{G}_1$ of $\mathcal{G}$, one can discover some $v \notin \mathcal{G}_1$ such that $\text{pa}_{\mathcal{G}}(v) \subseteq \mathcal{G}_1$, if the corresponding rows in each B_k , after projecting onto the row spaces corresponding to 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 noting:

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). We omit this step here because the empirical model is inexact and the paper focuses on the lower-triangular fit. Equivalently, we are only imposing the constraint that each B_k is lower-triangular, without assuming that any other entries are also zero.
2. The identifiability guarantee of LiNGCReL makes the restrictive assumption that the distribution $\epsilon^{(k)}$ does not depend on k . This assumption can be relaxed, at the cost

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

of a more complicated matching step for the components of $\epsilon^{(k)}$. This step could be computationally expensive, but works well in our context where d is small.

3. We determine the matrices $B_k, k \in [K]$ by explicitly optimizing the distance between the recovered unmixing matrix and the target unmixing matrix. Compared with LiNGReL that sets the entries of 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($S, \{A_k\}_{k=1}^K$)

Input: Ordered set $S = \{s_1, s_2, \dots, s_m\} \subseteq [d]$, index $i \notin S, A_k \in \mathbb{R}^{d \times n}$ for $k \in [K]$.

Output: Residual matrices $\{R_k\}_{k=1}^K$.

```

for  $k \leftarrow 1$  to  $K$  do
     $W \leftarrow \text{span}\{(A_k)_s : s \in S\};$  //  $(A_k)_s$  is the  $s$ -th row of  $A_k$ 
     $R_k \leftarrow \text{proj}_{W^\perp}(A_k);$  // Row-wise orthogonal projection
end

```

Algorithm 2: Hierarchical component analysis

Input: Matrices $M_k \in \mathbb{R}^{d \times n}, k \in [K]$.

Output: The optimal unmixing matrix $\hat{H}^*$ and weight matrices $\{\hat{B}_k^*\}_{k=1}^K$.

Here $\text{TopRSV}(\cdot)$ denotes the top right singular vector.

Let S_d be the set of all permutations of $\{1, 2, \dots, d\}$. $\text{min_mic_score} \leftarrow \infty$;

$\hat{H}^* \leftarrow \text{null}; \{\hat{B}_k^*\}_{k=1}^K \leftarrow \text{null};$

```

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

```

C.2 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 complete DAG with $i \rightarrow j$ for all $i < j$. Equivalently, this means that each A_k is lower-triangular.

Assumption 1. (Node-level non-degeneracy, adapted from (Jin & Syrgkanis, 2024, Assumption 5)) We assume that the matrices $\{\mathbf{B}_k\}_{k=1}^K$ are node-level non-degenerate, i.e., for all node $i \in [d]$, we have $\dim \text{span} \langle (\mathbf{B}_k)_i : k \in [K] \rangle = |\text{pa}_G(i)| + 1$, where $(\mathbf{B}_k)_i$ is the i -th row of $\mathbf{B}_k$.

As shown in Jin & Syrgkanis (2024), 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 $\epsilon^{(k)}$ follows a different distribution, and all of them are non-Gaussian.

Remark 3. With a more involved procedure, Jin & Syrgkanis (2024) 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 $j \in \text{sur}_G(i) := \{i\} \cup \{i' \in \text{pa}_G(i) : \text{ch}_G(i) \subseteq \text{ch}_G(i')\}$. 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 $\mathbf{G}$ up to left multiplication by a lower-triangular matrix. Equivalently, it recovers latent factors $z'_1, \dots, z'_d$ 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 $\epsilon^{(k)} = \mathbf{B}_k \mathbf{H} \mathbf{x}$. Since we assume that the ICA is exact, the uniqueness of ICA in the non-Gaussian setting (Eriksson & Koivunen, 2004) implies that the unmixing matrix that leads to independent source variables must be unique up to row permutations. In other words, there exists some permutation matrix $\mathbf{P}_k^*$, such that $\mathbf{P}_k^* \mathbf{M}_k = \mathbf{B}_k \mathbf{H}, \forall k \in [K]$. Without loss of generality, we also assume that $\mathbf{H}$ is orthonormal, since otherwise one can always consider a QR factorization $\mathbf{H} = \mathbf{U} \tilde{\mathbf{H}}$ where $\mathbf{U}$ is lower-triangular and $\tilde{\mathbf{H}}$ is orthonormal, and one can treat $\mathbf{B}_k \mathbf{U}$ as the new $\mathbf{B}_k$.

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

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

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

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

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

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

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 $P_k, k \in [K]$ leads to a zero MIC, let $Q_k = P_k P_k^*$, then $M_k = Q_k B_k H$. We show that

1. $Q_1 = Q_2 = \dots = Q_K$, and
2. Suppose that the j -th row of Q_1 is e_{i_j} , then $i_1, i_2, \dots, i_d$ is a topological ordering of the graph $\mathcal{G}$, meaning that $\text{pa}_{\mathcal{G}}(i_j) \subseteq \{i_1, \dots, i_{j-1}\}$.

We say that a row index j is "good" if the j -th row of $Q_k, k \in [K]$ are equal and the second condition above is satisfied up to j (i.e. $i_1, \dots, i_j$ 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 $k \in [K]$,

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

Hence,

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

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

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

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

$$\text{rank} \left\langle \text{proj}_{V_{j-1}^\perp}(M_k)_j : k \in [K] \right\rangle = 1.$$

Let the j -th row of Q_k be $e_{j_k}, k \in [K]$. Then the above equation becomes

$$\text{rank} \left\langle \text{proj}_{V_{j-1}^\perp}(B_k H)_{j_k} : k \in [K] \right\rangle = 1. \quad (4)$$

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

$$\text{proj}_{V_{j-1}^\perp}(B_k H)_{j_k} = \sum_{i \in \bar{\text{pa}}_{\mathcal{G}}(j_k) \setminus \{i_1, \dots, i_{j-1}\}} (B_k)_{j_k, i} \text{proj}_{V_{j-1}^\perp}(h_i), \quad (5)$$

where $\bar{\text{pa}}_{\mathcal{G}}(i) = \text{pa}_{\mathcal{G}}(i) \cup \{i\}$. For $k \neq l$, equation 4 implies that $\text{proj}_{V_{j-1}^\perp}(B_k H)_{j_k}$ and $\text{proj}_{V_{j-1}^\perp}(B_l H)_{j_l}$ are colinear, but since H has full row rank, $\text{proj}_{V_{j-1}^\perp}(h_i), i \in [d] \setminus \{i_1, \dots, i_{j-1}\}$ are independent, so we must have $\bar{\text{pa}}_{\mathcal{G}}(j_k) \setminus \{i_1, \dots, i_{j-1}\} = \bar{\text{pa}}_{\mathcal{G}}(j_l) \setminus \{i_1, \dots, i_{j-1}\}$. In particular, $j_k \in \text{pa}_{\mathcal{G}}(j_l)$ and $j_l \in \text{pa}_{\mathcal{G}}(j_k)$, so we must have $j_k = j_l$. Thus $j_1 = j_2 = \dots = j_K$.

By Assumption 2,

$$\text{rank} \langle (B_k H)_{j_1} : k \in [K] \rangle = |\text{pa}_{\mathcal{G}}(j_1)| + 1,$$

so that

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

Hence it must be the case that $\text{pa}_{\mathcal{G}}(j_1) \subseteq \{i_1, \dots, i_{j-1}\}$, concluding the proof.

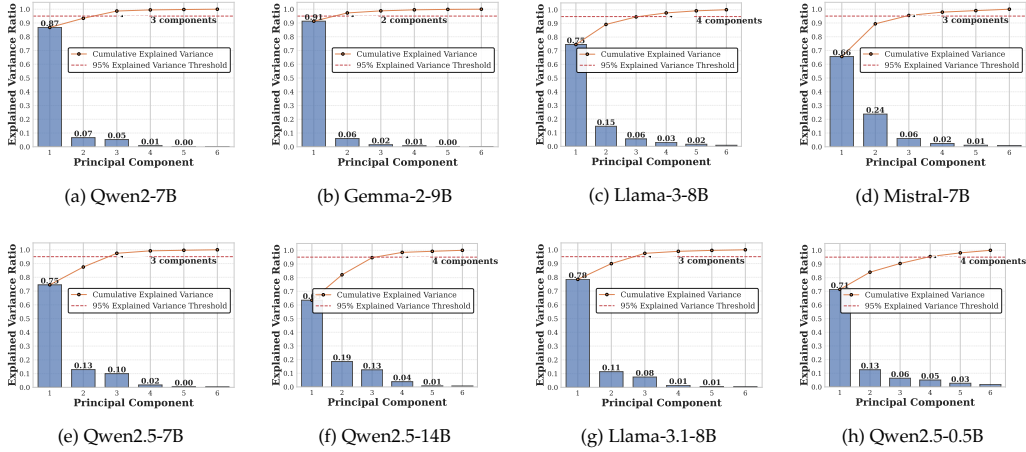


Figure 10: PCA on individual base-model families.

D Additional PCA and Family-Selection Diagnostics

In this section, we report the PCA diagnostics that motivate family selection before HCA fitting.

Domain-specific PCA. While Figure 2a indicates that the pooled leaderboard is approximately rank-3, this does not imply the same structure within each family. We therefore run PCA separately on the most common base-model families. As shown in Figure 10, the benchmark data in each selected family are themselves close to rank-3, with Gemma-2-9B appearing closer to rank-2.

Subspace similarity. Figure 11 extends the main-paper PCA similarity view to the 20 most common base models. The main candidate pool appears as a relatively coherent cluster inside this larger matrix.

Alternative domain definitions. We also compare principal-component subspaces under alternative groupings such as MoE versus non-MoE, coarse (N, D) bins, and upload-time buckets. These views are useful as diagnostics, but they are not used for the main HCA fit.

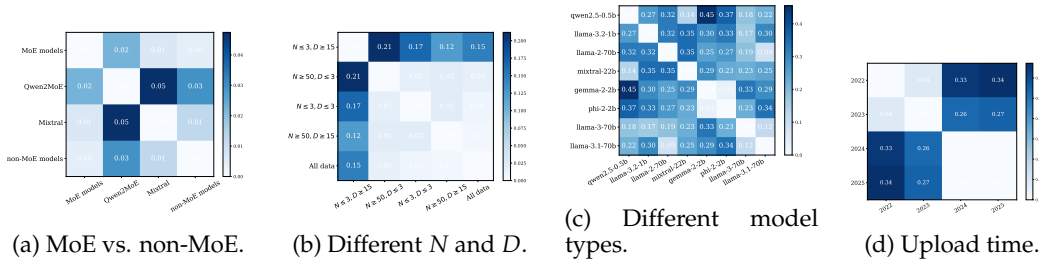


Figure 12: Additional PCA diagnostics for alternative domain definitions.

E Additional Supporting Results

E.1 Primary-set diagnostics

All lowest-MIC subsets among the PCA-screened candidate families exclude Qwen2-7B. Figure 13 therefore supports the main-text choice of S^* as the primary analysis set.

We also report the benchmark-alignment R^2 values used to interpret the latent coordinates together with their out-of-family behavior. The key asymmetry is the same one emphasized

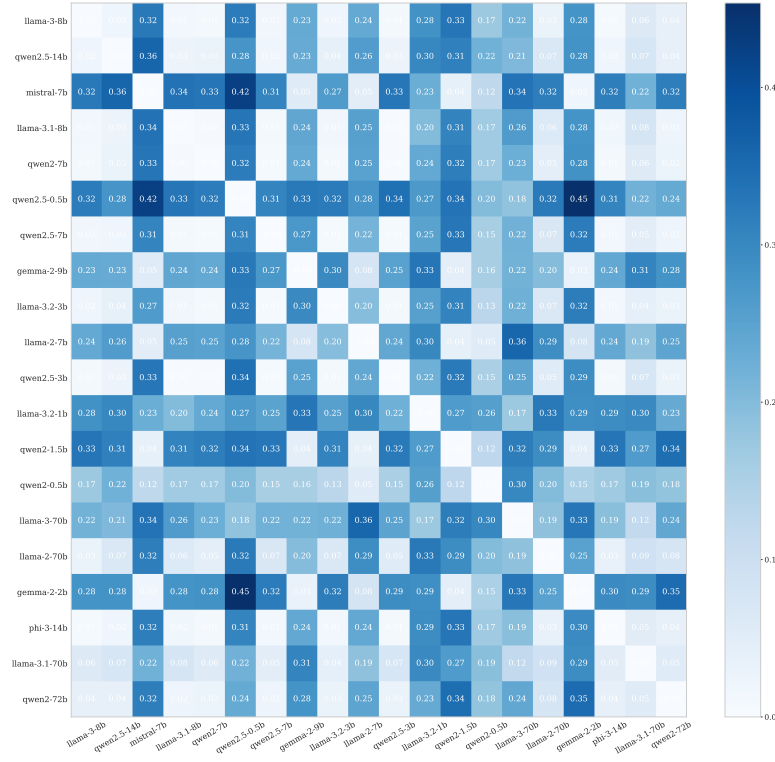


Figure 11: Pairwise cosine distance matrix for 20 common base models.

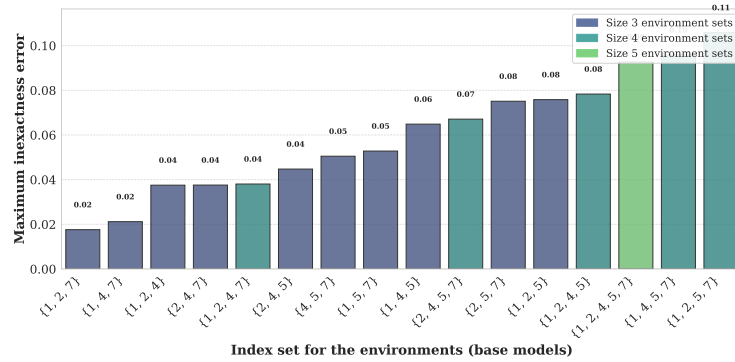


Figure 13: MIC across PCA-screened candidate subsets. The lowest-MIC subsets all exclude Qwen2-7B.

in the main paper: z_3 extrapolates much better than z_1 and z_2 once we move outside the selected family cluster.

	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) In-sample R^2 for aligning each latent factor to benchmarks after lower-triangular adjustment.

	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	-0.20	0.89	0.97	0.66	-0.01
z_2	0.92	0.94	0.74	-1.18	0.73	0.98	0.45	0.90
z_3	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00

(b) Out-of-family R^2 for the same OLS alignment.

Table 3: Benchmark alignment is strong in-sample, but out-of-family extrapolation is much weaker for z_1 and z_2 than for z_3 .

E.2 Matched math-focused control on Qwen2.5-7B

The matched math-focused control reuses the Qwen2.5-7B base model and a matched QLoRA recipe but replaces the instruction-following intervention data with a math-focused dataset screened against benchmark prompts. As discussed in the main paper, this control improves MATH but does not improve the clean instruction-following metrics used for the asymmetry check.

Seed	Δ IFEval	Δ MATH Lvl 5	Δ hard constraint acc.	Δ soft constraint acc.
17	-2.52	+5.82	+0.00	-0.75
23	-2.88	+3.55	+0.12	-1.88

Table 4: Matched Qwen2.5-7B math-focused control. The control improves MATH, but it does not improve IFEval or Tulu-style validation constraint accuracy. Hard-constraint accuracy is essentially unchanged, while soft-constraint accuracy decreases.

Node	z_1	z_2	z_3
Rank-1 error	0.05	0.13	0.02

(a) Variation in the row-extraction residuals not captured by a rank-1 approximation.

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) Relative unmixing error $\|\mathbf{M}_k - \mathbf{B}_k \mathbf{H}\|_F / \|\mathbf{M}_k\|_F$ by domain.

Table 5: Additional fit metrics for the primary analysis set.

F Sensitivity Analysis of Hierarchical Factor Recovery

While few works study finite-sample guarantees for CRL (Acartürk et al., 2024), empirical stability is crucial for interpreting a fitted hierarchical latent-factor model. We therefore ask how the recovered factor alignments change when the domain set changes. Two conclusions matter for the main paper:

- The full three-factor parameterization *can* be sensitive to the choice of base models, especially for the upstream factor z_1 .
- The most stable result is the relation from the instruction-aligned factor to the math-aligned factor.

Adding two related domains. We add Qwen2.5-3B and Llama-3.2-3B to the four-domain primary analysis set. These two domains have principal-component subspaces that remain close to the main cluster, but they contain fewer models than the four domains used in the main paper. The recovered alignment in Figure 14 shows the same downstream pattern as the primary fit: z_2 remains aligned with IFEval and z_3 remains aligned with MATH, while z_1 rotates toward a broader tradeoff between general reasoning and more specialized benchmarks.

This is the appendix version of the main-text stability claim in Section 4.3 and Fig. 7. The subspace spanned by the downstream pair remains qualitatively stable, and the fitted $z_2 \rightarrow z_3$ relation remains positive, even though the precise interpretation of z_1 changes under alternative domain choices.

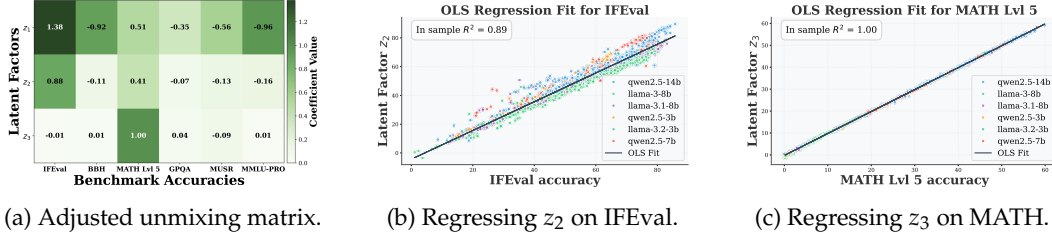


Figure 14: Sensitivity analysis with two additional related domains. The downstream z_2/z_3 pair remains aligned with instruction following and mathematical reasoning, while the upstream factor changes more under the enlarged domain set.

G Archived Supporting Results

G.1 Full forward base-model intervention panel

The main text uses directional plots rather than a seed-level intervention table. This table reports the full forward intervention panel across the six main-paper benchmarks together with the held-out clean instruction-following check. The clearest pattern is strong instruction gains in all completed base families and larger secondary MATH transfer in Qwen2.5-7B than in the two Llama families.

Domain	Seed	Δ held-out clean	Δ IFEval	Δ BBH	Δ MATH	Δ GPQA	Δ MUSR	Δ MMLU-Pro
Qwen2.5-7B	17	+2.20	+26.02	+1.41	+8.69	+2.90	+0.40	+0.54
Qwen2.5-7B	23	+5.30	+25.18	+1.13	+10.65	+1.79	+1.19	+0.85
Llama-3.1-8B	17	+24.95	+43.17	+1.79	+1.28	-1.79	+5.42	-0.61
Llama-3.1-8B	23	+22.75	+41.37	+1.37	+1.06	-2.68	+5.82	+0.19
Meta-Llama-3-8B	17	+23.70	+42.69	+2.14	-0.08	-2.90	+3.04	-0.69
Meta-Llama-3-8B	23	+22.60	+42.21	+2.93	+0.76	-1.56	+5.16	-1.28

Table 6: Full forward intervention panel for the completed base-model TuluIF-Clean-v1 runs. “Held-out clean” denotes the benchmark-prompt-disjoint normalized exact-match instruction-following check used during training. The intervention strongly improves instruction following in all completed base families, while secondary MATH transfer remains much larger in Qwen2.5-7B than in the two Llama families.

G.2 Tulu validation constraint re-evaluation by starting state

Figure 9b in the main text is based on a constraint-scored re-evaluation on the filtered Tulu validation set. Table 7 reports the exact hard- and soft-constraint deltas behind that

figure. One legacy Qwen2.5-7B base-start tuned merged checkpoint is unavailable, so the comparison uses the reduced nine-run panel that could be re-evaluated cleanly.

Model	Start	Seed	Δ hard constraint acc.	Δ soft constraint acc.
Qwen2.5-7B	Base	23	+3.63	+7.25
Llama-3.1-8B	Base	17	+4.50	+11.12
Llama-3.1-8B	Base	23	+5.12	+10.75
Meta-Llama-3-8B	Base	17	+4.12	+10.94
Meta-Llama-3-8B	Base	23	+4.62	+10.94
Qwen2.5-7B-Instruct	Instruct	17	+1.12	+2.19
Qwen2.5-7B-Instruct	Instruct	23	+1.75	+2.31
Meta-Llama-3-8B-Instruct	Instruct	17	+2.12	+5.06
Meta-Llama-3-8B-Instruct	Instruct	23	+1.38	+4.56

Table 7: Constraint-based Tulu validation deltas for the same TuluIF-Clean-v1 recipe under different starting states. The gains are consistently larger from base checkpoints than from already-instruct checkpoints.

G.3 Recovered factorization before benchmark alignment

The semantic labels in the main paper are introduced only after a post-hoc lower-triangular coordinate choice. Accordingly, Figure 15 shows a compact view of the raw HCA outputs before benchmark alignment: four domainwise source-correlation summaries together with the shared mixing matrix. The main paper’s semantic interpretation is added after these fitted objects are obtained.

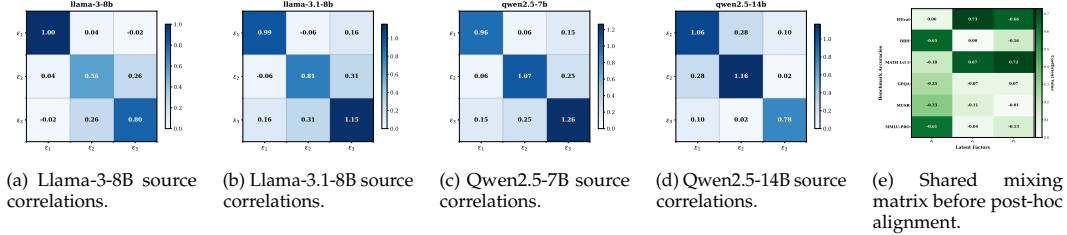


Figure 15: Compact view of the raw HCA fit before benchmark alignment. The domainwise source-correlation summaries remain low-MIC, while the shared mixing matrix shows the raw latent coordinates before the benchmark-based re-labeling used in the main text.

Figure 16 shows the full HCA recovery panel before benchmark alignment. It reports source summaries, source-variable views, family-specific coefficient matrices, latent-capability views, the shared mixing matrix, and the benchmark-space reconstructions.

G.4 Low-rank approximation diagnostics

The hierarchical fit is built on a rank-3 latent space, so it is useful to inspect the induced approximation error benchmark by benchmark. Figure 17 reports the actual-versus-predicted comparisons for the six benchmarks in the primary analysis set. Four benchmarks are approximated almost perfectly, while GPQA and especially MUSR exhibit larger residual structure. This benchmark-wise residual structure is one reason to read the fitted hierarchy as an approximation within the selected families.

This benchmark-wise asymmetry also helps explain two patterns emphasized elsewhere in the appendix. First, when the broad upstream factor rotates toward GPQA and MUSR under alternative domain choices, its interpretation becomes less stable than that of the downstream z_2/z_3 pair. Second, in the filtered-model robustness check below, removing badly fine-tuned models compresses variance on BBH and MMLU-Pro and increases the relative weight of these harder-to-fit GPQA/MUSR dimensions, which in turn degrades the overall fit diagnostics.

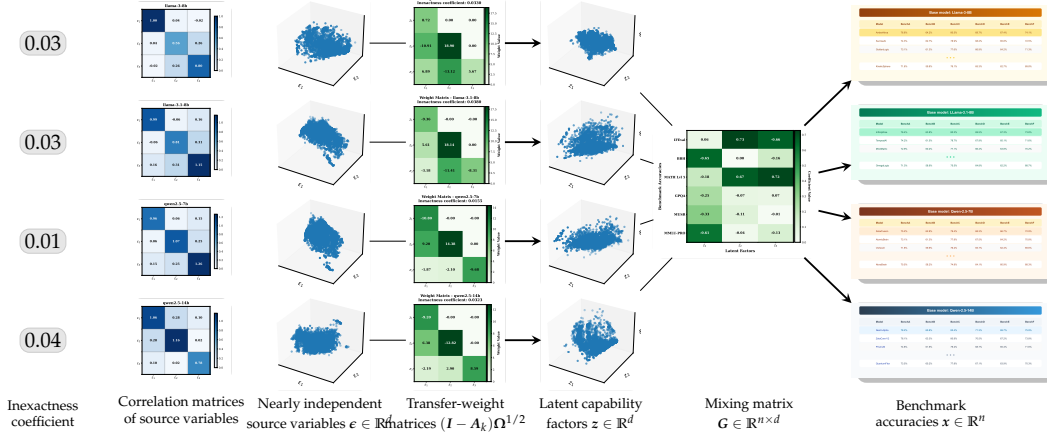


Figure 16: Full raw HCA recovery panel before benchmark alignment for the four-domain primary analysis set. The figure shows the full fitted decomposition: source summaries, source-variable views, family-specific coefficient matrices, latent-capability views, the shared mixing matrix, and benchmark-space reconstructions. The leaderboard screenshots are illustrative; the fitted objects themselves are learned from the Open LLM Leaderboard v2 data used throughout the paper.

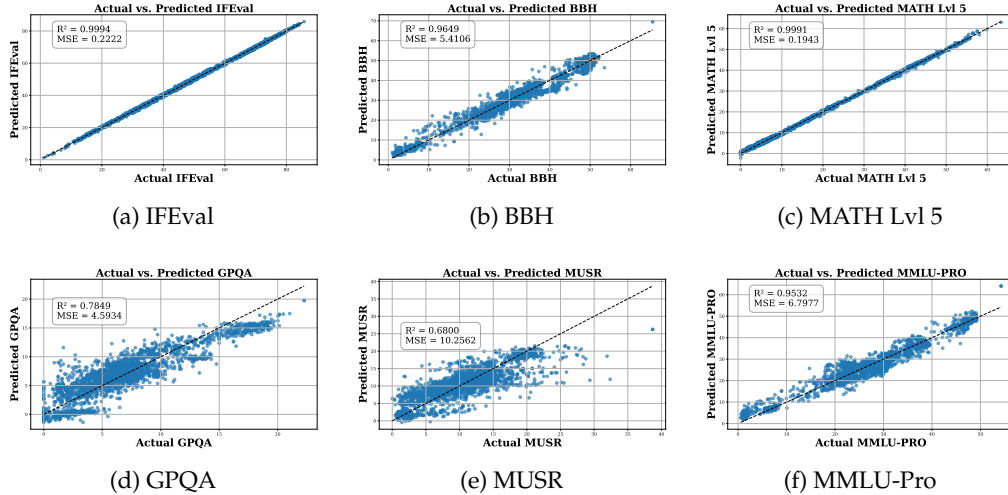


Figure 17: Benchmark-wise approximation diagnostics for the rank-3 latent space. The fitted space captures most of the variation in four benchmarks very closely, while GPQA and MUSR show the largest remaining residual structure.

H Exploratory Extensions

This section collects exploratory diagnostics, side analyses, and replications that help interpret the main results.

This subsection documents descriptive regularities in the leaderboard data that helped motivate the family-conditioned analysis.

Existing predictive models for language model performance are typically restricted to pretrained models. This is not unexpected, since it is hard to characterize post-training gains in terms of a small set of observed covariates. The plots below are descriptive. They show why pooled post-training comparisons are hard to interpret without conditioning on base-model family.

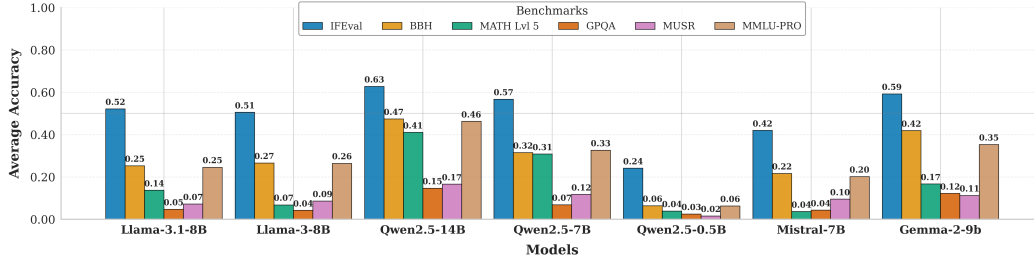


Figure 18: The average benchmark performance of fine-tuned models on the open LM leaderboard with three base models in different sizes.

As illustrated in Figure 18, models fine-tuned on stronger base models tend to exhibit uniformly better performance across all benchmarks. In other words, base model acts as a common confounder of benchmark performance and also of the apparent gains achievable through post-training. As a descriptive check, we compute backdoor-style plug-in estimates using log pretraining compute as an observed covariate. Figure 19 suggests that post-training is associated with larger gains on math and instruction-following benchmarks than on general reasoning and QA-style tasks, and that these associations differ between Llama- and Qwen-based models.

Remark 4. *These estimates are descriptive because they require conditional ignorability to be read as treatment effects, and that assumption is hard to justify when model families receive systematically different tuning recipes. We use them to motivate conditioning on base-model family.*

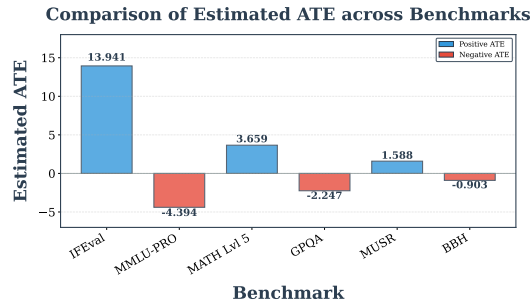


Figure 19: Estimates of the average effect of fine-tuning.

H.1 Descriptive scaling and post-training heterogeneity

Scaling-law analysis (Kaplan et al., 2020) has been widely adopted to predict benchmark performance from pretraining compute. Later work (Ruan et al., 2024; Ren et al., 2025) used sigmoid fits to model low-dimensional summaries of multi-benchmark evaluation data. Could scaling-law style summaries alone explain the leaderboard data? As an exploratory

descriptive exercise, we let $C \approx 6 \cdot N \cdot D$ denote pretraining compute and fit sigmoid regressions of the form $Y \approx \frac{L}{1 + \exp(-k(\log C - \log C_0))} + \tau T + b$, where T indicates whether a model is fine-tuned or pretrained.

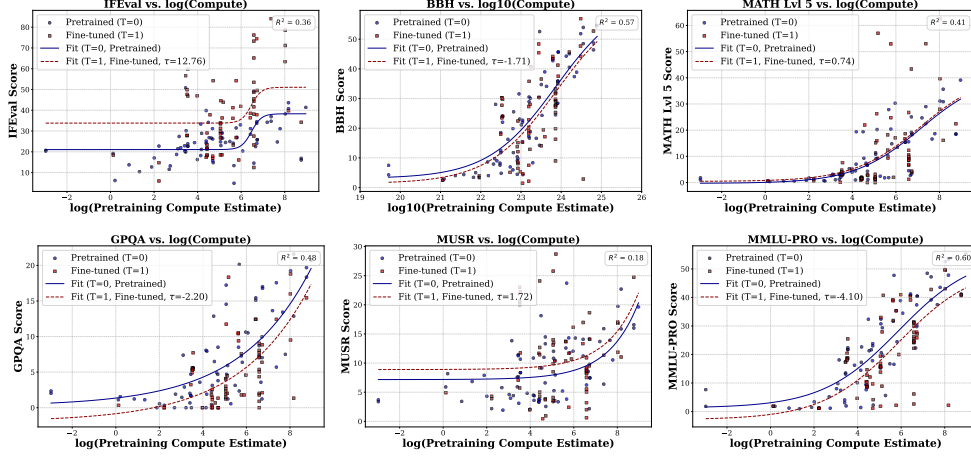


Figure 20: Exploratory sigmoid scaling-law fits for the six benchmarks when pretrained and fine-tuned models are pooled together.

We fit these sigmoid curves to the benchmark results of officially released models on the leaderboard (see Figure 20). The descriptive pattern is that scaling-law style fits appear tighter on BBH, MMLU-Pro, and GPQA than on the remaining benchmarks. A plausible interpretation is that the former three are relatively more knowledge-driven, whereas instruction following, mathematical reasoning, and multi-step soft reasoning are more sensitive to post-training choices. We include this section to show why pooled scaling summaries are informative but insufficient for the family-conditional transfer questions studied in the main paper.

H.2 Why base model is the preferred domain variable

The main paper defines domains by base-model family because that control absorbs architectural, data, and recipe differences from pretraining that are difficult to represent with a single scalar proxy. The alternative PCA diagnostics in Figure 12 show that domains defined by MoE status, coarse (N, D) bins, or upload-time buckets can reveal heterogeneity, but they do not isolate the same within-family structure as cleanly as base-model conditioning.

H.3 Filtered-model robustness check

We rerun the benchmark-alignment step after removing models whose average benchmark performance falls below the corresponding pretrained checkpoint. The downstream instruction-aligned / math-aligned pair remains visible, but the fit becomes less stable because filtering compresses variance on the broad reasoning benchmarks and increases the relative weight of the harder-to-fit GPQA/MUSR dimensions. This analysis is a robustness check under a narrower variance structure.

H.4 Open LLM Leaderboard v1 archive

We also apply the same screening-and-fit procedure to the older Open LLM Leaderboard benchmark suite. The recovered structure differs in semantics, as expected under a different benchmark panel, but the same screening-and-fit procedure still recovers an interpretable later-factor relation.

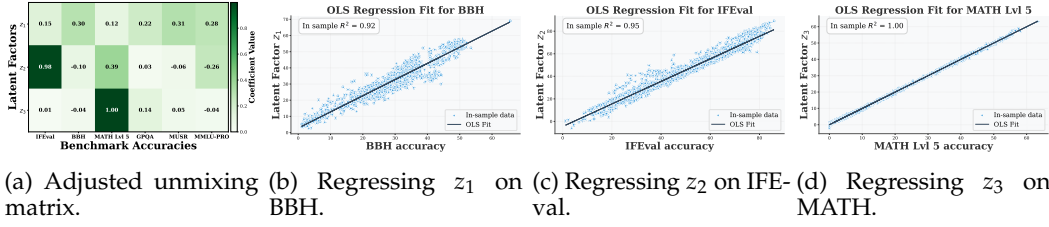


Figure 21: Filtered-model robustness check. The downstream z_2/z_3 pair remains benchmark-aligned after excluding badly fine-tuned models, while the upstream factor becomes noisier.

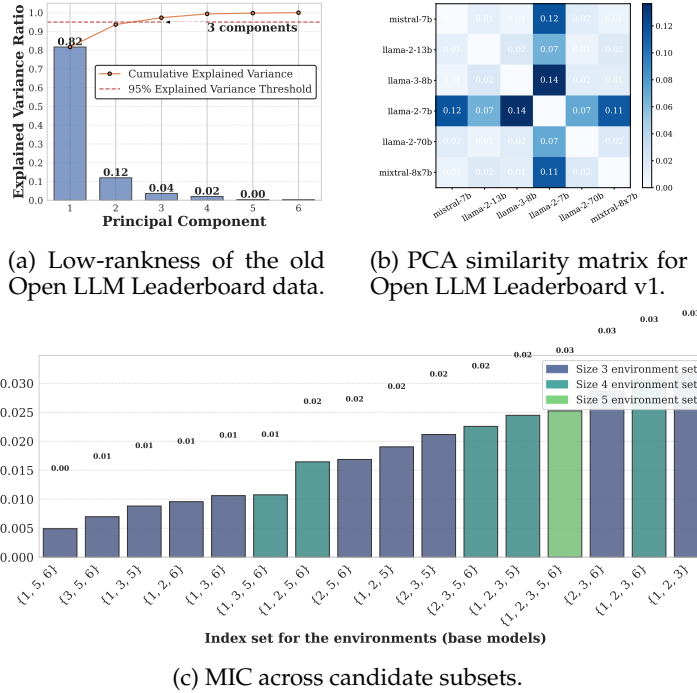


Figure 22: Exploratory archive for Open LLM Leaderboard v1. The same screening-plus-fit procedure reveals a coherent low-MIC family cluster in the older benchmark panel as well.

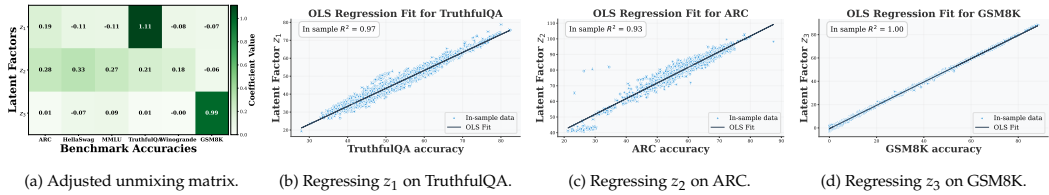


Figure 23: Post-hoc benchmark alignment for the exploratory Open LLM Leaderboard v1 archive. The semantics differ from the v2 panel, but the fitted hierarchy again isolates an interpretable downstream factor.

H.5 MMLU-by-task archive

HCA can also be applied to task-level decompositions within a single benchmark. The examples below show how the same procedure behaves when the benchmark panel is replaced by a small set of MMLU subject clusters.

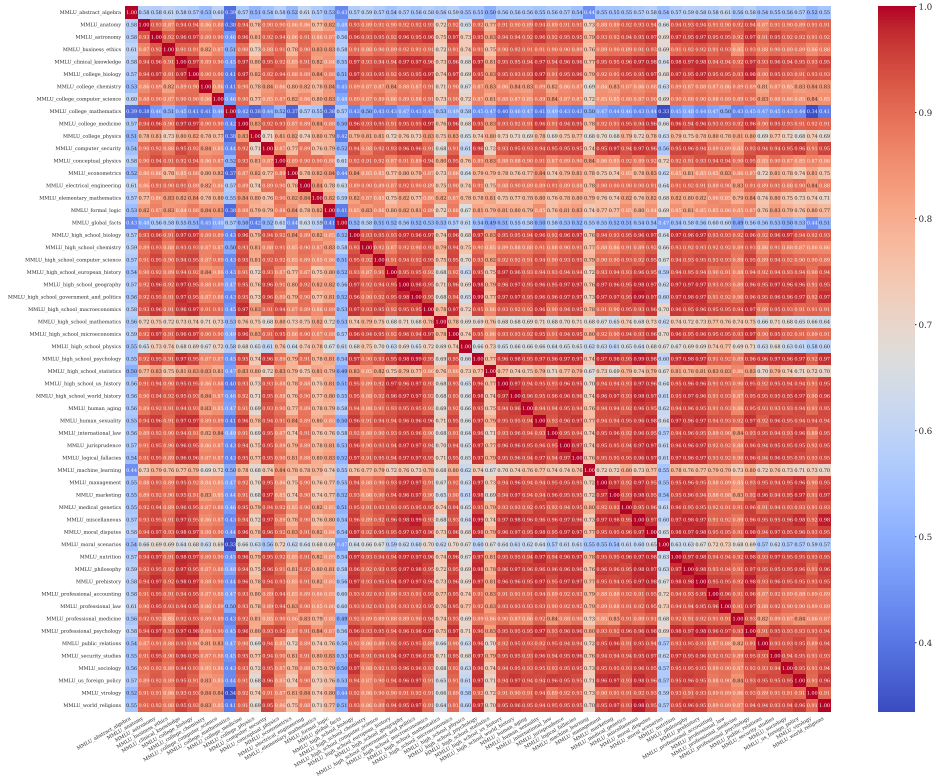


Figure 24: Task-level correlation matrix for the MMLU benchmark. The broad positive dependence reinforces the idea that task-level decomposition is meaningful but substantially more knowledge-driven than the six-benchmark main-paper panel.

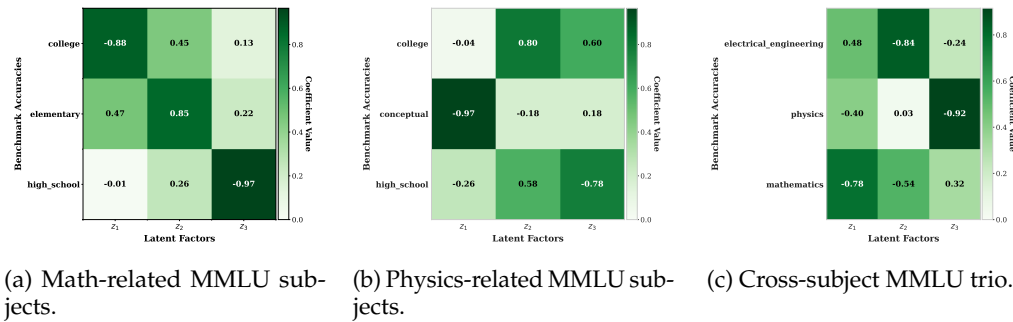


Figure 25: Exploratory HCA applications to MMLU-by-task panels. These examples are not used in the main narrative, but they illustrate that the same within-family procedure can also uncover a low-rank pattern inside narrower benchmark collections.

H.6 Local versus global matrix completion

One side benefit of conditioning on a coherent family cluster is that the resulting local low-rank structure can help with missing-entry imputation. The two summary plots in Figure 26

compare local versus global matrix completion under random-missing and block-missing regimes for the Qwen2.5-14B domain. The local approach is consistently competitive and often materially better.

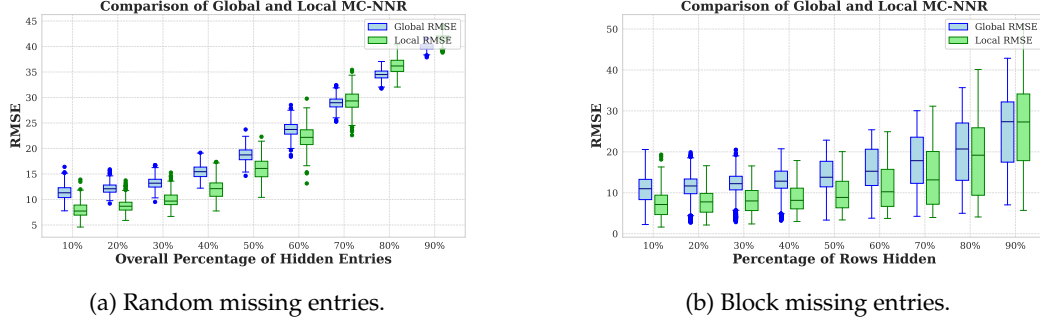


Figure 26: Exploratory matrix-completion archive. Restricting attention to a coherent within-family submatrix often improves imputation accuracy relative to a single global low-rank model fit to the full leaderboard.

To make the setup explicit, we use the same synthetic matrix-completion protocol. Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ denote the leaderboard performance matrix with $d = 6$ benchmarks and partially missing entries. Restricting attention to the Qwen2.5-14B domain, we compare a global approach that runs nuclear-norm regularization on the full leaderboard matrix against a local approach that runs the same procedure only on the coherent within-family submatrix. This archive shows that restricting to a coherent family cluster can also help missing-entry imputation in this setting.

We evaluate two missingness regimes. In the random-missing regime, each benchmark entry is deleted independently with probability $p = 0.8$. In the block-missing regime, three benchmarks are fully observed and the remaining three have rows deleted with probability $p = 0.5$. The summary plots in [Figure 26](#) average over these settings. [Figure 27](#) restores the full block-missing breakdown over all $\binom{6}{3} = 20$ possible choices of fully observed benchmarks, where benchmark indices $1, \dots, 6$ correspond to IFEval, BBH, MATH Lvl 5, GPQA, MUSR, and MMLU-Pro respectively.

H.7 Domains defined by pretraining compute

As a weaker alternative to conditioning directly on base-model family, we also group models by coarse pretraining-compute bins. This analysis is informative precisely because it is less clean: some low-MIC structure remains for the lower-compute bins, but the grouping is visibly noisier than base-model conditioning and degrades quickly when larger heterogeneous bins are added.

Pretraining compute range	[0, 10)	[10, 20)	[20, 40)	[40, 80)	[80, 160)	[160, 320)	≥ 320
Number of models	539	371	199	405	1540	181	154

Table 8: Exploratory partition of matched models by coarse pretraining-compute class, where compute is measured by $N \times D$ in sextillion parameter-token units.

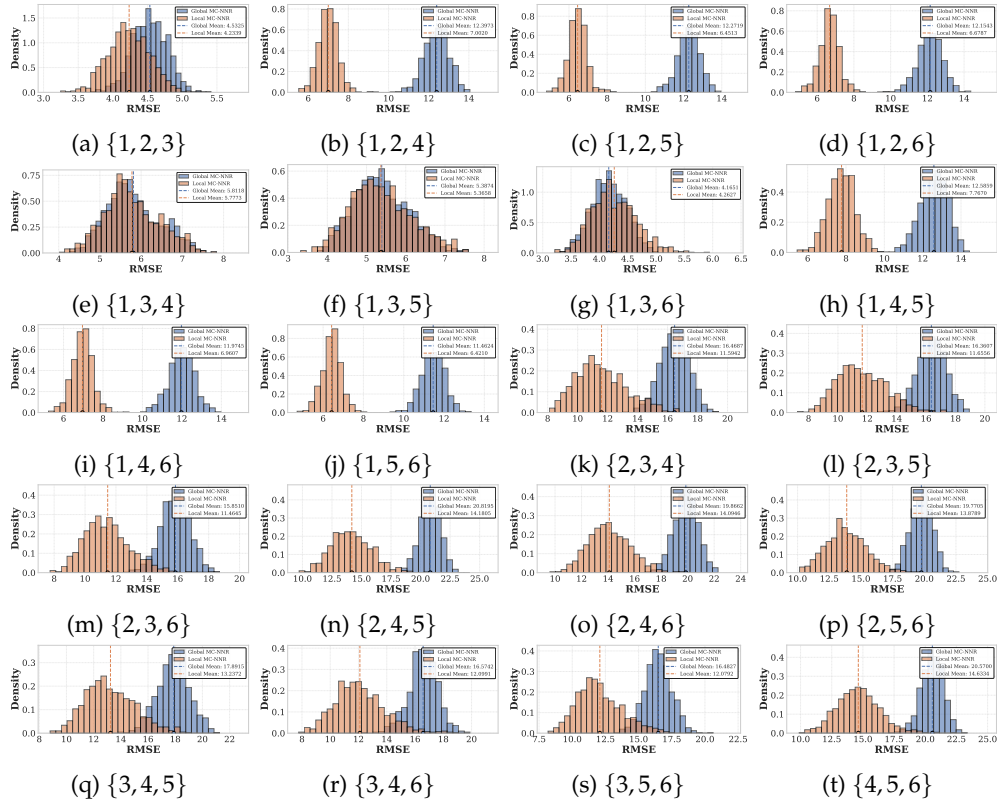
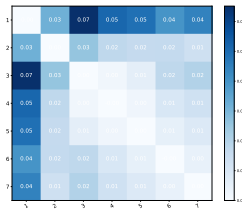
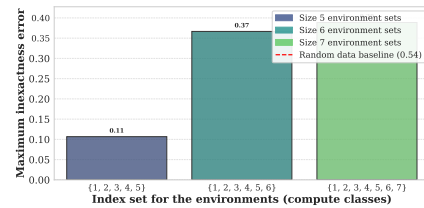


Figure 27: RMSEs of global versus local matrix completion for all possible choice of fully observed benchmarks in the block-missing setting. The local approach is never worse and is often materially better, which is consistent with the broader family-conditioned low-rank story used in the main paper.



(a) PCA similarity across compute-defined domains.



(b) MIC after adding higher-compute classes.

Figure 28: Exploratory compute-domain archive. Some structure remains when domains are defined by pretraining compute, but the fit is less targeted than when domains are defined directly by base-model family.