From Benchmarks to Skills: Low-Rank Factors for LLM Evaluation

Aviya Maimon[1,2] **Amir David Nisan Cohen**[2] **Gal Vishne**[3]
Shauli Ravfogel[4] **Reut Tsarfaty**[1]

[1]Bar-Ilan University [2]OriginAI [3]Data Science Institute, Columbia University
[4]Center for Data Science, New York University

{aviyamn, amirdnc, gal.vishne, shauli.ravfogel, reut.tsarfaty}@gmail.com

Abstract

Current evaluations of large language models (LLMs) rely heavily on a growing collection of benchmarks and on aggregate benchmark scores, yet it remains unclear what this comparison actually captures, and what these scores reveal about models' underlying capabilities. Here, we propose a new paradigm for LLM evaluation, by asking whether benchmark performance reflects many independent abilities, or rather, relies on a small number of shared dimensions. To answer this, we apply Factor Analysis (FA) to a massive performance matrix of LLMs versus benchmarks (60×44) revealing an *intrinsically low-rank* structure of that matrix. That is, a small number of latent factors captures most of the structure in the full task space. This low-rank geometry reveals substantial redundancy across existing tasks and explains why many benchmarks appear to be measuring overlapping abilities. We further show that these latent factors correspond to coherent, skill-like, dimensions of LLM behavior. Leveraging this latent skill-space, we deliver three practical tools for LLM evaluation and downstream users: (i) identifying redundant tasks, (ii) profiling new models using a small subset of tasks, and (iii) selecting models aligned with desired skill profiles. Our method provides a solid alternative to the de-facto standard of a single aggregate score, and establishes an interpretable and practical framework for understanding and benchmarking LLM core capabilities.

1 Introduction

Large Language Models (LLMs) now achieve state-of-the-art (SOTA) performance across a rapidly growing set of benchmarks, spanning question answering (Rajpurkar et al., 2016), complex reasoning (Lyu et al., 2023), summarization (Hermann et al., 2015), and many others.

As these benchmarks proliferate, evaluation has increasingly relied on long lists of task-specific scores or on simple aggregate averages (e.g., Chatbot Arena by HuggingFace (2024)). Yet it remains unclear what such evaluations reveal about models' underlying capabilities. Although individual datasets are designed to assess specific abilities, in practice tasks often combine multiple underlying skills, making it unclear how dataset scores relate to shared model behaviors. This raises a set of fundamental questions: What latent information is captured by the diverse evaluation benchmarks? Moreover, to what extent do different datasets overlap in what they measure? And how can we meaningfully compare models for a given downstream case beyond aggregate scores?

In this paper, we propose a new, data-driven paradigm for LLM evaluation, inspired by psychometric theory. Specifically, we propose to treat benchmark tasks as test items, and LLMs as subjects, and to apply exploratory Factor Analysis (FA; Thurstone (1931)) (§2) to empirical models' performance data. This analysis is designed to reveal whether LLM evaluation records exhibit a low-rank structure, in which a small number of latent factors can explain most of the variation in model performance across tasks. This kind of structure would indicate that performance across many datasets is governed by a small number of shared dimensions, providing a compact view of model behavior.

To uncover this structure, our methodology proceeds in three stages: *comprehensive leaderboard construction*, *factor analysis*, and *skill naming*. Concretely, we first assemble a comprehensive leaderboard evaluating 60 diverse LLMs on 44 commonly used benchmarks, forming a new and diverse model–task performance matrix that serves as the basis for our analysis (§3). Then, we apply exploratory FA to this matrix to recover a small set of latent dimensions that summarize shared performance patterns across tasks. Lastly, we interpret

Transactions of the Association for Computational Linguistics, vol. 14, pp. 1660–1684, 2026. https://doi.org/10.1162/TACL.a.745
Action Editor: Francois Yvon. Submission batch: 7/2025; Revision batch: 1/2026; Published 7/2026.

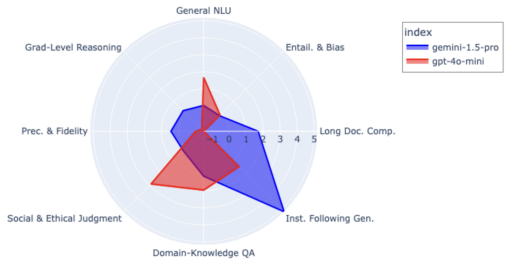


Figure 1: Despite similar Chatbot Arena (HuggingFace, 2024) ratings (1322 vs. 1316), the models differ substantially in latent skill profiles, highlighting limitations of aggregate preference scores.

each latent factor by analyzing its highest-loading tasks and assign it a descriptive label, yielding a set of coherent dimensions of model behavior. Throughout this paper, we refer to these labeled dimensions as *skills* (§4).

The latent skill space reveals systematic differences across models, which are obscured by aggregate evaluation scores. For example, it reveals that models with similar Chatbot Arena rankings can occupy markedly different positions in the latent skill space, reflecting distinct strengths across the underlying dimensions (Figure 1; §5.2). We show that the recovered latent structure is robust to task and model perturbations (§5.1). We also show that it generalizes to unseen datasets (§6.1).

Crucially, beyond the theoretical implications and the interpretation of models' scores, the existence of a low-rank latent skill space enables practical uses. Subsequently, we deliver three automatic tools for practical applications: (i) quantifying the novelty or redundancy of a new benchmark (§6.1), (ii) efficiently profiling a new model from a small subset of tasks (§6.2), and (iii) selecting a model for unseen tasks by estimating their latent skill requirements (§6.3). Together, these applications define a skill-centric approach to LLM evaluation that moves beyond opaque benchmark averages.

Taken together, our results establish FA as a principled and interpretable framework for understanding the latent structure of LLM evaluation data, giving rise to practical tools for more effective, efficient and far more transparent model assessment. We release our code, leaderboard, analytical matrices, and tools to facilitate further research on LLM evaluation and benchmarking.

2 Proposal: Modeling Latent Skills

In this section we introduce our latent-skill modeling framework. We first motivate our approach, then we formalize our proposed factor-analytic approach for extracting shared performance dimensions from benchmark data, and lastly we describe how these dimensions may be interpreted.

2.1 Methodological Motivation

Our approach draws inspiration from classical psychometrics, which studies how latent constructs can be inferred from patterns of responses across many items. FA was developed precisely to separate *shared* variance (interpreted as a common underlying ability) from *item-specific* variance (Spearman, 1904), and later became the foundation of widely used measurement frameworks, e.g., the Big Five personality traits (Tupes and Christal, 1961; Costa and McCrae, 1992).

In our setting, tasks play the role of *specific items* and LLMs the role of *respondents*. The analogy is mathematical (rather than cognitive): FA provides a principled tool for identifying shared structure in model performance across tasks, by modeling cross-task covariance. Viewed through this lens, FA enables the isolation of latent dimensions that explain systematic performance variation across benchmarks, separated from task-specific noise.

2.2 Formalization

Our formal approach, in a nutshell, is as follows. We begin with a *performance matrix* $\mathbf{\Pi}$ (§2.3): a *models* $\times$ *tasks* matrix that contains the scores of all tasks on all LLMs. We then use *Principal Axis Factoring* (PAF; Harman, 1976), treating each task (column) as a variable and each model (row) as an observation. PAF extracts factors accounting for *shared* variance across tasks while discarding task-specific noise (§2.4). PAF yields three benefits:[1] (i) the resulting factors **generalize** more reliably to unseen tasks, since task-specific variance is removed; (ii) the solution is more **robust to outliers**, as variance unique to one task is treated

[1]Alternative decompositions such as PCA (Bishop, 2006) were considered; PCA is the closest baseline for variance-based dimensionality reduction. However, while PCA also yields low-dimensional representations, it maximizes the *total* variance, mixing *shared* and *unique* variation — so a single high-variance idiosyncratic feature can dominate. PAF instead models only shared variance across variables, making it more appropriate here. Our comparison in §5.1 confirms that PAF yields more stable and interpretable components.

as noise and cannot distort the factors; and (iii) the extracted factors are more **interpretable**, reflecting broad abilities rather than fragmented patterns. Finally, we present how we name the extracted latent factors as skills, and conceptualize an intuitive, interpretable *skill-based leaderboard* (§2.5).

2.3 The Performance Matrix

We first define a leaderboard-style performance matrix $\mathbf{\Pi}$ to capture the systematic comparison of M LLMs across B diverse tasks. Constructing $\mathbf{\Pi}$ involves (i) aggregating publicly available tasks, (ii) harmonizing heterogeneous evaluation metrics onto a unified 0–10 scale, and (iii) recording model–task scores. Further details on the actual construction of our leaderboard are provided in §3.

2.4 Principal Axis Factoring

Generative view. Given $M \times B$ performance matrix $\mathbf{\Pi}$, PAF assumes factor decomposition model

$$\mathbf{\Pi} \;=\; \mathbf{\Theta}\mathbf{\Lambda}^\top + \boldsymbol{\epsilon}, \tag{1}$$

with $\mathbf{\Theta} \in \mathbb{R}^{M\times C}$ (*skills* matrix), $\mathbf{\Lambda} \in \mathbb{R}^{B\times C}$ (task *loadings*), and $\boldsymbol{\epsilon} \in \mathbb{R}^{M\times B}$ (task-specific noise).

The *skills* matrix scores each model on each latent skill, and the *loadings* are coefficients measuring how strongly a latent skill c contributes to the explanation of the performance of task b.

Assumptions. For every model $i \in M$, let $\mathbf{p}_i \in \mathbb{R}^B$ denote its observed task-performance vector:

$$\mathbf{p}_i := \boldsymbol{\theta}_i^\top \mathbf{\Lambda}^\top + \boldsymbol{\varepsilon}_i^\top \in \mathbb{R}^B,$$

where $\boldsymbol{\theta}_i^\top$ and $\boldsymbol{\varepsilon}_i^\top$ are the i-th rows of latent skills and task-specific noise. We treat $\boldsymbol{\theta}_i$ and $\boldsymbol{\varepsilon}_i$ as random vectors with:

$$\begin{aligned} &\mathbb{E}[\boldsymbol{\theta}_i] = \mathbf{0}, \quad \mathrm{Cov}[\boldsymbol{\theta}_i] = \mathbf{\Gamma} = \mathbf{I}_C \in \mathbb{R}^{C\times C}, \\ &\mathbb{E}[\boldsymbol{\varepsilon}_i] = \mathbf{0}, \quad \mathrm{Cov}[\boldsymbol{\varepsilon}_i] = \mathbf{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_B), \\ &\qquad\qquad \boldsymbol{\theta}_i \perp \boldsymbol{\varepsilon}_i. \end{aligned}$$

We adopt the *orthogonal-factor* convention $\mathbf{\Gamma} = \mathbf{I}_C$, so, each latent skill has unit variance, and skills are a-priori uncorrelated.

Population covariance. These assumptions give the factor-analysis decomposition:

$$\mathbf{R} := \mathrm{Cov}[\boldsymbol{p}] = \mathbf{\Lambda}\mathbf{\Gamma}\mathbf{\Lambda}^\top + \mathbf{\Psi} = \mathbf{\Lambda}\mathbf{\Lambda}^\top + \mathbf{\Psi},$$

so that each off-diagonal entry of $\mathbf{R}$ is explained by the low-rank term $\mathbf{\Lambda}\mathbf{\Lambda}^\top$.

The diagonal element $\left(\mathbf{\Lambda}\mathbf{\Lambda}^\top\right)_{jj}$ is known as the *communality* of task j:

$$h_j^2 \;=\; \sum_{c=1}^{C} \lambda_{jc}^2,$$

namely the portion of its variance explained by the common factors. The remainder $\psi_j = 1 - h_j^2$ is its *uniqueness* (task-specific variance).

The loading matrix $\mathbf{\Lambda}$ is identifiable only up to rotation, so PAF focuses on the *subspace* it spans. Orthogonal rotations such as Varimax maximize sparsity and improve interpretability.

Learning objective. Given $\mathbf{R}$ and a chosen number of skills $C \ll B$, PAF estimates $(\widehat{\mathbf{\Lambda}}, \widehat{\mathbf{\Psi}})$ by

$$\min_{\mathbf{\Lambda},\ \mathbf{\Psi}\ \text{diagonal}} \left\|\mathbf{R} - \mathbf{\Lambda}\mathbf{\Lambda}^\top - \mathbf{\Psi}\right\|_F^2. \tag{2}$$

See appendix A for the iterative algorithm used to estimate $\mathbf{\Lambda}$ and $\mathbf{\Psi}$.

Representing a new model. Given a new model with task scores $\mathbf{p}_{\text{new}} \in \mathbb{R}^B$[2], we compute its skill vector using *regression (Thomson) scores*, which minimize $\mathrm{MSE} = \mathbb{E}[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2]$. This involves two steps: regress the latent skills from FA ($\mathbf{\Theta}$) on the observed scores ($\mathbf{\Pi}$) via population multiple-OLS and then apply $\mathbf{B}_{\text{reg}}$ to the new task vector:

$$\mathbf{B}_{\text{reg}}^\top = \mathbf{\Lambda}^\top \mathbf{R}^{-1}, \qquad \widehat{\boldsymbol{\theta}}_{\text{reg}} = \mathbf{B}_{\text{reg}}^\top \mathbf{p}_{\text{new}}. \tag{3}$$

Note that because $\mathbf{B}_{\text{reg}}$ depends only on $\widehat{\mathbf{\Lambda}}$ and $\widehat{\mathbf{\Psi}}$, it can be stored once and reused for any new model via a single matrix–vector product.

2.5 Labeling Factors as Skills

PAF yields C latent factors capturing shared structure across tasks. To make them interpretable, we assign each factor a concise label based on its highest-loading tasks. Note that factor labels are used only for exposition; all analyses depend on the learned *loadings* and scores.

Factor interpretation is grounded in the semantic commonalities among the highest-loading tasks for each factor, which serve as diagnostic indicators of the underlying dimension. To support scalability and to reduce individual subjective biases, we employ an LLM-assisted labeling procedure that proposes names for each factor given short summaries of its representative tasks. The procedure is agnostic to any pre-defined skill taxonomy.

[2] z-scored using the task means and standard deviations of the original performance matrix.

We present the named latent dimensions in §4.1 and refer to them as *skills* throughout the paper. The role of labeling is interpretive: it provides a concise semantic summary of each latent dimension uncovered by FA, but does not influence factor extraction, loadings, or model scores. All quantitative results and robustness analyses are therefore independent of the specific phrasing used to describe each factor. Full details of the naming procedure are provided in appendix B.

3 The Task-Based Leaderboard

Developers face major challenges in reconciling diverse evaluation protocols, including diverse metrics and task formats, and in ensuring scalability and reproducibility. To mitigate this, we deliver a standardized, transparent leaderboard comparing LLMs across *skills*, rather than tasks, highlighting their current strengths and limitations, and guiding future research. Concretely, we compile a matrix of 60 models evaluated on 44 tasks, which forms the empirical basis for our FA. Here, we describe the construction of the performance matrix $\mathbf{\Pi}$.

Tasks. We evaluate models on a suite of $B = 44$ publicly available tasks, balancing classification and generation tasks across short- and long-context settings. Of these, 25 are classification-based: 13 binary (e.g., QNLI, BoolQ) and 12 multiple-choice (e.g., MMLU, MedQA). The remaining 19 generation tasks cover tasks such as summarization (XSum), open-domain QA (TriviaQA), and dialogue (DailyDialog), and span different domains such as legal, medical, and conversational. The complete list of tasks is provided in appendix D. Integrating these diverse tasks into a single leaderboard enables comprehensive model evaluation via the method described in §2.4.

Prompts. For fair and robust evaluation across diverse models, we designed three prompts per task that produced valid outputs across models (see appendix E). Task scores average performance over all three prompt variants per task. Prompt averaging may weaken correlations, but the dominant factor structure remains stable.

Metrics. For all classification tasks we used standard metrics, prioritizing the most common metric when multiple exist. For generation tasks, we adopted an LLM-as-a-judge approach (Zheng et al., 2023) using Deepseek-R1 (DeepSeek-AI, 2025). Full detailing of the tasks and evaluation method are provided in appendix D.

Normalization and Standardization. All task scores were normalized to [0, 1] and then standardized (zero mean, unit variance) before FA. Because FA uses correlations, results are invariant to these linear transformations.

Models. We compile a leaderboard spanning all tasks on $M = 60$ instruction-tuned LLMs from 24 model families, including both open-source (e.g., LLaMA) and proprietary systems (e.g., Gemini). Most models (53/60) use decoder-only architectures, while the remainder follow an encoder–decoder design. Open-source parameter counts range from 2B to 27B; proprietary sizes are largely undisclosed. The set includes models from commercial labs (e.g., GPT), academia (e.g., Bloom), and startups (e.g., DeepSeek), with several RLHF variants (e.g., Claude).

4 Results: A Skill-Based Leaderboard

4.1 The Latent Skills

We applied PAF to the model–task performance matrix we constructed in §3, and uncovered 8 latent dimensions, labeled using the procedure in §2.5. We refer to these dimensions as *skills*. The recovered skills are: *(1) General NLU, (2) Entailment & bias, (3) Long-document comprehension, (4) Instruction following & generation, (5) Domain QA, (6) Social & ethical judgment, (7) Precision & fidelity* (i.e., exact adherence to reference outputs, numerical values or factual details), and *(8) Grad-level reasoning*. Table 1 lists each factor with its name, description, and most strongly associated tasks.

Assigning human-interpretable names to each skill relied on a two-step procedure (§2.5) which leverages GPT-4o and DeepSeek-R1 (DeepSeek-AI, 2025) to summarize the most influential tasks. The number of latent factors (C) was chosen via Kaiser's $\lambda > 1$ rule and an 85% cumulative variance threshold (Fig.4 in appendix F) (Kaiser, 1960; Harman, 1976), ensuring that each dimension retained shared variance without overfitting noise.

Note that the resulting granularity reflects a balance between statistical support, robustness, and interpretability for the current task suite — rather than a claim of a unique "correct" skill resolution. Also note that we use the term *skill* to descriptively refer to a latent dimension of model behavior, rather to refer to, e.g., a pre-defined psychological trait.

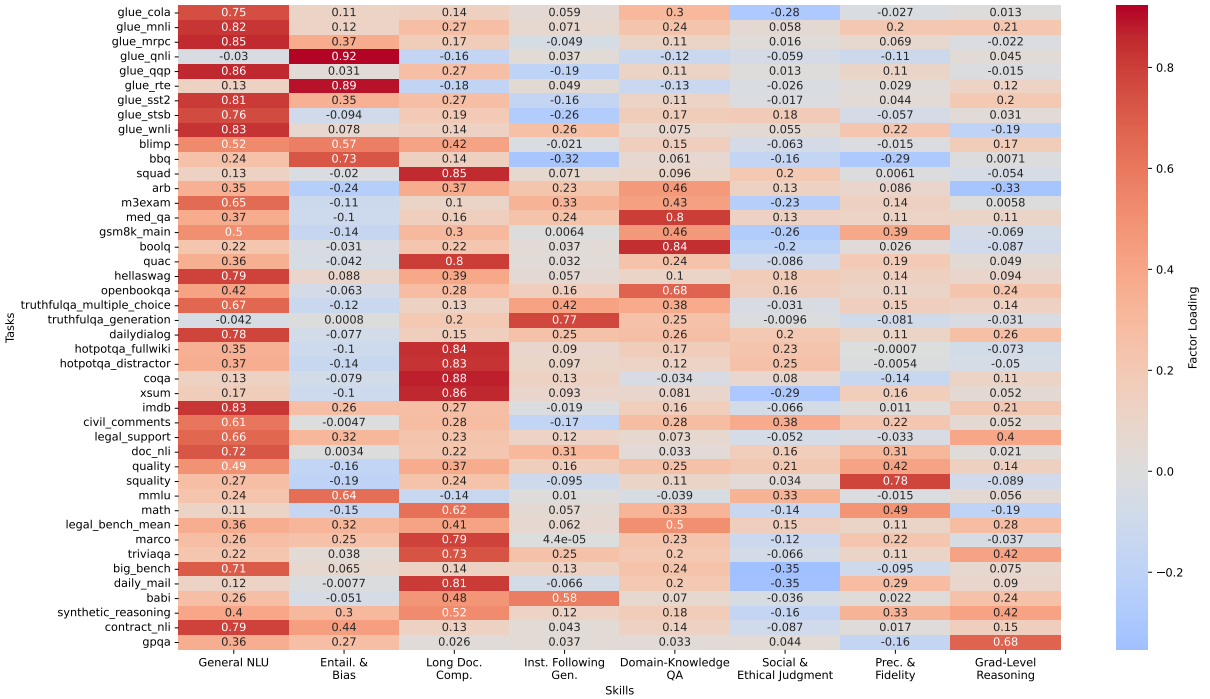


Figure 2: Task–factor loading matrix ($\mathbf{\Lambda}$) for 8 skills. Larger value indicate stronger association; positive values mean the task reflects the skill, negative values suggest an inverse pattern.

4.2 The Skill-Based Leaderboard

To evaluate models along these dimensions, we construct a skill-based leaderboard from the score matrix $\mathbf{\Theta}$ (model–skill relationships; Fig. 3). Unlike traditional task-level leaderboards, which report average performance across benchmarks, this representation ranks models by skill profiles, offering a far more compact, structured view of abilities.

4.3 The Analysis of Latent Skills

We analyze the uncovered structure by examining how tasks align or separate in the learned skill space. In particular, we focus on cases where tasks that appear similar by format or domain diverge across factors, as well as cases where superficially different tasks converge on the same dimension.

Surface-similar tasks diverge. First, let us consider pairs of tasks that share a format yet fall on different factors. For example, GPQA and TriviaQA are both defined as question-answering tasks, but GPQA loads on Factor 8. GRAD.-LEVEL REASONING, because it stresses physics derivations, whereas TriviaQA lands on Factor 3. LONG DOC. COMP., wherein evidence extraction dominates; it appears that the *skills* needed for question-answering differ between these datasets. A similar split appears inside GLUE: MRPC (paraphrase detection) sits with Factor 1's GENERAL NLU, but its sibling QNLI (question–sentence entailment) migrates to Factor 2, reflecting the finer-grained inference needed for entailment.

Generative tasks diverge as well: CoQA (conversational, document-grounded QA) joins Factor 3 LONG DOC. COMP., while TruthfulQA-Generation, whose main challenge is hallucination avoidance, aligns with Factor 4's INST.-FOLLOWING GEN. Even two entailment datasets separate — GLUE's MNLI stays in Factor 1, but QNLI heads to Factor 2 — highlighting how increased reasoning granularity reshapes the latent map.

Surface-dissimilar tasks converge. The reverse pattern also appears: tasks with different formats or domains can load on the same factor due to shared underlying demands. For example, SQuALity (faithful-summary scoring) and Math (symbolic arithmetic) both rely on token-level exactness; unsurprisingly, they co-load on Factor 7, our PREC. & FIDELITY axis. SQuAD and XSum, extractive versus abstractive summarization, meet in Factor 3 because both demand document comprehension and content selection. Finally, news-domain MRPC and legal Contract-NLI share Factor 1, GENERAL NLU, suggesting that their relative simplicity (and high model scores) overrides domain differences.

Factor Name	Factor Nickname	Description	Strong + tasks (loading)	Strong − tasks (loading)
1. General NLU	GENERAL NLU	Everyday syntax, semantics, sentiment classification.	QQP (.86), MRPC (.85), WNLI (.83), IMDB (.83), MNLI (.82), SST-2 (.81), HellaSwag (.79), DailyDialog (.78), Contract-NLI (.78)	
2. Fine-Grained Entailment & Token Bias	ENTAILMENT & BIAS	Short entailment and token-level bias/grammar probes.	QNLI (.92), RTE (.89), BBQ (.73), MMLU (.64), BLiMP (.57)	
3. Document Reading, Retrieval & Summarization	LONG DOC. COMPREHENSION	Long-context reading, retrieval, and abstractive summarization.	CoQA (.88), XSum (.86), SQuAD (.85), Hotpot-FW (.84), Hotpot-Distr. (.83), CNN/DM (.81), QuAC (.80), MS MARCO (.79)	
4. Truthful Instruction Generation	INST. FOLLOWING GEN.	Truthful, instruction-following open-ended generation.	TruthfulQA-Gen (.77), bAbI (.58), M3Exam (.33)	BBQ (–.32)
5. Specialized Domain-Knowledge QA	DOMAIN QA	Expert factual QA in medical, legal, scientific domains.	BoolQ (.84), MedQA (.80), OpenBookQA (.68), LegalBench (.50), GSM8K (.46)	
6. Social / Ethical Judgment	SOCIAL / ETHICAL JUDGMENT	Social and ethical judgment vs. news-style summarization.	CivilComments (.38), MMLU (.33)	BigBench (–.35), DailyMail (–.34)
7. Faithful Summarization & Quantitative Precision	PREC. & FIDELITY	Token-level fidelity and quantitative precision.	SQuALity (.78), Math (.49), Quality(.41), GSM8K (.39), Synth-Reason (.33)	BBQ (–.29)
8. Grad-Level Scientific and Legal Reasoning	GRAD-LEVEL REASONING	Graduate-level scientific and legal reasoning.	GPQA (.68), Synth-Reason (.42), TriviaQA (.42), Legal-Support (.40)	ARB (–.33)

Table 1: Latent skill factors, with names, short descriptors, and the most strongly associated task loadings.

Taken together, these examples illustrate that the latent factors organize tasks according to shared behavioral demands rather than surface characteristics such as format or domain.

Breadth and interpretability of the latent skills. As summarized in Table 1, the eight recovered skills differ in both breadth and diagnostic coverage across benchmarks. Some, such as GENERAL NLU and DOMAIN QA, integrate a broad family of tasks, whereas others — e.g., PREC. & FIDELITY and ETHICAL JUDGMENT — are defined by a smaller set of highly discriminative tasks that capture specialized competencies. Negative loadings are expected within such latent dimensions: they indicate inverse co-variation, where excelling on one group of tasks (e.g., factual precision) often accompanies lower performance on another emphasizing complementary demands (e.g., stylistic alignment or social sensitivity). Hence, the sign of a loading indicates relative emphasis along a shared latent dimension, not whether a model is better or worse overall.

5 Evaluation of the Uncovered Skill Set

In this section, we aim to evaluate the validity and interpretability of the uncovered skill dimensions. To this end, we address two key questions:

(i) Does the factor structure reflect a stable and internally coherent latent space?

(ii) Do the derived skill scores align with external (human) judgments of model behavior?

To answer (i), §5.1 draws on standard FA metrics (Harman, 1976; Cronbach and Meehl, 1955), measuring internal consistency, stability under task perturbation, and factor specificity (uniqueness, outlier detection). We also compare our results using the PAF method to results using PCA for dimension redundancy. To address (ii), §5.2 compares our skill-based rankings to pairwise human preferences from Chatbot Arena (HuggingFace, 2024), testing whether aggregated human judgments align with models' inferred skills.

5.1 Latent Space Validation and Reliability

We evaluated the inherent quality of the identified skills by assessing the latent skill space's coverage (uniqueness) and internal consistency (reliability).

Uniqueness. Uniqueness ($u_j^2 = 1 - h_j^2$) quantifies how much of a task's variance is *not* captured by latent factors. Low values show the skill space adequately explains task behavior, i.e., tasks are well-covered by the factor model. All tasks in our eight-factor solution (Fig. 5 inappendix F) sit below the $\sim$0.40 threshold (max: MMLU 0.38), indicating strong coverage (Costello and Osborne, 2005).

Reliability. To confirm that tasks grouped under each factor measure the same underlying skill, we assessed their *internal consistency* — a standard proxy for reliability. For each factor, we selected tasks with relatively high loadings (z-score > 0.8; minimum four tasks). Using the resulting model-score matrix (models × tasks), we computed inter-task correlations and evaluated consistency via Cronbach's α (Cronbach, 1951) and McDonald's ω (McDonald, 1999).[3] Both statistics ex-

[3]Cronbach's α assumes equal contributions (tau-equivalence) of all items to the latent factor, while McDon-

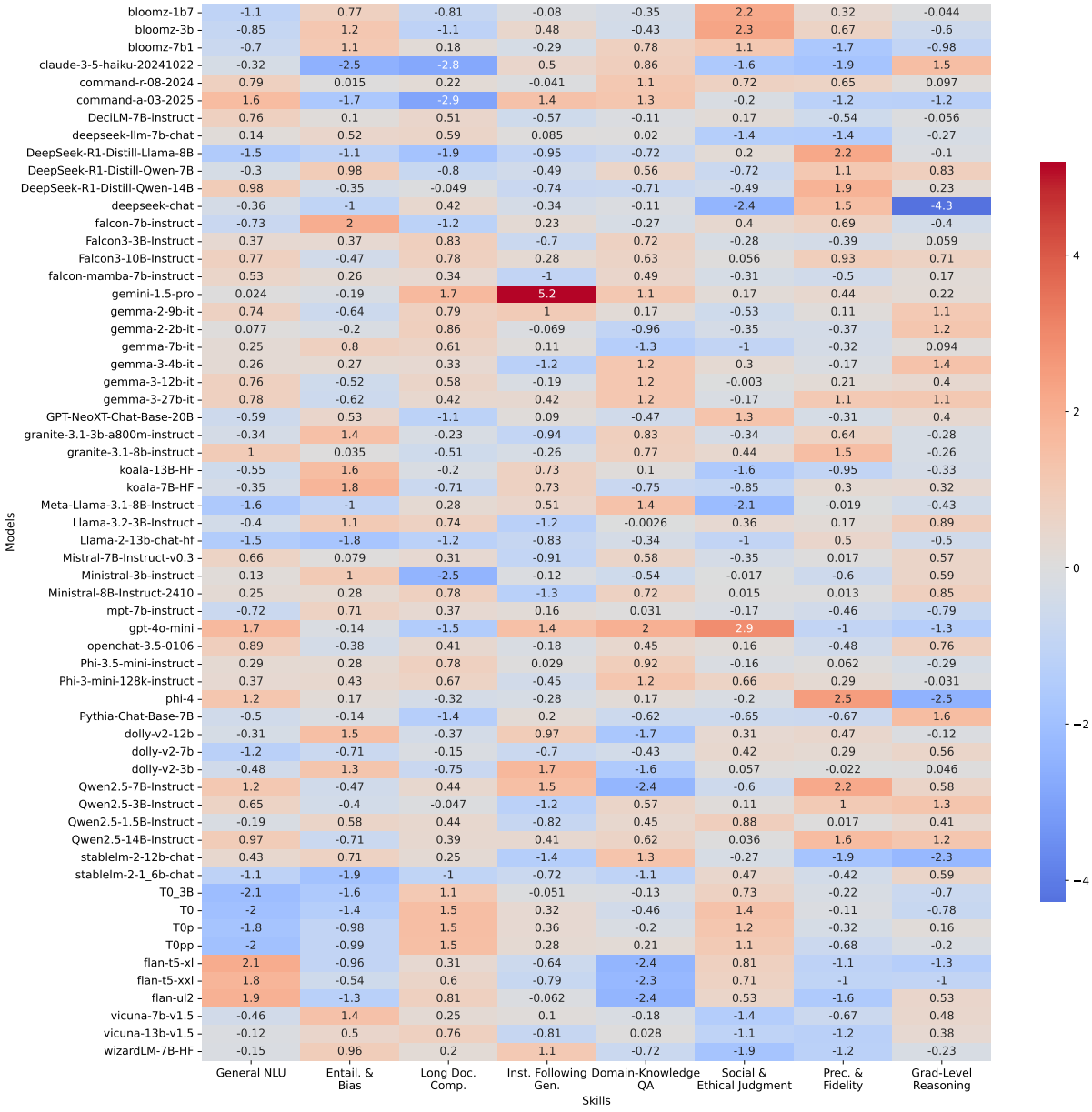


Figure 3: Model scores on 8 latent skills. Higher values indicate stronger skill performance; positive = above average, negative = below average.

ceeded the conventional 0.80 threshold ($\alpha = 0.836$, $\omega = 0.821$), showing models behave consistently across tasks within each factor — evidence that the factors are **reliable** measures of latent skills.

Robustness of the latent-skill space. In this section we verify that the eight-factor solution is not a fragile artifact of any single modeling choice. To this end, we apply three stress tests: (i) **factor dimensionality** — refitting the model at $k = 6, 7$ (under-extraction) and $k = 9, 10$ (over-extraction); (ii) **model subsampling** — five runs in which 30 % of the LMs were held out at random; (iii) **task removal** — leave-one-task-out in the PAF fit.

We aligned each perturbed loading matrix to the reference solution using orthogonal procrustes analysis (Schönemann, 1966), and deviation was measured with principal angles.[4] Tucker's congruence coefficient (φ) provided a complementary similarity check.[5] We expect high congruence coefficient

ald's ω accounts for differing loadings and is generally more reliable in factor models (McDonald, 1999).

[4]Principal angles quantify the minimal angular deviation between corresponding subspaces (A and B) of two factor solutions, defined by $\cos(\psi) = \max_{u \in A, v \in B} \frac{u^\top v}{\|u\| \|v\|}$. Values $\psi \leq 25^\circ$ suggest strong factor similarity (Tucker, 1951).

[5]Tucker's congruence coefficient φ measures the cosine

and low principal angles if the factor space after each ablation is similar to the original one.

Across perturbation analyses, two complementary findings emerge: varying the number of factors confirms that the eight-factor model captures the data without over- or under-fitting, while removing data (models or tasks) shows that this space is remarkably stable. Specifically, under-extraction ($k = 6, 7$) caused skill merging or loss (principal angles $43° - 60°$), whereas over-extraction ($k = 9, 10$) produced redundant, near-orthogonal axes while the eight core factors rotated by at most $26°$. When 30% of models were removed, every factor re-emerged with low alignment error ($< 14°$, s.d.$< 8°$), and leave-one-task-out runs yielded minimal rotation ($\leq 2.4°$). Together, these results confirm that the eight-factor solution captures a set of distinct, stable capabilities and remains robust to realistic data perturbations.

Together, these results confirm that the eight-factor solution captures a set of distinct, stable capabilities and remains robust to data perturbations.

Models outlier detection. To verify that all current models lie within the latent-skill space, we also computed Mahalanobis distances[6] (Mahalanobis, 1936) from the distribution of other models; none exceeded the 99.5th percentile of the χ^2_B distribution (Mclachlan, 1999), indicating no outliers. Overall, the skill space is structurally stable under design choices, without statistical outliers.

PCA vs. PAF Principal Axis Factoring (PAF), the method we use, and Principal Component Analysis (PCA), a viable alternative, differ fundamentally in what they model: PAF explicitly isolates *shared variance* across tasks, whereas PCA maximizes *total variance* and does not distinguish shared structure from task-specific effects. To assess whether this distinction matters empirically, we repeated all robustness, interpretability, and downstream analyses using PCA. The results show that PCA yields a substantively different and less useful latent representation (Table 2).

First, PCA recovers a different latent representation than PAF. Beyond the strongest global direction, alignment between PCA components and PAF factors is limited (Tucker congruence $\varphi = 0.29 - 0.93$), and model rankings along corresponding dimensions differ substantially, as evidenced by only moderate correlation between PCA- and PAF-derived model scores (Spearman $r = 0.49 - 0.73$). Furthermore, this difference is reflected in external validation. As will be seen shortly in §5.2, the aggregate PAF skill score aligns strongly with human preference judgments in Chatbot Arena ($\rho = 0.73, p < 0.005$), whereas the aggregate PCA score shows no reliable correlation ($\rho = 0.28, p > 0.3$). We interpret this gap as a consequence of PCA's objective: by maximizing total variance, PCA mixes shared performance structure with task-specific variation, which limits its ability to recover the latent capabilities that consistently influence human preference judgments.

Second, these differences reflect PCA's objective rather than genuine recovery of shared task structure. Because PCA maximizes total variance, its components receive small contributions from many tasks, resulting in diffuse loadings. Consequently, removing individual tasks or subsampling models has little effect on the solution. This insensitivity reflects PCA's focus on global variance directions, not the identification of latent dimensions supported by consistent task co-variation. Third, PCA components are less interpretable and less useful for evaluation. Several PCA components are supported by a single dominant task, making standard reliability measures (Cronbach's α, McDonald's ω) undefined, whereas PAF factors consistently exhibit high internal consistency ($\alpha, \omega > 0.80$).

Finally, when we apply the same downstream reconstruction and prediction pipelines we used for task novelty assessment, new-model profiling, and model selection (§6) to PCA components, rather than PAF factors, performance degrades substantially: reconstruction error increases and predictions for unseen models or tasks become unreliable.

Taken together, this comparison shows that while PCA identifies directions of maximal variance, it conflates shared and task-specific effects and yields latent dimensions that are less interpretable, reliable, and predictive. PAF more directly recovers shared performance structure across tasks, justifying its use as a principled latent-skill model.

5.2 Comparison with Human Preferences

Chatbot Arena (HuggingFace, 2024) provides large-scale human judgments and a single global

similarity between two factor-loading vectors ℓ_a and ℓ_b: $\varphi = \frac{\sum_j \ell_{a,j}\ell_{b,j}}{\sqrt{\sum_j \ell^2_{a,j}}\sqrt{\sum_j \ell^2_{b,j}}}$.

[6] an 'elliptical' metric accounting for feature covariance, is defined as $d_M(\boldsymbol{\phi}) = \sqrt{(\boldsymbol{\phi} - \bar{\boldsymbol{\phi}})^\top \Sigma^{-1} (\boldsymbol{\phi} - \bar{\boldsymbol{\phi}})}$, where $\bar{\boldsymbol{\phi}}$ and Σ are the sample mean and covariance.

Criterion	PAF (ours)	PCA
Variance treatment	Models shared variance and estimates task-specific variance	Maximizes total variance without separating task-specific effects
Alignment with PAF factor structure	Reference latent structure	Limited alignment beyond first component (Tucker $\varphi = 0.29$–0.93)
Agreement in model rankings (PAF vs. PCA)	Stable across re-fits	Moderate agreement with PAF ($r = 0.49$–0.73)
Loading concentration	Sparse, multi-task loadings defining each factor	Broad loadings spread across many tasks
Reliability (α, ω)	High; multi-task latent skills	Often undefined (components dominated by a single task)
Structural sensitivity (resampling & dimensionality)	Meaningful factor merging/splitting and interpretable changes	Minimal change due to variance maximization
Interpretability	Shared latent capabilities	Maximal variance directions
Correlation reconstruction	Lower error	Higher error
Arena alignment	Strong ($\rho = 0.73$, $p < 0.005$)	None ($\rho = 0.28$, $p > 0.3$)

Table 2: Comparison of PAF and PCA across alignment, robustness, interpretability, and external validation criteria.

Elo score (Elo, 1978)[7] per model, and is often treated as a proxy for overall quality. We therefore use Arena as an external reference for validating our multidimensional skill scores. We find that while aggregating out skill scores aligns strongly with Arena Elo, individual skills contribute unevenly to this alignment — showing that Arena reflects only a subset of model skills. As a result, models with similar Elo scores can differ substantially in underlying strengths, motivating skill-aware evaluation beyond a single global preference.

Overall correlation. To test whether our skill space captures what Arena raters reward at a global level, we correlate Arena Elo ranking with the average of our eight latent skill scores for the 13 models shared across both settings. This yields a strong Spearman correlation ($\rho = 0.73, p < 0.005$). A leave-one-out analysis over 12-model subsets gives $\rho = 0.722 \pm 0.058$, indicating that the agreement is stable under small changes in the model set.

Task-level baselines. To contextualize this alignment, we evaluate task-level baselines. A simple average of task scores correlates strongly with Arena Elo ($\rho \approx 0.86$), but this alignment is driven by a narrow tasks subset. The most Arena-aligned task, `HotpotQA-FullWiki`, achieves near-perfect correlation ($\rho \approx 0.97$), and averaging the top-5 Arena-correlated tasks yields a similarly strong baseline.

[7]Adapted from competitive games, this dynamic rating system updates based on blind A/B test outcomes.

These results indicate that Arena preferences are heavily influenced by specific interaction styles—particularly multi-hop QA and short-form reasoning—rather than reflecting a balanced assessment of model capabilities. While task-level aggregation can reproduce Arena rankings, it relies disproportionately on a small subset of highly Arena-aligned tasks.

By contrast, the latent-skill framework groups tasks into stable dimensions that capture shared behavioral structure. This abstraction reveals both where human preferences align with underlying capabilities and where they diverge, yielding a more interpretable and generalizable evaluation.

Skill–level alignment. Although the aggregate skill score correlates with Arena Elo, individual skills exhibit systematically different relationships with human preference judgments (Table 4 in appendix G). To characterize these differences, we analyze both each skill's-Arena correlation, and the correlations of its highest-loading tasks. This reveals 4 patterns.

(1) *Arena-rewarded skills* (Case A), such as GENERAL NLU and DOMAIN QA, show strong positive correlations, and their highest-loading tasks are also Arena-aligned. This suggests that broad comprehension and confident factual recall are consistently salient to raters.

(2) *Under-recognized skills* (Case B), including PREC. & FIDELITY and GRAD-LEVEL REASONING, exhibit weak skill-level correlation despite high in-

ternal consistency. Tasks loading on these skills emphasize numerical exactness, reference faithfulness, and multi-step reasoning, and show heterogeneous Arena alignment—some correlating strongly (e.g., GSM8K), others weakly (e.g., GPQA)—yet they co-vary reliably across models. As shown in §5.1, these skills are internally coherent ($\alpha > 0.85$, $\omega > 0.80$). Their weak Arena alignment therefore reflects a limitation of preference-based evaluation, which favors fluency, brevity, and apparent confidence over capabilities that require careful verification, such as factual precision or deep reasoning.

(3) *Context-dependent skills* (Case C), such as LONG-DOCUMENT COMPREHENSION, INSTRUCTION FOLLOWING / GENERATION, and SOCIAL–ETHICAL JUDGMENT, are rewarded only when Arena prompts elicit the relevant capability. For example, LONG-DOCUMENT COMPREHENSION includes `HotpotQA-FullWiki`, which correlates strongly with Arena, but also aggregates long-context and summarization tasks that are rarely exercised in Arena-style interactions. Reduced skill-level alignment thus reflects limited prompt coverage in preference comparisons, rather than insensitivity of human raters to these capabilities.

(4) *Arena-penalized* skills (Case D), such as ENTAILMENT & BIAS, emphasize formal entailment and bias avoidance, often yielding cautious or rigid responses that are often judged as less helpful in preference-based comparisons.

Taken together, these results show that Arena reflects only a subset of the latent skill space. Human preferences consistently reward general comprehension and confident recall, while underweighting or penalizing deep reasoning, factual precision, and bias mitigation. Skill-based evaluation therefore complements preference-based metrics by explaining which capabilities are visible, overlooked, or penalized under current paradigms.

Illustrative cases. To illustrate these effects, Fig. 1 compares two models with nearly identical Arena Elo scores. Despite similar rankings, the models differ substantially across latent skills, revealing complementary strengths invisible to aggregate evaluation. This example shows how skill-based analysis enables more informed model selection than reliance on a single global score.

6 Applications: Skill-Based Evaluation

Beyond descriptive analysis, the latent skill space enables practical tools that address concrete evaluation challenges faced by LLM developers and users. Rather than asking how a model performs on dozens of benchmarks, our framework supports targeted questions such as: *Is this new benchmark actually novel?*, *How can I characterize a new model without running a full evaluation suite?*, and *Which existing model is best suited for a new task?*

We instantiate this perspective through three applications: (i) diagnosing whether a new dataset adds genuine signal or largely re-measures existing skills; (ii) profiling an unseen model from a small, diverse subset of tasks; and (iii) selecting the best model for a new task with minimal evaluation cost. Each application is validated on held-out data. All tools rely on the same structural property: the model×task performance matrix is well-approximated by a low-dimensional skill space. This low-rank structure makes it possible to infer missing entries reliably. Estimating a new model corresponds to *row completion* (inferring a model's skill profile from a few task scores). Analyzing a new task corresponds to *column completion* (inferring task loadings from a few model evaluations).[8]

6.1 Is this Task Redundant or Novel?

The growing number of evaluation datasets raises the question whether a new task adds genuinely new information or merely re-measures existing skills. To answer this, we place each task on a *novel–redundant* continuum using 3 diagnostics:

(1) Raw Score Redundancy: We test if a task's performance vector is predictable from others. High collinearity (max Pearson $r > 0.90$) or regression $R^2 > 0.80$ signals redundancy (Kutner et al., 2005).

(2) Alignment with latent skill space: We assess how well a task is represented by existing factors by: (i) explained variance change when adding the task ($< 1\%$ suggests redundancy) (Harman, 1976; Fabrigar et al., 1999), (ii) reconstruction error of the task's scores from the factor model (MSE > 0.10 indicates novelty) (Jolliffe, 2002), and (iii) factor loadings (uniformly low < 0.3 imply poor alignment) (Comrey and Lee, 1992). (3) Structural influence: We compare PAF solutions with and without the new task. Minimal angular shift ($< 15°$) indicates the structure is unaffected, suggesting redundancy (Björck and Golub, 1973; Absil et al., 2006).

[8] All applications depend on the shared-variance structure recovered by PAF; analogous pipelines using PCA yield substantially higher error and instability (§5.1).

To compute the **task novelty-redundancy score**, we thus average the different diagnostics (+1 for novelty, −1 for redundancy) into a continuous score ranging [-1, 1], situating each task on a novel-to-redundant continuum for richer assessment rather than a simple categorical label.

Experimental settings. We apply a *leave-one-task-out* procedure, re-fitting the PAF model without each task and recomputing all criteria. This revealed a clear spectrum of task contributions.

At the **redundant end**, we find tasks like `glue_sst2`, `hellaswag`, and `imdb`, whose performance is highly predictable from other tasks with minimal structural impact. At the **novel end**, we find tasks that inject distinct signal or sharpen underrepresented skill axes. These cluster into themes such as bias probes (`BLiMP`), safety tests (`TruthfulQA`), and domain-specific depth (`GPQA`).

Relation to robustness. While in §5.1 we showed that the eight-factor structure itself is stable, this analysis quantifies how much each task sharpens or enriches that structure. Novel tasks do not induce new latent dimensions; rather, they contribute distinctive signal within the existing skill space, improving its resolution along specific axes. Alternative diagnostics (e.g., leave-one-skill-out) may further distinguish new skills from noise.

6.2 How To Efficiently Evaluate New LLMs?

Evaluating a new LLM typically requires running it across a broad suite of tasks, which is costly and time-consuming. Instead, we estimate a model's full skill profile from a small, diverse task subset. Our 3-stage pipeline is:

(1) Sample a diverse task set: We select k=12 tasks (corresponding to $k \approx 1.5\,C$ (James et al., 2013)) with the highest communalities to maximize latent-skill coverage. The new model is run only on this subset, yielding a performance vector $p_k \in \mathbb{R}^k$.

(2) Estimate full task performance: Using the k observed scores and loadings of the chosen tasks $\Lambda_k \in \mathbb{R}^{k \times C}$ computed based on the training data, we estimate the model's factor scores $\hat{\theta} \in \mathbb{R}^C$ via least squares: $\hat{\boldsymbol{\theta}} = (\Lambda_k^\top \Lambda_k)^{-1} \Lambda_k^\top \mathbf{p}_k$.

We then reconstruct the model's full performance with: $\hat{\boldsymbol{\phi}}_{\text{model}} = \Lambda \hat{\boldsymbol{\theta}}$.

(3) Diagnose model behavior: We assess whether the model conforms to the learned skill space by: (i) applying the Mahalanobis-distance test (§5.1) to detect outliers, (ii) evaluating reconstruction error (flagging cases where MSE exceeds $\mu + \sigma$) (Hawkins, 1980), and (iii) analyzing the inferred skill profile $\hat{\boldsymbol{\theta}}$ for comparative evaluation across models (Thurstone, 1947). This approach yields a low-cost yet informative skill profile.

Experimental settings. To demonstrate this pipeline on unseen models, we perform a *leave-one-model-out* evaluation over 20 held-out LLMs. Each is evaluated on top-k tasks, full profile reconstructed, and diagnostics computed. All models passed the outlier test (§5.1); only `bloomz-1b7` exceeded the MSE threshold, while overall reconstruction error remained low (mean MSE = 0.095) . [9]

We illustrate two characteristic profiles (Figure 9: `Flan-T5-XL` peaks on Factor 1 (GENERAL NLU) but underperforms on Factors 2, 5, 7, 8 (ENTAILMENT, DOMAIN QA, LONG-DOC. COMP., GRAD-LEVEL REAS.), matching its C4+Flan training (Fig. 9a in appendix H). `OpenChat-3.5` excels on Factors 1 and 8 but lags on factors 2 and 7, reflecting known RLHF-era tradeoffs (Fig. 9b in appendix H). Overall, `Flan-T5-XL` favors instruction-following with limited domain depth, whereas `OpenChat` trades fine-grained rigor for recall and academic reasoning.

6.3 Which Model Is Optimal for my Case?

Selecting the optimal model for a new task often demands costly evaluations across many candidates. Using the same low-rank structure, we instead infer a task's skill requirements from a small, representative subset of models and predict performance for the rest. We proceed in three steps:

(1) Sample a diverse model set: We select k=12 ($k \approx 1.5\,C$ (James et al., 2013)) models that maximize latent-space coverage via a Max–Min diversity sampler[10] (Berger-Wolf and Saia, 2007). Only these k models are run on the new task.

(2) Estimate task loadings: Given a new task with evaluation scores $\phi_{\text{task}} \in \mathbb{R}^k$ from the k chosen models, and the corresponding factor scores $\hat{\boldsymbol{\Theta}}_k \in \mathbb{R}^{k \times C}$, we estimate the new task's loading vector $\hat{\boldsymbol{\lambda}} \in \mathbb{R}^C$.

(3) Predict the rest: We predict performances on the new task for all remaining models, rank by the

[9] `bloomz-1b7` differs substantially in scale and training regime from the instruction-tuned models used to estimate the latent skill space. As a result, its performance profile lies outside the empirical distribution on which the factor model was learned, where higher reconstruction error is expected.

[10] greedily maximizes the smallest pairwise distance (often giving better coverage than k-means centers)

resulting $\hat{\phi}_{\text{task}}$ and select the top candidate.

Experimental settings. We use a *leave-one-task-out* protocol across all B tasks. For each held-out task, we fit a PAF model on the remaining $B-1$ tasks, evaluate k selected models , estimate factor loadings, and predict the remaining $M-k$ models. We compute MSE and Pearson r between predicted and actual scores on the held-out models, and average across tasks. This yields a mean normalized MSE below 0.20 and a mean Pearson correlation $r > 0.85$ for well-represented tasks (e.g., GLUE_RTE, BOOLQ, COQA); higher errors on underrepresented tasks (e.g., BABI) highlight gaps in the latent skill map (Muennighoff et al., 2023).

With only k pilot evaluations and a fixed skill basis, this approach efficiently shortlists strong candidate models for new tasks within or near the existing skill space.[11] These results show that model performance can be predicted from a small subset.

7 Related and Future Work

Recent work has begun to analyze LLM evaluation benchmarks through latent-variable models, moving beyond aggregate leaderboard averages to uncover shared structure across tasks. Burnell et al. (2023) applied maximum-likelihood factor analysis (MLFA) to a small, homogeneous set of classification and short-form reasoning tasks, identifying three broad factors interpreted as linguistic understanding, reasoning, and memorization. Ilić and Gignac (2024) extended this psychometric framing to a leaderboard-scale Bayesian factor model, emphasizing a dominant general-ability (g) factor with a small number of secondary components.

These studies establish that low-dimensional latent structure exists in task performance, but differ from our work in both scope and emphasis. They analyze more limited task collections and focus primarily on identifying broad or hierarchical ability factors, without systematically testing factor stability under task or model perturbations, or exploring downstream evaluation use cases. A related line of work models benchmark performance using linear latent-skill assumptions to study scaling laws and extrapolation (Polo et al., 2025). While effective for prediction, these approaches assume proportional skill transfer across tasks and do not aim to recover interpretable or stable skill dimensions.

Our work builds on these foundations by applying PAF to a substantially larger and more diverse model–task matrix (60 models × 44 tasks), explicitly isolating shared covariance and validating the resulting latent space under task removal, model subsampling, and dimensionality variation. Rather than centering on a single general factor or human cognitive taxonomy, we recover a finer-grained set of stable, data-driven skill dimensions and anchor them to human preferences using Chatbot Arena.

Our recovered skills refine earlier findings: broad dimensions such as 'language understanding' or 'reasoning' in prior work are decomposed into multiple distinct capabilities (e.g., general NLU, long-doc. comp.) that behave differently across tasks, models, and human evaluations. Where earlier studies establish latent structure, our contribution characterizes it at scale, validates robustness, and supports practical evaluation workflows.

A natural direction is item-level analysis, which may reveal finer-grained skill structure, enable minimal representative subsets, and assess sensitivity to instance selection within benchmarks. We aim to extend the framework to multilingual and multimodal benchmarks, where skill structure may differ and yield new insights into model capabilities.

8 Conclusion

This study introduces a psychometric framework for evaluating LLMs, based on factor analysis of model performance across diverse tasks (Fig. 3). By uncovering latent structure in the task performance matrix, our approach identifies eight skills underlying model behavior, ranging from general NLI to instruction-following to domain reasoning. This moves beyond single-score metrics, offering a more compact, interpretable, and multifaceted view of LLM abilities. These skills remain stable across model and task perturbations, capturing a robust structure that supports consistent evaluation and generalization to new tasks and models. We develop suite of applications to: assess the novelty of tasks, project new models into the skill space, and efficiently select the best model for a given use case. To support adoption of this methodology, we release all data and code publicly. Our ultimate goal is to encourage a shift from single-score leaderboards to a more transparent, and practical, skill-based LLM evaluation.

[11] Our final two applications resemble coreset selection (Mirzasoleiman et al., 2020; Feldman, 2020), but target efficient inference in latent skill space. We release code for easy reuse with new benchmarks and models.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work has been presented at various seminars and colloquia: the BIU-NLP meeting, Technion CS NLP team meeting, the HUJI NLP Seminar. We thank all participants for comments and fruitful discussion. This research has been funded by the following funding agencies: a grant from the Israeli Science Foundation (ISF grant 670/23), a KAMIN grant by the Israeli Innovation Authority (IIA), and a VATAT grant by the the Israeli Planning and Budgeting Committee (PBC), for which we are grateful.

References

Pierre-Antoine Absil, Alan Edelman, and Plamen Koev. 2006. On the largest principal angle between random subspaces. *Linear Algebra and its Applications*, 414(1):288–294.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Tanya Y. Berger-Wolf and Jared Saia. 2007. Maxmin diversity. *Computational Geometry*, 36:67–85.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, New York.

Åke Björck and Gene H. Golub. 1973. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification.

Ryan Burnell, Han Hao, Andrew R. A. Conway, and Jose Hernandez Orallo. 2023. Revealing the structure of language model capabilities.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac : Question answering in context.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Andrew L. Comrey and Howard B. Lee. 1992. *A First Course in Factor Analysis*, 2nd edition. Lawrence Erlbaum Associates, Hillsdale, NJ.

Paul T. Costa and Robert R. McCrae. 1992. *Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO-FFI) Professional Manual*. Psychological Assessment Resources, Odessa, FL.

Anna B. Costello and Jason W. Osborne. 2005. Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical Assessment, Research, and Evaluation*, 10(7):1–9.

Lee J. Cronbach. 1951. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3):297–334.

Lee J. Cronbach and Paul E. Meehl. 1955. Construct validity in psychological tests. *Psychological Bulletin*, 52(4):281–302.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. 2016. Evaluating prerequisite qualities for learning end-to-end dialog systems.

Arpad E. Elo. 1978. *The Rating of Chessplayers, Past and Present*. Arco Publishing, New York.

Leandre R Fabrigar, Duane T Wegener, Robert C MacCallum, and Erin J Strahan. 1999. Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods*, 4(3):272–299.

Dan Feldman. 2020. Introduction to core-sets: an updated survey.

Clémentine Fourrier, Nathan Habib, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

Gemini-Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models.

Harry H. Harman. 1976. Modern factor analysis. *University of Chicago Press*.

D. M. Hawkins. 1980. Identification of outliers. *Journal of the American Statistical Association*, 75(372):432–439.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend.

HuggingFace. 2024. Hugging face lmsys leaderboard. `https://huggingface.co/spaces/lmarena-ai/chatbot-arena-leaderboard`.

David Ilić and Gilles E. Gignac. 2024. Evidence of interrelated cognitive-like capabilities in large language models: Indications of artificial general intelligence or achievement? *Intelligence*, 106:101858.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Springer.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams.

Ian T Jolliffe. 2002. *Principal Component Analysis*. Springer.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension.

Henry F. Kaiser. 1960. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20(1):141–151.

Yuta Koreeda and Christopher D. Manning. 2021. Contractnli: A dataset for document-level natural language inference for contracts.

Michael H Kutner, Christopher J Nachtsheim, John Neter, and William Li. 2005. *Applied Linear Statistical Models*, 5th edition. McGraw-Hill Education.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher

Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Prasanta Chandra Mahalanobis. 1936. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55.

Roderick P. McDonald. 1999. *Test Theory: A Unified Treatment*. Lawrence Erlbaum, Mahwah, NJ.

G. Mclachlan. 1999. Mahalanobis distance. *Resonance*, 4:20–26.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. 2022. Quality: Question answering with long input texts, yes!

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R. Bowman. 2022. Bbq: A hand-built bias benchmark for question answering.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. 2025. Sloth: scaling laws for llm skills to predict multi-benchmark performance across families.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark.

Tomohiro Sawada, Daniel Paleka, Alexander Havrilla, Pranav Tadepalli, Paula Vidas, Alexander Kranias, John J. Nay, Kshitij Gupta, and Aran Komatsuzaki. 2023. Arb: Advanced reasoning benchmark for large language models.

Peter H. Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin

Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers,

Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Louis L Thurstone. 1947. *Multiple-Factor Analysis*. University of Chicago Press.

Louis Leon Thurstone. 1931. Multiple factor analysis. *Psychological review*, 38(5):406.

Ledyard R. Tucker. 1951. A method for synthesis of factor analysis studies. *Psychometrika*, 16(2):169–192.

Ernest C. Tupes and Raymond E. Christal. 1961. Recurrent personality factors based on trait ratings. Technical Report USAF ASD Technical Report No. 61-97, U.S. Air Force, Lackland Air Force Base, TX.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. Squality: Building a long-document summarization dataset the hard way.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2023. Blimp: The benchmark of linguistic minimal pairs for english.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. Docnli: A large-scale dataset for document-level natural language inference.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence?

Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization.

A principal Axis Factoring

In PAF, we assume the latent-variable factorization

$$\mathbf{\Pi} \;=\; \mathbf{\Theta}\mathbf{\Lambda}^{\top} + \boldsymbol{\epsilon}, \tag{1}$$

so that each row satisfies $\mathbf{p}_i = \boldsymbol{\theta}_i^{\top}\mathbf{\Lambda}^{\top} + \boldsymbol{\varepsilon}_i$.

We further assume the following statistical conditions:

$$\mathbb{E}[\boldsymbol{\theta}_i] = \mathbf{0} \qquad \mathrm{Cov}(\boldsymbol{\theta}_i) = \mathbf{I}_C$$
$$\mathbb{E}[\boldsymbol{\varepsilon}_i] = \mathbf{0} \qquad \mathrm{Cov}(\boldsymbol{\varepsilon}_i) = \mathbf{\Psi}$$
$$\mathbf{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_B) \qquad \mathrm{Cov}(\boldsymbol{\theta}_i, \boldsymbol{\varepsilon}_i) = \mathbf{0}$$

The population covariance (or correlation) matrix of task scores is decomposed as

$$\mathbf{R} \;=\; \mathbf{\Lambda}\mathbf{\Lambda}^{\top} + \mathbf{\Psi},$$

where $\mathbf{\Lambda}\mathbf{\Lambda}^{\top}$ captures the *shared* variance explained by the latent skills and $\mathbf{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_B)$ collects the *unique* (task-specific) variances.

Relation to principal–component analysis (PCA) PCA factorizes the same matrix as $\mathbf{R} = \mathbf{U}\mathbf{D}\mathbf{U}^{\top}$ with orthonormal loadings $\mathbf{U}$ and diagonal eigenvalues $\mathbf{D}$, seeking directions that maximise total variance. Unlike FA, PCA *(i)* does not separate shared from unique variance, *(ii)* requires orthogonality of the components, and *(iii)* identifies a unique loading matrix up to sign. FA, by contrast, models only the off-diagonal structure and therefore allows rotational indeterminacy: any nonsingular $\mathbf{T}$ with $\mathbf{\Lambda}\mathbf{T}$ spans the same latent subspace.

In FA the loading matrix $\mathbf{\Lambda}$ is not identifiable up to *any* non-singular rotation $(\mathbf{\Lambda}, \mathbf{f}) \to (\mathbf{\Lambda}\mathbf{T}, \mathbf{T}^{-1}\mathbf{f})$, so demanding $\mathbf{\Lambda}^{\top}\mathbf{\Lambda} = \mathbf{I}_m$ would merely freeze one arbitrary rotation without improving model fit. PAF aims to determine the *subspace* spanned by the latent loadings. PCA identifies each loading with a concrete direction that maximizes total variance, and rotating that direction will violate optimality. Orthogonality in the ordinary dot-product is also conceptually mismatched: each variable is weighted by its reliability $1/\psi_j$, making the natural metric $\langle \mathbf{a}, \mathbf{b} \rangle_{\Psi^{-1}} = \mathbf{a}^{\top}\mathbf{\Psi}^{-1}\mathbf{b}$, not the Euclidean one. PAF allows non-orthogonal loadings, and later apply Varimax or Promax rotations, which maximize sparsity and yield factors that align with interpretable domain concepts.

Iterative solution algorithm for PAF

PAF estimates the loading matrix $\mathbf{\Lambda}$ and uniquenesses $\mathbf{\Psi}$ by minimizing $\|\mathbf{R} - \mathbf{\Lambda}\mathbf{\Lambda}^{\top} - \mathbf{\Psi}\|_F^2$. The key quantity is the *communality* $h_j^2 = \sum_{c=1}^{C} \lambda_{jc}^2$, the portion of task j's variance explained by the common factors. The procedure alternates between (i) recomputing $\mathbf{\Lambda}$ from a reduced correlation matrix whose diagonal equals the current communalities, and (ii) updating the communalities from the new loadings, until successive updates differ by less than a tolerance ε.

1. **Initial communalities** Set $h_j^{2,(0)}$ to the squared multiple correlation of task j with all others (or simply 1).

2. **Reduced correlation matrix** Replace the diagonal of $\mathbf{R}$ with the current communalities to obtain $\mathbf{R}^{(t)}$.

3. **SVD / eigen-step (loadings update)** Compute the truncated eigen-decomposition

   $$\mathbf{R}^{(t)} = \mathbf{Q}_c\, \mathbf{\Delta}_c\, \mathbf{Q}_c^{\top},$$

   where $\mathbf{Q}_c \in \mathbb{R}^{B \times C}$ contains the eigenvectors associated with the C largest eigenvalues, and $\mathbf{\Delta}_c = \mathrm{diag}(\delta_1, \ldots, \delta_C)$. The updated loading matrix is

   $$\mathbf{\Lambda}^{(t+1)} \;=\; \mathbf{Q}_c\, \mathbf{\Delta}_c^{1/2},$$

   i.e. each retained eigenvector is scaled by the square root of its eigenvalue.

4. **Update communalities** $h_j^{2,(t+1)} = \sum_{c=1}^{C} \lambda_{jc}^{2,(t+1)}$.

5. **Convergence check** Stop when $\max_j |h_j^{2,(t+1)} - h_j^{2,(t)}| < \varepsilon$ (e.g. $\varepsilon = 10^{-4}$); otherwise iterate.

After convergence, set $\mathbf{\Psi} = \mathrm{diag}(1 - h_1^{2,*}, \ldots, 1 - h_B^{2,*})$. Because $\mathbf{\Lambda}$ is only unique up to rotation, an orthogonal (e.g. Varimax) or oblique (e.g. Promax) rotation can be applied to $\mathbf{\Lambda}^*$ to enhance interpretability without altering model fit.

Rotating the loading matrix

After estimating an initial (possibly non-sparse) loading matrix, researchers often apply an additional rotation to enhance interpretability.

- **Orthogonal rotations** preserve factor independence. The Orthomax family maximises $\sum_{c=1}^{C}[\sum_{j=1}^{B} \lambda_{jc}^4 - \gamma B^{-1}(\sum_{j=1}^{B} \lambda_{jc}^2)^2]$. Common

choices:

• *Varimax* ($\gamma = 1$): encourages each factor to load strongly on a small subset of tasks;

• *Quartimax* ($\gamma = 0$): encourages each task to load on as few factors as possible;

• *Equamax* ($\gamma = C/2$).

- **Oblique rotations** allow correlated factors. Promax starts from a Varimax solution, then "bends" each column by raising its entries to a power ($p \approx 3$–4) and re-orthogonalising, yielding a sparse but correlated loading pattern. Other popular oblique criteria include *Direct Oblimin* and *Geomin*.

Assessing the novelty of a candidate task

Suppose a new, z-scored task vector $\mathbf{p}_{\text{cand}} \in \mathbb{R}^M$ (scores of the M models) is proposed. With the skill matrix $\mathbf{\Theta} \in \mathbb{R}^{M \times C}$ and the fitted loadings $\mathbf{\Lambda} \in \mathbb{R}^{B \times C}$ fixed, we evaluate whether the candidate adds *shared* information that is not already captured by the existing B rows of $\mathbf{\Lambda}$.

1. Project the candidate into skill space Obtain its loading vector by least–squares regression on the factor scores (the analogue of eq. (3)):

$$\boldsymbol{\lambda}_{\text{cand}} = (\mathbf{\Theta}^\top \mathbf{\Theta})^{-1} \mathbf{\Theta}^\top \mathbf{p}_{\text{cand}} \quad \in \mathbb{R}^C.$$

2. Compute communality and uniqueness

$$h^2_{\text{cand}} = \|\boldsymbol{\lambda}_{\text{cand}}\|_2^2, \qquad \psi_{\text{cand}} = 1 - h^2_{\text{cand}}.$$

A high h^2_{cand} indicates the task is well explained by the latent skills; a high ψ_{cand} signals mostly idiosyncratic variance.

3. Compare loading profiles Measure cosine similarity in skills space:

$$\rho_j = \frac{\boldsymbol{\lambda}_{\text{cand}}^\top \boldsymbol{\lambda}_j}{\|\boldsymbol{\lambda}_{\text{cand}}\|_2 \, \|\boldsymbol{\lambda}_j\|_2}, \quad j = 1, \ldots, B.$$

If $\max_j \rho_j$ exceeds a threshold (e.g. 0.9) the candidate's pattern is essentially a duplicate of task j.

4. Residual correlation check Compute the residual vector $\mathbf{r} = \mathbf{p}_{\text{cand}} - \mathbf{\Theta}\boldsymbol{\lambda}_{\text{cand}}$ and its correlation with each existing task's residual. Large residual correlations ($|\operatorname{corr}(r, \varepsilon_j)| > 0.2$) suggest that the current factor structure would need to expand to accommodate truly new shared variance.

B Skill Naming and Interpretation Procedure

This appendix describes the procedure used to assign descriptive labels to the latent factors extracted by PAF. The goal of this two-step process is to provide concise, interpretable summaries of each factor's semantic content, without affecting the underlying factor structure or model scores.

After PAF produces C latent factors, we treat each as a skill which models vary on, and assign to it a concise skill label.

Step 1 - Selecting representative tasks. For each latent factor c, we identify the tasks with the highest absolute loadings ($|\lambda_{dc}|$), by computing the z-scored absolute loadings [12] and selecting the tasks where $z \geq 1$ ($\approx$ top 16%). Tasks with $\lambda_{dc} > 0$ are assigned to the positive set; those with $\lambda_{dc} < 0$ are assigned to the negative set.

Step 2 - LLM-assisted labeling. Each factor is labeled by prompting an LLM with short descriptions of its representative tasks identified in Step 1. The model is instructed to propose a concise descriptive name that summarizes the shared capability reflected by these tasks. The prompt does not restrict the label to predefined categories, allowing labels to emerge naturally from task semantics. The authors review and refine the suggested labels for clarity and consistency.

In order to capture what each factor name is as a skill we had the next prompt. "We produced a PAF model where the original data matrix contains the performances of LLMs on a set of tasks (LLMs as the observations and tasks are the variables). The PAF model produced 8 factors we treat as LLM skills. In order to give a skill name to each factor - attached are the strongest loadings for each one of the factors. Additionally, here is a description of each one of the tasks that were evaluated. Can you please take into account both the positive and negative loadings and label each one of the factors as a skill?" [13] In Table 3 are the different tasks descriptions.

[12] $z_{dc} = (|\lambda_{dc}| - \mu_c)/\sigma_c$, each loading's deviation from its factor's mean in STD units.

[13] We verified that varying the labeling prompt (e.g., per-factor vs. all-factors, with or without the word 'skill') produced identical factor interpretations, confirming that the semantic consistency arises from the loadings themselves rather than prompt phrasing.

Interpretation scope. The descriptive names assigned to each latent skill (e.g., "Precision & Fidelity") are interpretive summaries based on the highest-loading tasks. These labels aid exposition but do not affect the underlying factor structure, which is objectively derived from the shared-variance model. The assigned skill names are descriptive summaries intended to aid interpretation; while alternative labels are possible, the underlying factor structure and model scores are unaffected.

The full prompt template and implementation details are provided in appendix C.

C Implementation Details

Factor Analysis. We fit a PAF model using the `factor_analyzer` Python package, extracting $C = 8$ factors with `rotation="varimax"`. The number of factors was selected based on eigenvalue thresholds and explained variance inspection.

Factor Scores. Model-level factor scores were computed using the regression method from `factor_analyzer`. These scores represent each model's strength on every latent skill and are used throughout all downstream applications (e.g., novelty detection, model profiling, skill-aware evaluation).

Factor Labeling. To label each factor as a latent skill, we passed the high-loading task sets through two LLMs—`deepseek-R1` and `gpt-o3-pro`—prompted to propose intuitive skill names based on shared task properties. Labels were selected via manual refinement.

D List of Tasks

Table 3 list the full list of tasks which we used to evaluate the models upon as well as the type of the task: i.e. binary classification, multiple choice classification or generation. LLM indicate LLM as a judge method (Zheng et al., 2023).

E Instruction per Task

In this section, we elaborate the algorithm for constructing the instructions that we used for each one of the tasks on all models.

We adopted a systematic, iterative method for instruction creation. First, a base instruction was generated using Gemini-pro-1.5 (Gemini-Team, 2024) and evaluated on a subset of 50 labeled examples across 10 diverse models. An instruction was accepted for a model if at least 70% of outputs were valid. If not, we refined the instruction using targeted prompts to rephrase it for better clarity, and re-evaluated it across all models to maintain broad applicability. This iterative refinement continued until the instruction achieved at least 70% valid outputs for all 10 models. Subsequently, the instruction was further validated on additional models, retaining only those models with valid outputs for at least 50% of the examples. Our algorithm for creating such instructions are detailed in Algorithm 1.

Algorithm 1 Iterative Instruction Refinement

Require: Dataset D, base instruction I_0, models M, subset $M_s = \{M_1, \ldots, M_{10}\} \subset M$, subset $S \subset D$, threshold $\tau = 70\%$. PROMPT(I,S) = rephrasing prompt

Ensure: Refined instruction I^*, final model list $M_{\text{final}} \subseteq M$.

Initialize $I \leftarrow I_0$, $M_{\text{final}} \leftarrow \emptyset$, randomize M_s.
Call REFINEINSTRUCTION(I, M_s, S, τ).
Validate I on additional models; include in M_{final} if $V_i \geq 50\%$.
Return $I^* \leftarrow I$, M_{final}.
procedure REFINEINSTRUCTION(I, M_s, S, τ)
    for each $M_i \in M_s$ **do**
        Evaluate S using I on M_i; compute validity rate V_i.
        if $V_i < \tau$ **then**
            Refine I using PROMPT(I,S); update $I \leftarrow I'$.
            for each $M_j \in M_{\text{final}}$ **do**
                Re-evaluate S using I on M_j; update V_j.
                if $V_j < \tau$ **then**
                    Call REFINEINSTRUCTION(I, M_s, S, τ) ▷ Recursive step
            Call REFINEINSTRUCTION(I, M_s, S, τ) ▷ Recursive step
        else
            Add M_i to M_{final}.

F Robustness, Factor Diagnostics

Number of factors. In Fig. 4 is the cumulative explained variance that we used to detect the optimal number of latent factors.

Uniqueness. In Fig. 5 we present the uniqueness, which quantifies how much of a task's variance is

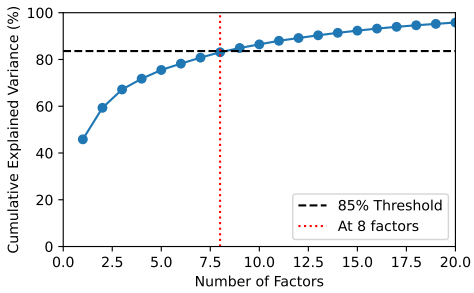


Figure 4: Cumulative explained variance to choose optimal number of factors.

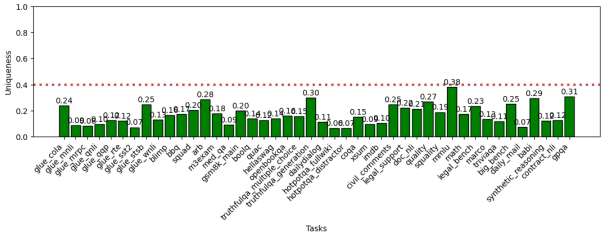


Figure 5: Uniqueness of each task

not captured by the latent factors.

Robustness to factor dimensionality. Fig. 6 contrasts the eight-factor reference with models fitted at $k \pm 1, 2$ factors, utilizing principal angles and Tucker's congruence coefficient ($\varphi \geq 0.90$ for strong similarity (Tucker, 1951)).

Robustness to model subsampling. Fig. 7 shows the mean and std values of the principle angles as well as tucker angles between PAF model produced from 70% of the models and the full PAF model.

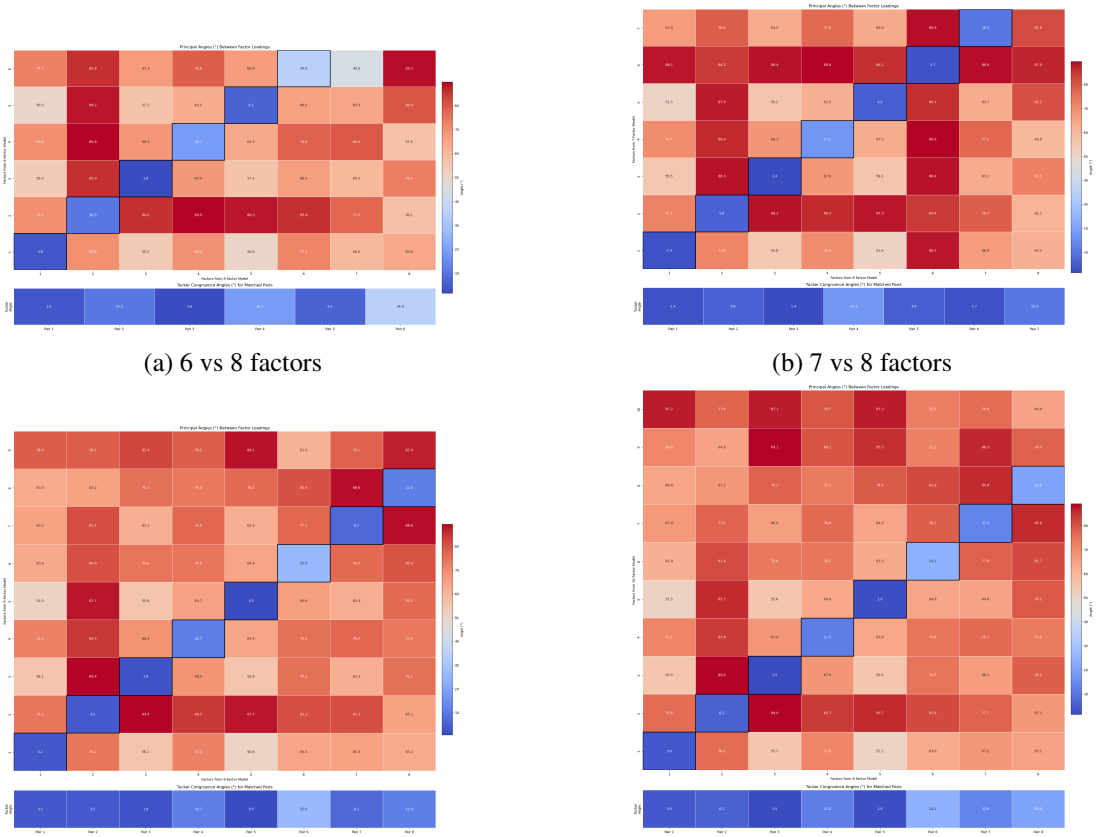
Robustness to task removal. Using a leave-one-task-out protocol, we refit the eight-factor PAF on $B - 1$ tasks, aligned the loadings, and computed principal angles (Fig. 8).

G Comparing to chatbot Arena.

Table 4 summarizes for each skill its correlation with Arena score as well as interpretation for why it is.

H Evaluate New Model Skill Profile

In Fig. 9 we show the evaluated skill profile for T5 and openchat models.



(a) 6 vs 8 factors

(b) 7 vs 8 factors

(c) 8 vs 9 factors

(d) 8 vs 10 factors

Figure 6: Principal angles and Tucker's-congruence angles between the 8-factor solution and other factor dimensions

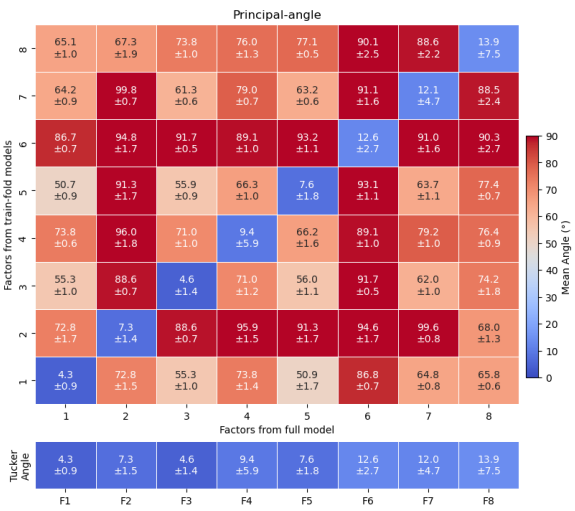


Figure 7: The principal angles and Tucker's-congruence angles between the full solution compared to the factors produced by 70% of the models

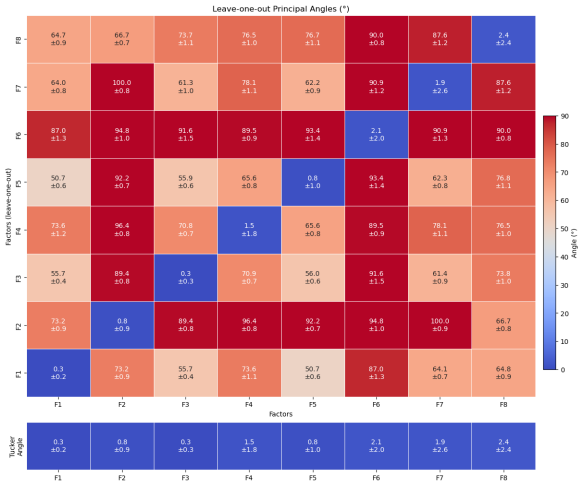
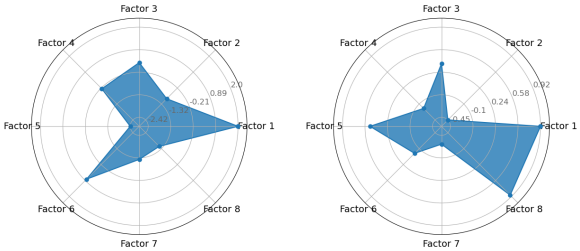


Figure 8: The principal angles and Tucker's-congruence angles between the produced factors when a single task is excluded compared to the factors produced by all tasks



(a) Flan-T5-XL (b) OpenChat-3.5

Figure 9: Projection of new models

Task	Output Type	Eval Metric	Short Description
GLUE – CoLA (Wang et al., 2019)	binary class.	F1	Acceptability judgments for grammatical well-formedness.
GLUE – MNLI (Wang et al., 2019)	mult. choice	F1	3-way natural-language inference across genres.
GLUE – MRPC (Wang et al., 2019)	binary class.	F1	Sentence-level paraphrase detection for news.
GLUE – QNLI (Wang et al., 2019)	binary class.	F1	Sentence-question entailment derived from SQuAD.
GLUE – QQP (Wang et al., 2019)	binary class.	F1	Paraphrase detection for Quora question pairs.
GLUE – RTE (Wang et al., 2019)	binary class.	F1	Textual-entailment classification from PASCAL challenges.
GLUE – SST-2 (Wang et al., 2019)	binary class.	F1	Sentiment analysis of movie snippets.
GLUE – STS-B (Wang et al., 2019)	regression	F1	Continuous sentence-pair semantic similarity.
GLUE – WNLI (Wang et al., 2019)	binary class.	F1	Coreference-switch paraphrase task with high variance.
MMLU (Hendrycks et al., 2021a)	mult. choice	Acc.	57-subject multi-choice exams measuring broad knowledge.
SQuAD (Rajpurkar et al., 2016)	span extract.	LLM	Answer extraction from paragraphs.
m3exam (Zhang et al., 2023)	mult. choice	Acc.	Multilingual, multimodal exam-style benchmark.
MedQA (Jin et al., 2020)	mult. choice	LLM	USMLE-style medical questions.
GSM8K (Cobbe et al., 2021)	generation	LLM	Grade-school math word problems with numeric answers.
Math (Hendrycks et al., 2021b)	generation	LLM	Diverse mathematics problem solving.
LegalBench (MC) (Guha et al., 2023)	mult. choice	Acc.	Suite of specialised legal-reasoning tasks.
LegalBench (Gen) (Guha et al., 2023)	generation	LLM	Generation-style subsets of LegalBench.
BoolQ (Clark et al., 2019)	binary class.	F1	Yes/No QA over Wikipedia passages.
HellaSwag (Zellers et al., 2019)	mult. choice	Acc.	Commonsense sentence completion with adversarial distractors.
OpenBookQA (Mihaylov et al., 2018)	mult. choice	LLM	Elementary-science QA with a supplied fact and commonsense.
TruthfulQA (Gen) (Lin et al., 2022)	generation	LLM	Generation benchmark rewarding factual truth over misconceptions.
TruthfulQA (MC) (Lin et al., 2022)	mult. choice	LLM	Multiple-choice version of TruthfulQA.
MS MARCO (Bajaj et al., 2018)	generation	LLM	Passage retrieval / ranking for web queries.
TriviaQA (Joshi et al., 2017)	span extract.	LLM	Open-domain trivia QA over large corpora.
DailyDialog (Li et al., 2017)	mult. choice	LLM	Dialogue-act / emotion classification in informal chats.
HotpotQA (Yang et al., 2018)	generation	LLM	Multi-hop Wikipedia QA requiring two supporting docs.
CoQA (Reddy et al., 2019)	generation	LLM	Conversational QA grounded in documents.
BIG-Bench (MC) (Srivastava et al., 2023)	mult. choice	F1	Mixed-format challenge stressing broad reasoning.
BIG-Bench (Gen) (Srivastava et al., 2023)	generation	LLM	Generation subsets of BIG-Bench.
QuAC (Choi et al., 2018)	generation	LLM	Multi-turn dialogical QA over Wikipedia.
CNN/DailyMail (See et al., 2017)	generation	LLM	Abstractive news-article summarisation.
XSum (Narayan et al., 2018)	generation	LLM	One-sentence abstractive summaries of BBC news.
IMDB (Maas et al., 2011)	binary class.	F1	Sentiment of full movie reviews.
bAbI (Dodge et al., 2016)	generation	LLM	Synthetic stories with rule-based multi-step queries.
BLiMP (Warstadt et al., 2023)	binary class.	Acc.	67 minimal-pair syntactic phenomena for grammaticality.
BBQ (Parrish et al., 2022)	mult. choice	Acc.	Social-bias sensitivity under controlled knowledge settings.
ARB (Sawada et al., 2023)	generation	F1	Adversarial cross-domain questions to fool LLMs.
Legal-Support (Fourrier et al., 2023)	binary class.	F1	Statute sentences supporting/contradicting a claim.
Synthetic-Reasoning (Fourrier et al., 2023)	generation	LLM	GPT-generated science problems requiring chain-of-thought.
DocNLI (Yin et al., 2021)	binary class.	F1	Document-level natural-language inference.
Contract-NLI (Koreeda and Manning, 2021)	mult. choice	F1	Entail/contradict/neutral classification on contracts.
Quality (Pang et al., 2022)	mult. choice	F1	Long-document QA measuring factuality and completeness.
QMSum (Zhong et al., 2021)	generation	LLM	Query-focused meeting summarisation.
CivilComments (Borkan et al., 2019)	regression	F1	Toxicity classification of online comments.
GPQA (Rein et al., 2023)	generation	LLM	Graduate-level physics questions with diagrams.
SQuALity (Wang et al., 2022)	generation	LLM	Document-level summarisation graded for fidelity and coherence.

Table 3: Tasks used in the factor analysis, showing output type, common evaluation metrics, the metric adopted in this study, and a concise task description.

Skill	**Spearman** ρ	p**-value**	**Case**	**Interpretation**	**Top tasks (task** ρ_{Arena}**)**
General NLU	0.70	0.004	A	Core comprehension and coherence	QQP (.50), MRPC (0.68), WNLI (0.77), IMDB (0.86)
Entailment & Bias	−0.81	<0.001	D	Over-formality and neutrality reduce perceived helpfulness	QNLI (−.52), RTE (−.83), BBQ (−.76)
Long-Doc. Comp.	0.42	0.041	C	Rewarded in long-context reading; neutral elsewhere	CoQA (.32), XSum (.6), SQuAD (.67), HotpotQA (.97), MS MARCO (.81)
Inst. Following / Gen.	0.25	0.130	C	Helpful in structured prompts; less so in free chat	TruthfulQA-Gen (.24), bAbI (.57), M3Exam(.80), BBQ (−.76)
Domain QA	0.65	0.006	A	Accurate factual recall, domain reasoning	BoolQ (.76), MedQA (.81), OpenBookQA (.82)
Social–Ethical Judgment	0.39	0.058	C	Valued for civility; penalized when overly cautious	CivilComments (.55), MMLU (−.12), BigBench (.47)
Prec. & Fidelity	0.12	0.250	B	Factual precision overshadowed by fluency	SQuALITY (.36), Math (.62), Quality (.66), GSM8K (.92), BBQ (−.76)
Grad-Level Reasoning	0.29	0.120	B	Deep reasoning underrecognized by raters	GPQA (.24), TriviaQA (.54), Synt-Reason (.47)

Table 4: Skill-level correlation with Chatbot Arena Elo. Each skill's Spearman ρ reflects how its latent-factor scores align with Arena preference rankings. The final column lists representative high-loading tasks (Arena correlation), clarifying which task families drive each latent dimension and how these are valued in Arena evaluations.