SKILLEVAL: LEARNING INTERPRETABLE ABILITY PROFILES OF LLMS VIA COGNITIVE DIAGNOSIS MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current evaluation of LLMs on benchmarks typically aggregates performance over individual examples into a single scalar score. This approach is overly coarse and ignores the diverse scope of skills covered by benchmark datasets. To address this limitation and better exploit benchmark data, we propose SKILLEVAL, a new evaluation framework that performs cross-model, skill-level analysis. The framework integrates advances from two areas: (1) it leverages the meta-cognitive capabilities of LLMs to automatically discover interpretable skills from questions, and (2) it applies cognitive diagnosis models (CDMs) from psychometrics to estimate per-skill abilities of LLMs. We conduct experiments on 3,811 LLMs evaluated on 9,523 items across five benchmarks spanning diverse domains (MATH, BBH, GPQA, MuSR, IFEval). Compared to the traditional average-based evaluation paradigm, SKILLEVAL produces reliable, more calibrated and informative yet concise model profiles that more accurately predict model behavior on unseen items. These profiles also reveal how benchmarks share skills and how instruction-tuned models differ from their base models. We also demonstrate the utility of SKILLEVAL in two downstream tasks: (1) Computerized Adaptive Testing (CAT), where an unseen model can be efficiently evaluated using a trained CDM, and (2) LLM routing, which dynamically assigns each query to a capable but efficient model, reducing inference cost by over 60% while achieving improved performance. Code is provided as supplementary material to enable reviewer reproduction; the Q-matrix, skill taxonomy, trained NCDM parameters, and per-LLM mastery profiles will be released publicly upon publication.

1 INTRODUCTION

Benchmarks for large language models (LLMs) have rapidly proliferated in recent years. While this creates new opportunities, it also raises a key challenge: how to effectively leverage a large collection of benchmarks for comprehensive LLM evaluation? Current evaluation practices remain quite raw. The common approach is to evaluate a model on all items in a benchmark and then report the average score. This naive statistical way collapses performance across items into a single number, which obscures fine-grained capability differences. For example, two LLMs scoring 0.63 and 0.65 on the MATH dataset may behave very differently on sub-skills such as algebra or logical reasoning. This leads to a natural question: *Given a set of test items, how can we go beyond a scalar score to a more informative, human-understandable capability estimate?*

To answer this, we develop a new evaluation framework that aims to assess the *latent abilities* underlying observed item scores, and represents them as fine-grained skill profiles of LLMs. This problem has long been studied in psychometrics and the education community using various latent variable models. A representative family of approaches is Item Response Theory (IRT; Lord & Novick, 1968), which jointly models test-taker traits (e.g., ability) and item traits (e.g., difficulty and discrimination). IRT models are also increasingly used in machine learning, but typically for data-centric tasks such as dataset analysis or distillation (Kipnis et al., 2025; Maia Polo et al., 2024; Zouhar et al., 2025; Rodriguez et al., 2021), and less often for LLM evaluation as it still produces a scalar overall ability score that suffers from capability collapse. Some recent work explores multidimensional-IRT (Reckase, 2006) and its neural variants that learn ability vectors (Chen et al., 2025; Cui & Sachan, 2025; Song et al., 2025; Gor et al., 2024). However, these multidimensional

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

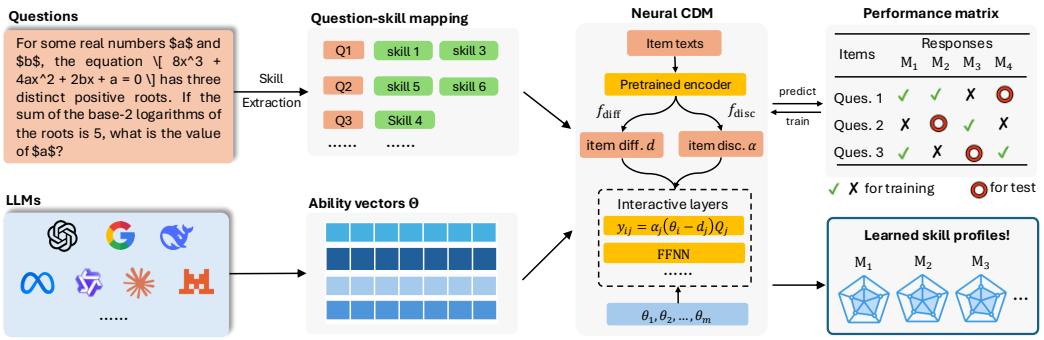

Figure 1: We first extract skills from the questions and construct a question-to-skill mapping (i.e., the Q-matrix). Then, using this Q-matrix together with the performance matrix of models on items, we train a neural CDM to estimate each model's skill-level abilities θ.

models are known to suffer from poor interpretability and non-identifiability, making them less suitable for reliable evaluation.

In this study, we propose SKILLEVAL, a new evaluation framework that identifies latent skills and estimates LLM abilities at the skill level. The framework consists of a skill discovery module, which extracts shared skills from problems, and a neural cognitive diagnosis model (CDM; Wang et al., 2020), which estimates per-skill abilities based on model performance on skill-associated items. We evaluate SKILLEVAL on a large-scale dataset of 36M model–item response records, covering 3,811 models and 9,523 items across five benchmarks from various domains, including MATH (Hendrycks et al., 2021), BBH (Suzgun et al., 2023), GPQA (Rein et al., 2024), MuSR (Sprague et al., 2024), and IFEval (Zhou et al., 2023). We conduct extensive experiments and demonstrate the superiority of SKILLEVAL from various aspects:

- Compared to MIRT-style methods, SKILLEVAL extracts high-quality, human-understandable skills and produces stable ability estimates across runs, ensuring reproducible and robust evaluation.
- SKILLEVAL accurately predicts model performance at both the item (with AUC $= 81.73$) and benchmark (with Pearson $r = 0.989$) levels, outperforming strong baselines. In a downstream routing task where the goal is to assign queries to a cheap yet capable model, SKILLEVAL matches the strongest LLM at 25% of the cost (τ=0.80, 99.9% of strongest accuracy), and exceeds it by 3.7 points at 37% of the cost (τ=0.95). It also profiles previously unseen LLMs without retraining the shared network: 20 adaptively chosen answers yield a mean benchmark-score error comparable to 100 randomly chosen answers.
- In addition, SKILLEVAL serves as a useful tool for analyzing dataset structure and model behavior, such as skill coverage across benchmarks, co-mastery patterns among skills, and skill-level differences between base and instruction-tuned models, offering useful insights into data curation and model training.

2 PROFILING LLMS' SKILLS WITH COGNITIVE DIAGNOSTIC MODELS

2.1 PROBLEM STATEMENT

Given a set of models to be evaluated $\{\mathcal{M}_1, ..., \mathcal{M}_M\}$ and a collection of test items $\{x_1, ..., x_N\}$ from one or multiple benchmarks, we represent their performance using a matrix $\mathcal{S} \in \mathbb{R}^{M \times N}$, where each entry $\mathcal{S}_{i,j}$ denotes the score of the i-th model on the j-th item. We formulate LLM profiling as a function $\mathcal{F}$ that takes $\mathcal{S}$ as input and outputs some estimates for each model's capabilities, i.e., $\mathcal{F}(\mathcal{S}) \to \{\theta_1, ..., \theta_M\}$, where θ_i is a quantitative measurement of the capabilities of model i.

The common practice in evaluation is to model $\mathcal{F}$ as a simple average over individual items, i.e., $\theta_i = \frac{1}{n}\sum_{j=1}^{n} S_{ij}$. While this provides a concise summary of model performance, it suffers from two key limitations. First, it compresses the model's performance into a single scalar metric, failing

to capture variations across fine-grained skills. Second, the naive average can be affected by bias induced by the difficulty distribution of the test items, and may not faithfully reflect the model's true ability. Indeed, this type of ability modeling assumes that all items in a given benchmark contribute equally to the estimation of a single ability dimension, which is a strong and implausible assumption.

To address these limitations, we propose a new paradigm for profiling LLMs. We assume that the items in the benchmark are associated with a set of underlying latent skills $\mathcal{C} = \{c_1, \ldots, c_K\}$. Inspired by psychometric approaches for assessing latent traits, we use Cognitive Diagnosis Models (CDMs) to construct the profiling function $\mathcal{F}_{\mathrm{CD}}(\mathcal{S}) \to \Theta \in \mathbb{R}^{M \times K}$, where the profile θ_i for each model is a k-dimensional vector representing its proficiency over the set of skills.

2.2 Q-Matrix Construction

We use a binary Q-matrix $\mathcal{Q} \in \mathbb{R}^{N \times K}$ to denote the mapping between N items and K skills, where each entry $\mathcal{Q}_{i,j}$ indicates whether item i is associated with skill j, taking value 1 if so and 0 otherwise. Manually defining and annotating skills is challenging, costly, and difficult to scale. Recent work shows that LLMs can automatically identify meaningful skills from problem content (Didolkar et al., 2024). We adapt this approach to our setting, consisting of two main steps: skill labeling and skill clustering.

Skill labeling. For each item j, we prompt GPT-4o-mini with the question stem and ask for (i) a single *primary* skill and (ii) the full list of skills required for the solution (exact prompt in Appendix D). We use the full list rather than only the primary skill: primary-skill assignments produce Q-matrix rows of mean row-sum 1.0, which under-constrains NCDM. Full-list extraction yields a mean of 2.9 skills per item and $13{,}276$ unique skill strings across $9{,}523$ items.

Skill clustering. Because skills are extracted independently from different problems, they may contain redundancy and noise. To obtain a more compact and coherent skill set, we cluster similar skills and merge them accordingly. Didolkar et al. (2024) employ an LLM to perform this grouping process; however, such approaches may not scale well to a large number of skills. Therefore, we adopt an embedding-based clustering method, which is more efficient and scalable. Each unique skill string is first encoded by the pretrained SBERT `all-mpnet-base-v2` (Reimers & Gurevych, 2019) into $\mathbb{R}^{768}$. Then, we use the hierarchical agglomerative clustering (HAC) algorithm with cosine distance and average linkage, specifying the number of final clusters as K.

Q-matrix assembly. We set $q_{jk} = 1$ iff at least one of item j's extracted skills falls into cluster k. This multi-skill assignment reduces the fraction of clusters with fewer than ten items from 43% under primary-skill assignment to 36% at $K{=}100$[1], yielding a mean column-sum of ~ 139 items per skill (mean row-sum 1.5 clusters per item, reflecting the compression of the 2.9 extracted skill strings into shared clusters), the regime in which NCDM trains stably.

2.3 Neural CDM

We adopt the Neural Cognitive Diagnosis Model (Wang et al., 2020). The probability that LLM s answers item j correctly is estimated as:

$$p(\mathcal{S}_{ij} = 1) \;=\; \mathrm{sigmoid}\Big(f\big(\alpha_j \, (\boldsymbol{\theta}_i - \boldsymbol{d}_j) \odot \mathbf{q}_j \big) \Big), \tag{1}$$

where $\boldsymbol{\theta}_i \in [0,1]^K$ represents the learned skill mastery of model $\mathcal{M}_i$, $\alpha_j \in [0,1]$ is the discrimination parameter of item j, $\boldsymbol{d}_j \in [0,1]^K$ is its per-skill difficulty vector, and $\mathbf{q}_j \in \{0,1\}^K$ is the corresponding Q-matrix row. The scalar α_j scales all skill dimensions equally. The function f is a three-layer MLP with $\mathrm{PosLinear}$ layers. Each $\mathrm{PosLinear}$ layer uses non-negative weights, $W^+ = \max(W, 0)$. Together with the non-decreasing sigmoid activations, this ensures that increasing any mastery component cannot decrease the predicted probability, holding all other inputs fixed.

[1]We adopt $K{=}100$ from $\{50, 100, 150, 200, 300, 400, 500\}$ which achieves a better balance between interpretability and performance. See Appendix F for more details.

2.4 Learning Item Parameters from Text

The original NCDM treats item discrimination α_j and difficulty $\boldsymbol{d}_j$ as model parameters learned during training, preventing predictions on unseen items. To overcome this limitation, we follow previous context-aware approaches (Gor et al., 2024; Cui & Sachan, 2025) to use a lightweight neural network to learn item parameters based on their text features. Specifically, for a given item x_j, we first use a pretrained encoder to obtain its embedding $\boldsymbol{e}_j$. Then, we compute α_j and $\boldsymbol{d}_j$ via:

$$\alpha_j = \sigma\big(\mathbf{w}_\alpha^\top \boldsymbol{e}_j + b_\alpha\big), \qquad \boldsymbol{d}_j = \sigma\big(\mathbf{W}_d\, \boldsymbol{e}_j + \mathbf{b}_d\big). \tag{2}$$

To maintain computational efficiency, we freeze the pretrained encoder and only optimize the projection parameters $\mathbf{w}_\alpha, \mathbf{W}_d$ and $b_\alpha, \mathbf{b}_d$.

3 Evaluation of Skill Profiles

Experiment Setup. We train our text-conditioned NCDM on a subset of the RouterEval dataset (Huang et al., 2025), including 3,811 LLMs evaluated on 9,523 items from five benchmarks: MATH, BBH, GPQA, MuSR, and IFEval. For the number of skills, we set K=100 as the main experimental configuration, as it achieves a better balance between interpretability and performance. We provide comparisons with other values of K in Appendix F.

3.1 Quality of Discovered Skills

Since our evaluation is centered around skills, it is crucial to assess the quality of these LLM-discovered skills. We consider the following three criteria: (1) **Semantic clarity**, which measures whether a skill description is clear, unambiguous, and self-contained; (2) **Level of abstraction**, which captures the granularity of a skill, where a good skill should be neither overly broad nor overly specific to particular questions; (3) **Diversity**, which evaluates whether different skills are non-redundant and capture distinct aspects. For the first two criteria, we adopt LLM-as-a-judge. For the last one, we use intra-list diversity defined as averaged pair-wise dissimilarities, which is widely used in recommendation systems.

The experimental results are shown in Table 1. As a reference, we compare our skills with the Common Core State Standards[2], a widely used expert-curated framework for K–12 math education. Our skills achieve a similar level of semantic clarity to the Common Core. In terms of granularity, they are slightly more fine-grained. They also show lower variance (std. 0.29 vs. 0.55), indicating more consistent granularity across skills, which is desirable. In addition, our skills are more diverse (avg. 0.83 vs. 0.76), likely because they cover more domains and reduce redundancy through clustering. Overall, our automatically extracted skills are competitive with manually designed ones along key dimensions.

Table 1: Comparison between the LLM-extracted skill set (K = 100) and an expert-defined mathematics skill set (the Common Core standards; K = 406).

	LLM-extracted	Expert-defined
Clarity	4.07	4.14
Granularity	2.26	2.72
Diversity	0.83	0.76

Human validation of skill–question assignments. Clear skill descriptions do not by themselves establish that the skills are relevant to their assigned questions. We therefore evaluate the skill–question assignments through human annotation. Three annotators, each holding an MSc in engineering, evaluated 780 skill–question pairs covering all 100 skills. We sampled up to 10 questions uniformly at random from each skill's assigned items, retaining all items for smaller clusters. Annotators were shown revised descriptive skill names and question text, and judged whether answering each question required the named skill. Majority voting accepted 652 of 780 assignments (83.6%), with an acceptance rate of 81.7% when averaged equally across skills. Inter-annotator agreement was 0.715 in mean pairwise Cohen's kappa and 0.714 in Krippendorff's nominal alpha; both measures equal 1 for perfect agreement and 0 for agreement expected by chance. These judgments provide evidence for the semantic relevance of the sampled assignments.

[2] https://corestandards.org/

3.2 CONSTRUCT VALIDITY OF Θ

To assess construct validity (American Educational Research Association et al., 2014), we investigate how well Θ predicts model behavior in independent tasks. On held-out items, the mean per-LLM correlation between Θ and per-skill accuracy is 0.73 across five seeds,[3] validating Θ as a measurement target. We further test the full predictive power of Θ via item- and benchmark-level prediction with the trained NCDM.

Item-level Prediction. We consider both in-distribution (ID) and out-of-distribution (OOD) settings. In the ID setting, we perform an (LLM $\times$ Item) interaction-wise split, where all items are observed during training. In the OOD setting, we use an item-wise split, holding out a subset of items that are never seen during training. For comparison, we consider IRT-based models: the classical two-parameter logistic (2PL) IRT model, which gives each LLM a single ability value and each item a difficulty and a discrimination, and IRTNet (Chen et al., 2025), a multidimensional neural IRT model that represents each LLM with a d-dimensional ability vector. Table 2 reports two measures on the test responses. The AUC is the probability that a randomly chosen correct response receives a higher predicted probability than a randomly chosen incorrect one; in Table 2, 50 corresponds to chance. Acc is the share of responses predicted correctly when the predicted probability is thresholded at 0.5. The 2PL model scores highest in the ID setting, mainly because it learns a separate difficulty and discrimination for every item, which also means that it cannot score items unseen during training. We can see that our model performs on par with IRTNet in the ID setting, although IRTNet has many times more parameters than ours. However, in the OOD setting, our model exhibits a larger performance drop, and IRTNet performs better. This pattern is consistent with SKILLEVAL's skill-based inductive bias: each question interacts only with its assigned ability dimensions. This restriction can limit predictive flexibility on held-out questions.

Table 2: Performance of various models under two settings: ID (interaction-wise split) and OOD (item-wise split). IRT 2PL has no item features and therefore cannot generalise to held-out items; its OOD column is left blank. OOD values are means over seeds, five for SKILLEVAL and three for IRTNet. ID values are single runs.

Model	AUC		Acc	
	ID	OOD	ID	OOD
IRT (2PL)	**86.16**	–	**78.76**	–
IRTNET (d=232)	81.38	75.04	74.21	69.05
IRTNET (d=512)	81.36	75.14	74.21	69.07
IRTNET (d=1024)	81.74	**75.35**	74.68	**69.38**
SKILLEVAL (K = 100)	81.73	72.15	75.57	67.62

Benchmark-level prediction. Benchmark accuracy is often simply used to compare LLMs. We therefore ask whether our item-level predictions also recover each LLM's score on each benchmark. In the OOD setting, we average an LLM's predicted probabilities over a benchmark's held-out items and compare the result with its observed accuracy on those same items (Figure 2). SKILLEVAL achieves an overall Pearson correlation of r=0.989. Its per-benchmark correlations are MATH r=0.977, BBH r=0.984, GPQA r=0.809, IFEval r=0.966, and MuSR r=0.629. IRTNet predicts individual item responses more accurately (Table 2) and achieves a similar overall correlation (r=0.993). However, its correlation is lower on each individual benchmark, particularly MATH (r=0.852) and IFEval (r=0.718). Thus, the two models perform similarly when benchmark scores are considered together, while SKILLEVAL better tracks differences among LLMs within each benchmark.

3.3 STABILITY OF Θ

We test whether ability profiles remain consistent across retraining. Latent ability models can use different coordinates across runs; SKILLEVAL ties each dimension to a named skill through a fixed Q-matrix. We train SKILLEVAL with five random seeds and each reported IRTNet configuration with three. Within each configuration, we compute Pearson correlations across LLMs for corresponding dimensions in every pair of runs. Averaged over dimensions and run pairs, the correlation is 0.96 for SKILLEVAL and rounds to 0.00 for each IRTNet configuration. Procrustes alignment, which transforms one run's coordinates to best match another's, raises IRTNet's mean correlations to 0.10–0.43 (Appendix L). Thus, SKILLEVAL supports more consistent comparisons of relative ability across LLMs on named skills.

[3]Each skill is weighted by its number of held-out items. Appendix K reports controls for overall, benchmark-level, and subtask-level accuracy.

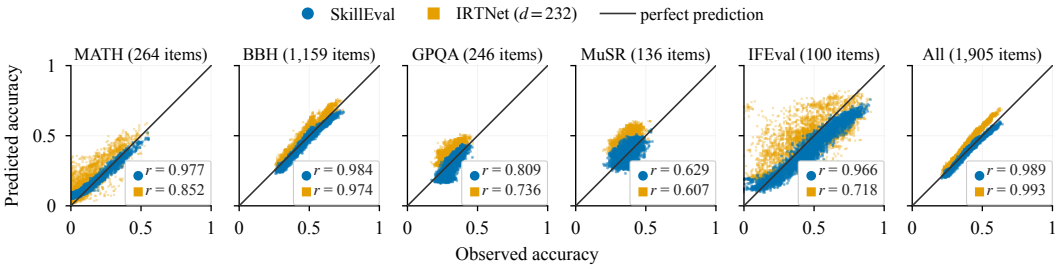

Figure 2: Predicted versus observed benchmark accuracy on held-out items in the OOD setting. Each point represents one LLM. The horizontal axis shows its observed accuracy on the benchmark's held-out items; the vertical axis shows the mean predicted probability of a correct answer on those items. The diagonal indicates exact agreement, and r is the Pearson correlation across 3,811 LLMs. IRTNet uses d=232.

4 Applications

4.1 Routing LLMs via Ability Profiles

LLM routing selects a model for each incoming query, with the aim of balancing answer accuracy and inference cost (Hu et al., 2024; Huang et al., 2025). We first test whether skill profiles improve accuracy by selecting an LLM for each query, rather than always using the strongest model overall. For a test item j SkillEval estimates the probability $P_i(j) = p(\mathcal{S}_{ij}{=}1)$ that LLM i answers it correctly (Equation 1). The router selects the LLM with the highest predicted probability:

$$\hat{\imath}(j) \;=\; \arg\max_{i\in\{1,\ldots,M\}} P_i(j). \qquad (3)$$

We evaluate on the N_{test}=1,905 held out items of the OOD split with M=3,811 candidate LLMs. For routers that rank multiple candidates, Acc@k is the fraction of test items for which at least one of the top-k LLMs answer correctly. Thus, Acc@1 measures the accuracy of the selected LLM, while larger values of k measure whether a shortlist contains a correct answer.

Table 3: Routing performance on 1,905 held-out items with 3,811 candidate LLMs. All scores are percentages. Rows marked "3 seeds" report the mean over three trainings with different seeds, with the standard deviation in parentheses; IRTNet uses d=1024, its best size in Table 2. The strongest single LLM and KNN involve no training or randomness, Random is a single random draw, and the other rows are single training runs. Bold marks the highest value in each column.

Method	Acc@1	Acc@3	Acc@5	Acc@10
Random	40.47	–	–	–
Strongest single LLM	61.57	–	–	–
IRT 2PL	61.10	77.74	81.94	85.93
KNN	62.05	**78.01**	**83.10**	**88.98**
IRTNet, 3 seeds	63.08	77.31	81.21	86.44
	(0.98)	(0.55)	(1.60)	(0.76)
SkillEval (K=100)	**65.67**	77.95	82.94	87.51
SkillEval, 3 seeds	65.46	77.55	82.43	87.49
	(0.34)	(1.26)	(0.37)	(0.47)

We compare against five baselines. *Random* routing selects an LLM uniformly at random for each item. The *strongest single LLM* always uses the model with the highest accuracy on the training items, `calme-3.2-instruct-78b`.[4] The *IRT 2PL* router ranks LLMs by the single ability value that a fitted 2PL model assigns to each of them (Lord & Novick, 1968), giving the same ranking for every query. Related work, IRT-Router (Song et al., 2025), instead uses multidimensional IRT for routing. The *KNN* router (Chen et al., 2025) retrieves the 10 training items with the most similar text embeddings to item j. For each candidate LLM, it estimates the probability of a correct answer as the fraction of these retrieved items that the LLM answered correctly. *IRTNet* routes like SkillEval, selecting the LLM with the highest probability predicted by its own model.

Among the methods in Table 3, SkillEval achieves the highest Acc@1: 65.67%, compared with 61.57% for the strongest single LLM, 61.10% for the IRT 2PL router, 62.05% for KNN, and 63.08% for IRTNet over three runs. Across three retrainings, SkillEval's Acc@1 is $(65.46 \pm 0.34)\%$ (mean $\pm$ standard deviation). IRTNet predicts individual responses more accurately (Table 2) but picks a correct LLM less often, since routing depends only on the top-ranked LLM for each question. For

[4] `https://huggingface.co/MaziyarPanahi/calme-3.2-instruct-78b`

larger shortlists, KNN is competitive: its Acc@3 is similar to that of SKILLEVAL, and it achieves higher Acc@5 and Acc@10. The per-benchmark breakdown (Appendix H, Table 9) shows how these results vary across domains. The largest gains over the IRT 2PL router occur on MuSR (+12.5 percentage points) and IFEval (+14.0 percentage points).

Cost–accuracy trade-off. The preceding experiment selects models for accuracy alone. We now consider inference cost, estimated as the computation per query in billions of floating-point operations (GFLOPs). For each item, the router selects the cheapest LLM whose predicted probability of a correct answer exceeds a threshold τ. If none qualifies, it selects the LLM with the highest predicted probability. Figure 3 compares SKILLEVAL, IRTNet (d=232), and KNN using this same rule, varying τ from 0.10 to 0.95 and expressing cost relative to the strongest single LLM. At τ=0.80, SKILLEVAL reaches 99.9% of that LLM's accuracy at 25% of its cost; at τ=0.95, it exceeds that LLM's accuracy by 3.7 percentage points at 37% of its cost. KNN is more accurate at low cost but levels off at 98% of the strongest LLM's accuracy. Its probability estimates take only the eleven values $0/10, \ldots, 10/10$, giving many tied candidates at the highest score. At matched cost budgets in the plotted range, SKILLEVAL achieves higher accuracy than IRTNet.

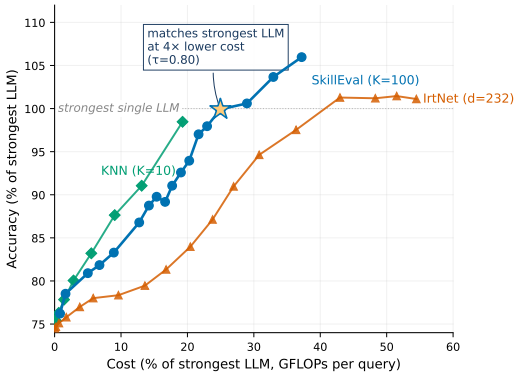

Figure 3: Accuracy versus estimated inference cost for three routers using the same threshold rule. Both axes are expressed relative to the strongest single LLM selected on training items. Each point corresponds to one threshold τ; the star marks SKILLEVAL at τ=0.80.

Skill-level "weak-beats-strong" inversions. A fixed ranking of LLMs cannot adapt to questions on which a model with lower average accuracy performs better. To assess how often such opportunities occur, we examine the observed test responses. On 723 of the 1,905 test items (38.0%), an LLM with at most 13B parameters answers correctly while the strongest single LLM selected on the training items fails. We call these cases "weak-beats-strong" inversions. They occur in every benchmark (BBH: 329 items, MATH: 156, GPQA: 155, MuSR: 64, IFEval: 19) and span 41 skill clusters (Figure 6 in Appendix I). These cases illustrate why selecting an LLM for each query can be useful even when one model leads on average.

4.2 PROFILING UNSEEN LLMS

When a new LLM becomes available, we want to estimate its skill profile from a small set of its answers without retraining the entire diagnostic network. We train the text-conditioned NCDM using only training-item responses from 3,048 LLMs. We then freeze the network, including the item difficulty and discrimination parameters, and estimate only the K-dimensional profile of each new LLM from its answers to N items drawn from the training-item pool. This is ability estimation with fixed item parameters, also called *scoring* in psychometrics (Segall, 1996). We evaluate this procedure on 200 held-out LLMs, using separate test items whose responses never enter profile estimation.

Starting point and related skills. We fit a multivariate Gaussian to the raw (pre-sigmoid) profiles of the 3,048 training LLMs and use it as a prior. Starting from the prior mean, we update the new LLM's profile with L-BFGS, using Equation 1 with a Gaussian prior regularizer. Because the Gaussian captures correlations between skills, an answer also shifts skills that are correlated with the item's assigned skills. As more answers are observed, the data term increasingly outweighs the fixed prior (Appendix G).

Choosing the questions. We compare uniform random selection with adaptive selection. Computerized adaptive testing (CAT) chooses each next question based on the answers given so far; CD-CAT

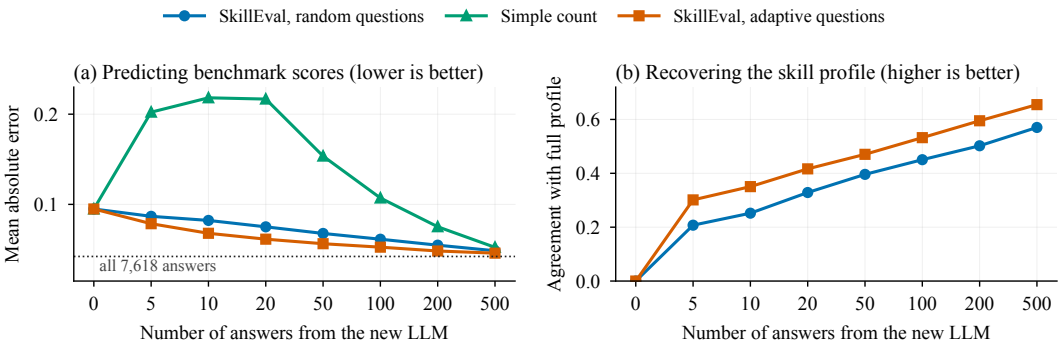

Figure 4: Profiling 200 held-out LLMs from N answers. (a) Mean absolute benchmark-score error across LLMs and the five benchmarks; the dotted line uses all 7,618 pool answers with the same estimator. (b) Pearson correlation with profiles estimated from all answers without the prior, computed across LLMs for each skill and averaged over skills.

applies this idea to cognitive diagnosis models (Cheng, 2009). Starting from the prior mean, we use the current profile to predict the probability p_j of a correct response to each unused item j and select:

$$j^* = \arg\max_j \; p_j \, (1 - p_j) \, \|\mathbf{q}_j\|_1. \tag{4}$$

The uncertainty term $p_j(1-p_j)$ is largest at p_j=0.5. The skill count $\|\mathbf{q}_j\|_1$ favours items assigned to several skills. After observing the answer, we update the profile using all answers so far and repeat until N items have been answered. Both selection methods use the same final profile-estimation procedure.

Measures. We measure what the answers reveal about each LLM in two ways. For each benchmark, we average the predicted probabilities over its test items and compare this predicted accuracy with the observed accuracy on those items. The *benchmark-score error* is their absolute difference, averaged over the five benchmarks and the LLMs. *Profile agreement* compares the relative skill estimates from N answers with a reference profile estimated from all 7,618 pool answers without the prior. We compute Pearson correlation across LLMs for each skill, then average over skills. A *simple-count* baseline uses the same randomly selected items to estimate each benchmark score as the fraction answered correctly. If no items from a benchmark have been answered, it uses the prediction from the zero-answer profile.

Results. With no answers, the starting profile predicts benchmark scores with a mean error of 0.095 (Figure 4). With 100 random answers, SKILLEVAL lowers the error to 0.061, and with 500 answers to 0.049, approaching the 0.042 obtained from all 7,618 answers. Adaptive selection reaches similar errors with fewer answers: its mean error after 10 answers is 0.068, comparable to 50 random answers, and after 20 answers it is 0.061, comparable to 100 random answers. It also yields higher profile agreement: 0.53 after 100 answers, versus 0.45 for random selection. Adaptive selection yields lower benchmark-score error and higher profile agreement than random selection at every tested budget from 5 to 500 answers (Appendix G). At the nonzero tested budgets up to 100 answers, the simple count has higher error than the zero-answer prediction; with 500 answers, its error is 0.052.

5 DIAGNOSTIC ANALYSES

The Q-matrix and the learned mastery profiles support complementary analyses of benchmark coverage and model behavior. We examine skill coverage across benchmarks and differences between base and instruction-tuned models.

Skill coverage across benchmarks. To identify distinct and overlapping skill coverage across benchmarks, we count a skill as covered when at least one item in a benchmark is assigned to it in

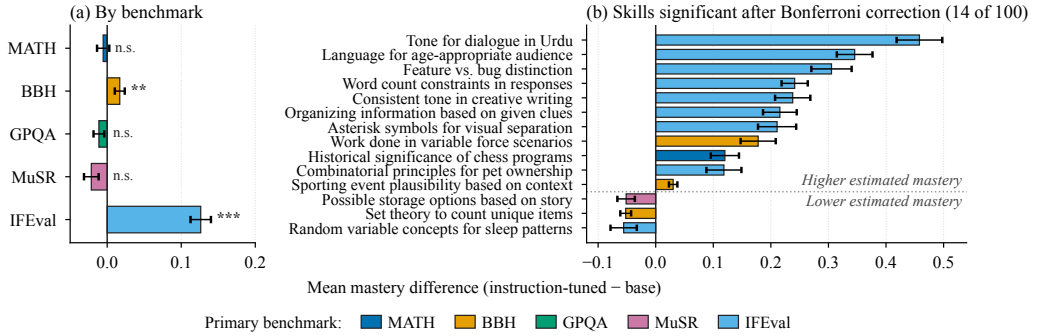

Figure 5: Estimated mastery differences between base and instruction-tuned models. (**a**) Benchmark means, assigning each skill to its primary benchmark by item count; stars indicate unadjusted p-values ($*\,p{<}0.05$, $**\;p{<}0.01$, $***\,p{<}0.001$). (**b**) The $14/100$ skills significant after Bonferroni correction ($\alpha{=}5{\times}10^{-4}$), sorted by mean difference and coloured by primary benchmark; names are abbreviated. Error bars: ±1 SEM across 81 matched model pairs. Omitted skills are not necessarily unchanged.

the Q-matrix. Under this criterion, 30 of the 100 skills occur in only one benchmark, including 13 unique to IFEval, 9 to GPQA and 6 to MATH (Figure 7 in Appendix N). MATH and IFEval together cover 87 skills; adding GPQA increases this to 98. BBH adds one further skill to this combined coverage despite contributing 5,761 items. Shared coverage suggests potential redundancy, but benchmarks can still differ in difficulty, item diversity and how well they distinguish between models. The concentration of unique skills in IFEval is consistent with the transfer result in Appendix J: withholding IFEval from training removes all training items for these skills, and routing performance on IFEval deteriorates.

Skill-level differences between base and instruction-tuned models. We compare 81 matched pairs, each comprising a base model and its instruction-tuned version. Averaged across pairs, estimated mastery is more than 0.01 higher in instruction-tuned models on 38 skills and more than 0.01 lower on 36 skills. Separately, paired Wilcoxon signed-rank tests identify 14 skills with significant differences after Bonferroni correction, which divides the significance level by the 100 tested skills (Figure 5). Eleven of these differences are positive, mostly on instruction-following skills, and three are negative, with magnitudes of at most 0.06. When skills are grouped by the benchmark contributing most of their items, IFEval has the largest mean difference ($\Delta{=}{+}0.126$, $p{<}10^{-10}$); Appendix O reports the differences by model family. Statistical significance does not establish a large effect, and these observational comparisons do not isolate the causal effect of instruction tuning.

6 Conclusion

SkillEval applies a Neural Cognitive Diagnostic Model to 3,811 LLMs across 9,523 items and five benchmarks, producing interpretable mastery profiles over $K{=}100$ named skills. Unlike IRTNet's latent dimensions, these profiles are stable across retraining: each skill's estimates correlate at 0.96 on average between runs. Skill-conditioned routing reaches Acc@1=65.67% (vs. 61.57% for the strongest single LLM, 62.05% for KNN, 63.08% for IRTNet over three runs and 61.10% for the IRT 2PL router) and matches the strongest LLM at $4\times$ lower inference cost. SkillEval also profiles previously unseen LLMs without retraining the shared network: 20 adaptively chosen answers give a mean benchmark-score error comparable to 100 randomly chosen answers. Skill profiles recover benchmark-level scores at Pearson $r{=}0.989$. The framework surfaces diagnostic findings invisible to aggregate scores: differences between base and instruction-tuned models that are statistically significant after Bonferroni correction on $14/100$ skills, and overlapping skill coverage across benchmarks. An exploratory analysis in Appendix M examines asymmetric co-mastery: LLMs mastering one skill often also master another, while the reverse is less common. Testing how these patterns generalise across LLM populations is a direction for future work. Limitations are in Appendix P.

AI USE STATEMENT

In this work, we used generative AI tools to help write and run experiment and analysis code, and to polish the wording of the paper. We have not used generative AI tools to generate synthetic data or to prove mathematical claims. LLMs are also part of the method itself: skills are extracted from benchmark questions with an LLM (Section 2). We have reviewed all AI-assisted work: code was tested and its outputs were checked against logged experiment results by the authors, and every reference was verified against its publisher record. We take responsibility for the final content of this work.

REFERENCES

American Educational Research Association, American Psychological Association, and National Council on Measurement in Education. *Standards for Educational and Psychological Testing*. American Educational Research Association, Washington, DC, 2014. ISBN 978-0-935302-35-6.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models, 2024. URL https://arxiv.org/abs/2405.14782.

bigdata ustc. Educdm. https://github.com/bigdata-ustc/EduCDM, 2021.

Jianhao Chen, Chenxu Wang, Gengrui Zhang, Peng Ye, Lei Bai, Wei Hu, Yuzhong Qu, and Shuyue Hu. Learning compact representations of llm abilities via item response theory, 2025. URL https://arxiv.org/abs/2510.00844.

Ying Cheng. When cognitive diagnosis meets computerized adaptive testing: Cd-cat. *Psychometrika*, 74(4):619–632, Dec 2009. ISSN 1860-0980. doi: 10.1007/s11336-009-9123-2. URL https://doi.org/10.1007/s11336-009-9123-2.

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. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Peng Cui and Mrinmaya Sachan. Investigating the zone of proximal development of language models for in-context learning. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6485–6498, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.362. URL https://aclanthology.org/2025.findings-naacl.362/.

Jimmy de la Torre. Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130, 2009. doi: 10.3102/1076998607309474. URL https://doi.org/10.3102/1076998607309474.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. Metacognitive capabilities of llms: An exploration in mathematical problem solving, 2024. URL https://arxiv.org/abs/2405.12205.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Maharshi Gor, Hal Daumé III, Tianyi Zhou, and Jordan Boyd-Graber. Do great minds think alike? investigating human-AI complementarity in question answering with CAIMIRA. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21533–21564, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1201. URL https://aclanthology.org/2024.emnlp-main.1201/.

J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. doi: 10.1007/BF02291478. URL https://doi.org/10.1007/BF02291478.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system, 2024. URL https://arxiv.org/abs/2403.12031.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. RouterEval: A comprehensive benchmark for routing LLMs to explore model-level scaling up in LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 3860–3887, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.208. URL https://aclanthology.org/2025.findings-emnlp.208/.

Alex Kipnis, Konstantinos Voudouris, Luca M. Schulze Buschoff, and Eric Schulz. metabench - a sparse benchmark of reasoning and knowledge in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4T33izzFpK.

Frederic M. Lord and Melvin R. Novick. *Statistical Theories of Mental Test Scores*. Addison-Wesley, 1968.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Mark D. Reckase. 18 multidimensional item response theory. In C.R. Rao and S. Sinharay (eds.), *Psychometrics*, volume 26 of *Handbook of Statistics*, pp. 607–642. Elsevier, 2006. doi: https://doi.org/10.1016/S0169-7161(06)26018-8. URL https://www.sciencedirect.com/science/article/pii/S0169716106260188.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4486–4503, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.346. URL https://aclanthology.org/2021.acl-long.346/.

Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451. URL https://doi.org/10.1007/BF02289451.

Daniel O. Segall. Multidimensional adaptive testing. *Psychometrika*, 61(2):331–354, Jun 1996. ISSN 1860-0980. doi: 10.1007/BF02294343. URL https://doi.org/10.1007/BF02294343.

Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15629–15644, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.761. URL https://aclanthology.org/2025.acl-long.761/.

Zayne Rea Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jenyYQzue1.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6153–6161. Association for the Advancement of Artificial Intelligence (AAAI), April 2020. doi: 10.1609/aaai.v34i04.6080. URL https://doi.org/10.1609/aaai.v34i04.6080.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yu Yin, Shijin Wang, and Yu Su. Neuralcd: A general framework for cognitive diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8312–8327, August 2023. doi: 10.1109/TKDE.2022.3201037.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Vilém Zouhar, Peng Cui, and Mrinmaya Sachan. How to select datapoints for efficient human evaluation of NLG models? *Transactions of the Association for Computational Linguistics*, 13:1789–1811, 2025. doi: 10.1162/tacl.a.60. URL https://aclanthology.org/2025.tacl-1.80/.

A Related Work

LLM Evaluation and Routing. Standard LLM evaluation aggregates item-level correctness into a single benchmark score (Gao et al., 2024; Hendrycks et al., 2021; Cobbe et al., 2021), creating leaderboards whose rankings are influenced as much by evaluation protocol (prompt format, shot count, decoding parameters) as by true model capability (Biderman et al., 2024). Platforms such as the Open LLM Leaderboard and RouterBench Hu et al. (2024) release per-item correctness matrices for thousands of LLMs, enabling but not themselves performing fine-grained analysis. Routing systems match queries to the cheapest capable model, typically using latent query-model embeddings without explicit semantic skills. Closest in spirit to our work, CAIMIRA Gor et al. (2024) applies IRT to factual QA, and Didolkar et al. Didolkar et al. (2024) use LLM-generated skill labels to probe mathematical reasoning. We build on the latter by scaling skill extraction to five benchmarks and fitting a full CDM over the extracted skills.

Cognitive Diagnostic Models. Cognitive Diagnostic Models (CDMs) originate in educational psychometrics and estimate latent skill mastery from observed item responses, extending Item Response Theory (IRT) by modelling proficiency as a vector over *named* skills rather than a scalar or a continuous embedding. Classical CDMs such as DINA de la Torre (2009) assume conjunctive skill requirements (an item is correct only if all required skills are mastered) and binary mastery. The Neural CDM (NCDM) Wang et al. (2020) relaxes these assumptions with neural interaction functions while preserving mastery-to-score monotonicity via non-negative weight constraints, and is generalised in NeuralCD Wang et al. (2023); reference implementations are collected in the EduCDM library bigdata ustc (2021). CDMs have been widely deployed for human assessment over hundreds of skills and tens of thousands of examinees, making them a natural fit for LLM evaluation where response matrices are now orders of magnitude larger.

Psychometric Methods for LLMs. Recent work applies psychometric models to LLMs at scale. IRT-Router Song et al. (2025) uses multidimensional IRT with a query-cluster relevance vector for routing. IRTNet Chen et al. (2025) jointly learns d-dimensional LLM ability embeddings, query discrimination, and query difficulty via a Mixture-of-Experts architecture, evaluated on 112 LLMs and 10 benchmarks. Metabench (Kipnis et al., 2025) compresses six public benchmarks to under 3% of their combined items using 2-PL IRT and Fisher-information filtering on $\sim 5{,}000$ LLMs, showing that a single latent factor explains 79% of benchmark variance. TinyBenchmarks (Maia Polo et al., 2024) similarly uses IRT-based item selection to reduce evaluation cost. All of these approaches treat LLM ability as a low-dimensional latent embedding without a named skill structure. SKILLEVAL commits to an *a priori* Q-matrix over $K{=}100$ semantically labeled skills derived from LLM metacognition and SBERT clustering, enabling skill-level analyses of co-mastery patterns, skill interaction within items, and base versus instruction-tuned differences that are not expressible over a continuous latent embedding. We scale to $3{,}811$ LLMs ($34\times$ IRTNet, comparable to metabench's $\sim 5{,}000$) across five benchmarks.

B NOTATION

Table 4 summarises the notation used throughout the paper.

Table 4: Notation summary.

Symbol	Meaning
M	Number of LLMs ($M{=}3{,}811$)
N	Number of items ($N{=}9{,}523$); also the number of answers used to profile a new LLM
K	Number of skill clusters ($K{=}100$ default)
$i \in \{1, \ldots, M\}$	LLM index
$j \in \{1, \ldots, N\}$	Item index
$k \in \{1, \ldots, K\}$	Skill index
$\mathcal{S} \in \{0,1\}^{M \times N}$	Binary response matrix; $\mathcal{S}_{ij}{=}1$ iff LLM i answers item j correctly
$\mathcal{Q} \in \{0,1\}^{N \times K}$	Q-matrix; $q_{jk}{=}1$ iff item j tests skill k
$\boldsymbol{\theta}_i \in [0,1]^K$	Skill mastery profile of LLM i
$\boldsymbol{d}_j \in [0,1]^K$	Item difficulty per skill
$\alpha_j \in [0,1]$	Item discrimination
$\boldsymbol{e}_j \in \mathbb{R}^{768}$	Frozen SBERT text embedding of item j
$\sigma(\cdot)$	Sigmoid function

C SKILL CLUSTER EXAMPLES

Table 5 shows four representative skill clusters with example items. Each cluster groups items that share a common cognitive requirement despite coming from different benchmarks or subtasks. For instance, the *Quadratic Formula* cluster contains 451 items dominated by MATH algebra problems but also includes 20 GPQA items that happen to require algebraic manipulation.

Divergent model performance. Table 6 illustrates how skill profiles explain item-level performance differences. On a BBH book-trading problem that requires both *sequential event tracking* and *object*

Table 5: Representative skill clusters with example items. Each cluster aggregates items from multiple benchmarks that share a common skill requirement.

Skill cluster	Items	Example items
Quadratic Formula (MATH-primary, 429/451)	451	"Find the domain of $f(x) = \sqrt{\sqrt{x^2 - 16} - 3}$" "If $-6 \leq a \leq -2$ and $3 \leq b \leq 5$, find the greatest value of $\frac{a+b}{a}$"
Truthfulness of Nested Statements (BBH-primary, 1836/1880)	1,880	"Then he tells Billy: 'Be sure that you do not come in at all tomorrow morning' " (causal judgement) "Teressa lies. Willian says Teressa lies. Jamey says Willian lies. Ryan says Jamey lies. Kandi says Ryan lies. Does Kandi tell the truth?" (web of lies)
Structural Isomers in Organic Compounds (GPQA-primary, 442/488)	488	"Identify product D [from NMR spectrum]: (A) 2-bromonaphthalene (B) 2-bromo-1,2,3,4-tetrahydronaphthalene. . . "
Word Count Constraints (IFEval-primary, 414/418)	418	"Write a social media post about the hajj pilgrimage. Use all lowercase and include the word 'story' at least twice"

Table 6: Accuracy by mastery group on two held-out items. The BBH item is conjunctive (needing both skills). The IFEval item is compensatory (either skill suffices).

	Both	Skill 1	Skill 2	Neither
BBH (conjunctive)	0.74	0.33	0.22	0.10
IFEval (compensatory)	0.93	0.97	0.42	0.12

trade tracking (conjunctive), models mastering both skills solve it 74% of the time while models mastering only one skill drop to 22–33%. For example, Llama-3.1-Tulu-3-70B (masters both) answers correctly, while Llama-3.2-Korean-5B (masters only event tracking) fails. On a compensatory IFEval item requiring *word count constraints* or *Urdu dialogue adaptation*, mastering either skill suffices: accuracy is 97% with word count skill alone and 42% with Urdu skill alone, both far above the 12% baseline.

D Q-Matrix Construction Details

Skill extraction prompt. For each item we prompt GPT-4o-mini at temperature 0 with the benchmark, the subtask and the item text, and ask for two to four skills, each with a label, the cognitive process and the content domain, together with the single most important skill, which we call the primary skill. Labels on the list of generic labels in the prompt are removed after extraction. The system prompt and the user template are reproduced below; long lines are wrapped for display. We extract the full skill list rather than only the primary skill, yielding a mean of 2.9 skills per item and $13{,}276$ unique skill strings across $9{,}523$ items. The primary skill alone is too coarse: it covers only the headline category and produces Q-matrix rows with mean row-sum 1.0, which under-constrains the diagnostic model.

System prompt.

```
You are a psychometrician designing a Q-matrix for cognitive diagnostic
   assessment of AI systems. Your task: identify the specific cognitive
   skills each test item measures.

RULES:
1. Each skill must be a compound phrase (5-10 words):
   [specific_cognitive_process] + [specific_content_domain]
2. Extract 2-4 skills per item. At least one must be domain-specific.
3. FORBIDDEN generic labels (will be rejected):
```

```
   - "reading comprehension", "logical reasoning", "critical thinking"
   - "mathematical reasoning", "problem solving", "analytical thinking"
   - "scientific knowledge", "attention to detail", "numerical
     reasoning"
   - Any label that would apply to >15% of items in the benchmark
4. Specificity test: would fewer than 100 items in a 10,000-item test
   share this exact skill? If not, decompose further.
5. Use lowercase snake_case labels.

GOOD examples by benchmark type:

MATH: "factoring_higher_degree_polynomials_over_integers",
   "applying_pigeonhole_principle_to_combinatorial_bounds",
   "computing_modular_arithmetic_in_residue_classes"

BBH: "tracing_boolean_operator_precedence_in_nested_expressions",
   "identifying_causal_direction_from_correlational_evidence",
   "tracking_object_positions_through_spatial_transformations"

GPQA: "applying_gauss_law_to_cylindrical_charge_distributions",
   "predicting_reaction_products_via_retrosynthetic_analysis",
   "interpreting_phylogenetic_trees_from_molecular_sequence_data"

MuSR: "integrating_alibis_and_motives_to_identify_suspects",
   "tracking_object_locations_across_sequential_room_transfers",
   "resolving_team_allocation_under_mutual_exclusion_constraints"

IFEval: "enforcing_exact_word_count_in_structured_output",
   "maintaining_consistent_formatting_across_nested_lists",
   "embedding_required_keywords_while_preserving_coherence"

BAD examples (too generic, rejected):
- "problem solving" -> decompose to
   "applying_substitution_to_solve_nonlinear_systems"
- "reading comprehension" -> decompose to
   "extracting_temporal_ordering_from_narrative_clues"
- "scientific knowledge" -> decompose to
   "applying_conservation_of_angular_momentum_to_rotating_bodies"
```

User template. The benchmark name, subtask and item text replace the placeholders in braces.

```
Benchmark: {benchmark} | Subtask: {subtask}

Item:
{question_text}

Respond in JSON:
{
  "skills": [
    {
      "label": "5-10 word compound skill in snake_case",
      "process": "the specific cognitive operation",
      "domain": "the specific content area"
    }
  ],
  "primary_skill": "label of the single most important skill"
}
```

Embedding and clustering. The 13,276 unique skill strings are embedded with the `all-mpnet-base-v2` sentence transformer into $\mathbb{R}^{768}$ and L_2-normalised. We compare three clustering families: HDBSCAN after UMAP reduction to 5D, K-Means, and HAC with cosine distance and average linkage. We select HAC because it yields the highest downstream test AUC at every K we tried, despite a lower silhouette score (0.053 at $K{=}50$ vs. 0.144 for K-Means and

0.160 for HDBSCAN at K=281). Intrinsic clustering quality is therefore not a reliable proxy for diagnostic utility.

Coverage fix. Our first pipeline iteration assigned each item to a single cluster via the primary skill only, which left 43% of clusters with fewer than 10 items and 10 empty columns in $\mathbf{Q}$ at K=100. Switching to the multi-skill assignment described above removed all empty columns, reduced the fraction of under-populated clusters to 36% and raised the mean column-sum from 95 to 139 items per skill.

Reference-answer ablation. Our default skill extraction does not use reference solutions. To test whether showing the extractor a reference solution changes the result, we ran two new extractions for the 2,470 MATH and GPQA items that have a reference solution, one from the question and one from the question plus the solution. Both use the same prompt, the same extraction model and the same question text, and they are separate from the extraction that produced the Q-matrix. The solution changes which skills are named: the token-level Jaccard overlap between the two label sets is 0.37, against 0.73 between two independent question-only runs. A blind judge (GPT-4.1-mini) compared both label sets on 600 items, each in both presentation orders. We retain the 357 items with a consistent, non-tied preference in both orders; the remaining judgments do not enter the percentages below. Among the retained items the judge preferred the solution-aware labels in 49.6% and the question-only labels in 50.4%, so it showed no clear aggregate preference for either extraction condition. Within the retained subset the preference differs by benchmark: question-only labels are preferred on MATH (59%) and solution-aware labels on GPQA (61%). We keep question-only extraction, which also applies to item banks without reference solutions.

E Training Details

The text-conditioned NCDM is trained end-to-end with binary cross-entropy loss on the 80% training split of the response matrix, which covers 7,618 training items and all 3,811 LLMs. The skill mastery embedding $\boldsymbol{\theta}_i$, the item difficulty and discrimination heads $\mathbf{W}_k \in \mathbb{R}^{K\times 768}$ and $\mathbf{w}_e \in \mathbb{R}^{768}$, and the PosLinear layers all use Xavier normal initialisation. The PosLinear prediction network has hidden widths $K\to512\to256\to1$; non-negativity is enforced by $W^+ = \max(W, 0)$ at each forward pass. We optimise with Adam at learning rate 0.002 for 15 epochs, with dropout 0.5 after each hidden layer. The main model and the K-sweep models use batch size 1024. The five-seed runs and the interaction-wise model use batch size 64. Model selection uses validation AUC on a held-out triplet split of 10% of the training pairs, keeping the weights of the best-AUC epoch. Training the main model takes about three hours on a single NVIDIA RTX 2080 Ti. The default seed is 42. Multi-seed stability uses seeds 42–46, see Appendix L.

F Full K-Sweep Results

Table 7 reports a sweep over $K \in \{50, 100, 150, 200, 300, 400, 500\}$. Test AUC is essentially flat (0.717–0.721), confirming that the diagnostic fit is robust to clustering granularity rather than dependent on a particular elbow. Routing Acc@1 improves modestly from 0.640 at K=50 to 0.676 at K=500, and is 0.657 at K=100. We adopt K=100 as the default because it preserves interpretable, named clusters while using about four times fewer parameters than K=500.

G Profiling Unseen LLMs: Details

Network. The frozen network is trained with the recipe of the main model in Appendix E (seed 42, batch size 1024, 15 epochs), using only the training-item responses of the 3,048 training LLMs.

Estimation. Each profile is represented by unconstrained raw parameters $\mathbf{r}$, with $\boldsymbol{\theta} = \mathrm{sigmoid}(\mathbf{r})$. The prior on $\mathbf{r}$ is a multivariate normal distribution. Its mean $\boldsymbol{\mu}$ is the mean raw profile of the 3,048 training LLMs; its covariance $\boldsymbol{\Sigma}$ is their empirical covariance shrunk by 10% towards its diagonal. Let A be the set of answered items, y_j indicate whether item j was answered correctly, and p_j be the

Table 7: Effect of the number of skill clusters K on diagnostic AUC and routing Acc@k. Diagnostic fit is essentially constant across K, and routing accuracy rises modestly with K. Each row is a single training run with batch size 1024; the K=100 row is the main model, so its test AUC differs from the five-seed mean in Table 2.

K	Test AUC	Acc@1	Acc@3	Acc@5	Acc@10
50	0.720	0.640	0.783	0.829	0.877
100	0.717	0.657	0.780	0.829	0.875
150	0.719	0.654	0.778	0.831	0.887
200	0.716	0.661	0.797	0.846	0.885
300	0.719	0.672	0.794	0.843	0.882
400	0.721	0.669	0.802	0.841	0.887
500	0.720	0.676	0.804	0.847	0.896

predicted probability of correctness from Equation 1. We estimate $\mathbf{r}$ by minimising

$$-\sum_{j\in A}\left[y_j \log p_j + (1-y_j)\log(1-p_j)\right] \;+\; \tfrac{1}{2}(\mathbf{r}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{r}-\boldsymbol{\mu}). \tag{5}$$

The first term measures fit to the observed answers; the second penalises departures from profiles supported by the prior. We optimise with L-BFGS, starting from $\boldsymbol{\mu}$. With no answers, the raw estimate is $\boldsymbol{\mu}$ and the skill profile is its sigmoid transform. In a check on the first 20 LLMs using all 7,618 pool answers, fits with and without the prior have a mean absolute difference of 0.005 in their test-item probabilities. During adaptive selection, each update starts from the previous raw estimate. At each evaluated budget, we refit from $\boldsymbol{\mu}$, using the same procedure for random and adaptive selection.

Evaluation. We use the first 200 of the 763 held-out LLMs. Random-selection metrics are averaged over three seeds; adaptive selection is deterministic for a given LLM and network. Both selection methods are evaluated at the same answer budgets up to 500. Table 8 lists the values shown in Figure 4. At each tested budget from 5 to 500 answers, paired 95% bootstrap intervals over LLMs favour adaptive selection on both measures: the adaptive-minus-random difference is below zero for benchmark-score error and above zero for profile agreement.

Reference profiles and robustness. Profile agreement uses a reference estimated from all pool answers without the prior, avoiding a shared prior term in the reference and the estimates being evaluated. This reference is itself an estimated profile. We assign zero agreement to a skill whose estimates are constant across LLMs; this includes the zero-answer setting, where Pearson correlation is undefined. Using a reference estimated with the prior, or restricting the measure to the 50 skills with at least 20 pool items, changes the agreement values but preserves the ordering of the compared methods.

Per-LLM AUC. Test-item AUC changes little in this setting, from 0.68 with no answers to 0.69 with all answers. The zero-answer prediction uses the same profile for every LLM, so reaching this AUC does not require estimating differences between their skills. We therefore use benchmark-score error and profile agreement to assess what additional answers reveal about each LLM.

H Per-Benchmark Routing

Table 9 expands the per-benchmark routing breakdown summarised in §4.1. SkillEval beats the strongest-model baseline and IRT 2PL on every benchmark, with the largest absolute gains over IRT on MuSR (+12.5 pts) and IFEval (+14.0 pts).

I Weak-Beats-Strong Inversions

Figure 6 expands the per-benchmark distribution of "weak-beats-strong" inversions summarised in §4.1. Across the 1,905 held-out test items, 723 (38%) admit at least one LLM with $\leq$ 13B parameters that out-performs the global strongest model, spanning 41 distinct skill clusters. The phenomenon is broad rather than driven by a single benchmark: BBH contributes 329, MATH 156, GPQA 155,

Table 8: Profiling 200 held-out LLMs from N answers. Error: mean absolute benchmark-score error over LLMs and benchmarks. Agreement: mean per-skill Pearson correlation with profiles estimated from all pool answers without the prior. AUC: mean per-LLM test-item AUC. Both selection methods are evaluated up to 500 answers; random-selection results are averaged over three seeds. The all-answers error and AUC use the estimator with the prior.

	Error			**Agreement**		**AUC**	
N	Random	Adaptive	Count	Random	Adaptive	Random	Adaptive
0	0.095	0.095	0.095	0.000	0.000	0.682	0.682
5	0.087	0.079	0.202	0.207	0.301	0.682	0.684
10	0.082	0.068	0.218	0.252	0.351	0.682	0.686
20	0.075	0.061	0.217	0.329	0.417	0.682	0.687
50	0.068	0.056	0.154	0.396	0.470	0.684	0.688
100	0.061	0.053	0.107	0.451	0.532	0.686	0.689
200	0.055	0.048	0.075	0.502	0.595	0.688	0.690
500	0.049	0.046	0.052	0.570	0.655	0.690	0.691
All 7,618	0.042					0.693	

Table 9: Routing Acc@1 by source benchmark on the held-out test items. **Strongest** is the train-selected top-1 LLM. SkillEval beats the single-LLM baseline and IRT 2PL on all five benchmarks.

Benchmark	n	**Random**	**Strongest**	**IRT 2PL**	**SkillEval**
MATH	264	0.152	0.375	0.530	**0.542**
BBH	1159	0.471	0.716	0.690	**0.735**
GPQA	246	0.280	0.370	0.402	**0.407**
MuSR	136	0.301	0.529	0.419	**0.544**
IFEval	100	0.450	0.810	0.680	**0.820**
Overall	1905	0.405	0.616	0.611	**0.657**

MuSR 64, IFEval 19. This is the headroom that skill-conditional routing exploits and scalar-ability routers cannot reach.

J Cross-Benchmark Transfer

To probe whether the learned skill space generalises across benchmark families, we hold out two entire benchmarks (MuSR and IFEval) and train SkillEval on the remaining three (MATH, BBH, GPQA), using the same K=100 HAC clusters, M=3,811 LLMs, and 15-epoch Adam schedule as the main run. The training set covers 8,231 items; the held-out evaluation set covers 1,292 items.

Aggregate diagnostic fit on the held-out items collapses to AUC 0.469 (test accuracy 0.460), indicating that on combined unseen-benchmark items the model is no better than chance. The picture sharpens when split by benchmark (Table 10). On MuSR, where the multi-step reasoning skills overlap with BBH, transfer is intact: SkillEval reaches Acc@1 0.470, matching the strongest-model baseline (0.467) but below the full-training reference (0.544). On IFEval, however, SkillEval drops to 0.410, below random (0.437) and 32 points below the strongest model (0.731).

Skill coverage in the transfer experiment. With MuSR and IFEval held out, 15 of the 100 skills have no training item. These skills were defined during Q-matrix construction on the full item bank, but their mastery parameters receive no response-based training in this experiment. IFEval items are assigned to 62 skills, 14 of which are among these 15, and 23% of IFEval items involve at least one of them; for MuSR the share is below 1%. This coverage gap is consistent with the larger failure on IFEval, but it does not explain it fully, since most IFEval items involve only skills that were trained. We therefore report the IFEval result as a limit of cross-benchmark transfer: deployments should include the target benchmark family in the training mixture rather than rely on zero-shot transfer.

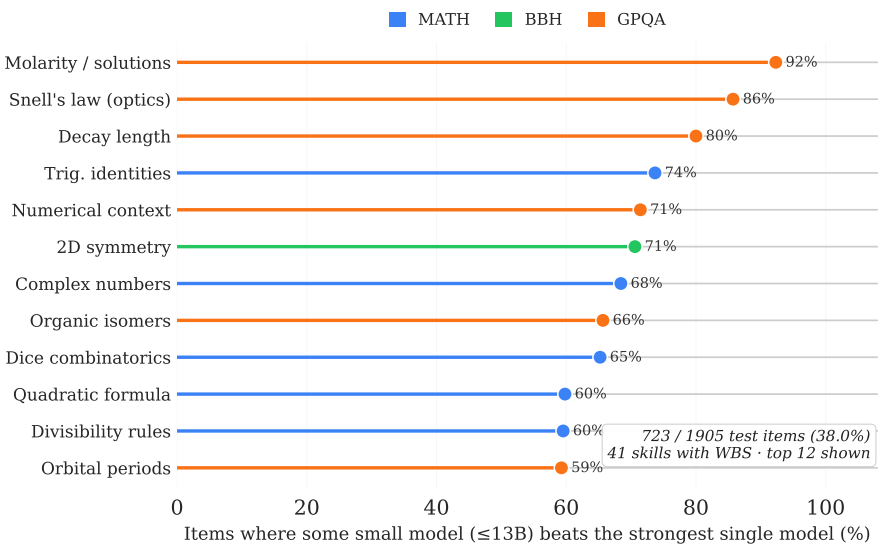

Figure 6: Distribution of "weak-beats-strong" items by source benchmark. 38% of held-out items (n=723/1,905) admit at least one $\leq$ 13B LLM that out-performs the global strongest model, spanning 41 skill clusters: BBH 329, MATH 156, GPQA 155, MuSR 64, IFEval 19.

Table 10: Cross-benchmark transfer with MuSR and IFEval held out entirely. "Reference" is the full-training Acc@1 from §4.1. Transfer succeeds on MuSR (overlap with BBH multi-step reasoning) but fails on IFEval, where 14 of the 62 assigned skills receive no training items.

Held-out	n	**Random**	**Strongest**	**SkillEval (transfer)**	**Reference**
MuSR	756	0.415	0.467	0.470	0.544
IFEval	536	0.437	0.731	0.410	0.820
Combined	1292	0.400	0.577	0.445	—

K Controls for the Validity of Θ

Section 3.2 reports the correlation between Θ and held-out per-skill accuracy. Such a correlation could arise if Θ only reflected how strong an LLM is overall. For each skill k we therefore compute the partial correlation, across LLMs, between the mastery θ_k and the held-out accuracy on skill k after regressing out the LLM's overall training accuracy. Over the 41 skills with at least ten held-out items, the mean partial correlation is 0.35, averaged over five seeds. For a descriptive null comparison, we fix the seed-42 mastery estimates and permute held-out item-to-skill assignments within each benchmark, preserving each skill's held-out size and benchmark composition; the mean partial correlation over 20 such permutations is 0.22. Controlling instead for the LLM's five per-benchmark training accuracies gives 0.135, against a placebo mean of -0.02. Controlling for the finer profile of per-subtask training accuracies (38 subtasks) reduces the mean partial correlation to 0.021, indicating substantial overlap with subtask structure. These comparisons indicate that Θ carries information beyond overall and benchmark-level accuracy; they do not separate it from subtask-level accuracy.

L Multi-Seed Stability

We train SkillEval with five seeds (42–46) on identical data splits to assess stability. Test AUC is highly stable: 0.7215 ± 0.0008 across seeds (mean and sample standard deviation, here and below). The learned mastery profiles are nearly identical: pairwise Pearson r=0.991 between flattened θ matrices, and per-skill r=0.959 averaged over all 100 dimensions. Routing Acc@1 exhibits higher variance (0.617 ± 0.019) because argmax over 3,811 LLMs is sensitive to small weight perturbations. The main model in Table 3 achieves Acc@1=0.657. It is trained with batch size 1024, whereas the

five seeds here use batch size 64. At batch size 1024, seeds 42, 43 and 44 give 0.655, 0.651 and 0.658 (0.655 ± 0.003). These results use different batch sizes and should not be pooled into a single estimate of variability across seeds.

Comparison with IRTNet. Table 11 compares the ability matrices for every pair of runs within each reported configuration: five seeds for SKILLEVAL and three for each IRTNet configuration. *Flat* correlates all entries of the two ability matrices after arranging them into vectors. *Per-LLM* correlates each LLM's ability profile between runs and averages over LLMs. These summaries can be high when some dimensions consistently take larger values than others. *Per-dim* instead compares values across LLMs separately for each dimension, then averages the correlations over dimensions. It therefore tests whether each dimension assigns similar relative values to LLMs across runs. Because different runs may learn different coordinate systems, we also compare dimensions after Procrustes alignment. *Raw* uses the dimensions as learned. *Orthogonal* applies the rotation or reflection that best aligns one run's ability matrix with the other (Schönemann, 1966), and *Full* additionally centres and rescales the matrices before alignment (Gower, 1975). Across the reported IRTNet configurations and both alignments, mean per-dimension correlations range from 0.10 to 0.43. The tested alignments therefore do not fully account for the differences between IRTNet's learned dimensions across runs.

Table 11: Mean Pearson correlations between ability matrices Θ across run pairs, with standard deviations across pairs in parentheses.

Model	Flat	Per-LLM	Per-dim		
			Raw	Orthogonal	Full
IRTNet (d=232)	0.02 (0.06)	0.02 (0.06)	0.00 (0.01)	0.43 (0.22)	0.43 (0.21)
IRTNet (d=512)	0.00 (0.04)	0.00 (0.04)	0.00 (0.00)	0.17 (0.02)	0.17 (0.01)
IRTNet (d=1024)	0.01 (0.00)	0.01 (0.00)	0.00 (0.00)	0.10 (0.02)	0.10 (0.02)
SKILLEVAL (K = 100)	**0.99** (0.00)	**0.99** (0.01)	**0.96** (0.00)	**0.96** (0.00)	**0.96** (0.00)

M EMPIRICAL CO-MASTERY PATTERNS

Mastery and link rule. This exploratory analysis asks whether LLMs that master one skill also tend to master another, even when the reverse is less common. We find such asymmetric co-mastery patterns across most skills, with similar patterns visible in the observed responses. In Equation 1, estimated mastery θ is interpreted relative to item difficulty, so its value alone does not specify expected accuracy on a skill's items. For this analysis, we classify an LLM as mastering a skill when its predicted probability of a correct answer, averaged over the items assigned to that skill, exceeds 0.5, that is, when the model expects it to answer more than half of those items correctly.

For two skills A and B, let $P(A \mid B)$ denote the share of LLMs mastering A among those mastering B. For example, 52 of the 55 LLMs mastering *combinatorics of dice rolls* also master *divisibility rules for factorials*, so $P(\text{divisibility} \mid \text{dice}) = 52/55 \approx 0.95$. In the reverse direction, only 52 of the 167 LLMs mastering divisibility rules also master dice-roll combinatorics, so $P(\text{dice} \mid \text{divisibility}) = 52/167 \approx 0.31$. Almost every LLM that masters the dice-roll skill thus also masters the divisibility skill, whereas most LLMs that master the divisibility skill do not master the dice-roll skill. We represent this pattern by a link from divisibility rules to dice rolls.

We draw a link from A to B when two conditions hold. First, we require $P(A \mid B) - P(B \mid A) > 0.3$, capturing an asymmetry like the one above. Second, we require a strong association. The odds of an event are its probability divided by the probability that it does not occur, so a probability of 0.75 corresponds to odds of 3. The odds ratio compares the odds of mastering A among LLMs that master B with those among LLMs that do not. A ratio of 1 indicates no association; a ratio above 1 means mastery of A is more likely among LLMs mastering B. We require an estimated odds ratio above 10. Without this condition, unrelated skills could be linked simply because one is mastered more frequently than the other.

Odds ratio. For two skills A and B, let n_{AB} be the number of LLMs that master both, $n_{A\bar{B}}$ the number that master A but not B, $n_{\bar{A}B}$ the number that master B but not A, and $n_{\bar{A}\bar{B}}$ the number that master neither. The Haldane correction adds 0.5 to each count, which keeps the odds ratio finite

when a count is zero:

$$\mathrm{OR}(A, B) \;=\; \frac{(n_{AB} + 0.5)\,(n_{\bar{A}\bar{B}} + 0.5)}{(n_{A\bar{B}} + 0.5)\,(n_{\bar{A}B} + 0.5)}.$$

Co-mastery graph. Applying this rule to every pair of skills gives a *co-mastery graph* with skills as nodes. The graph orders two skills when following links in their direction leads from one to the other. To simplify the graph, we remove a direct link when the original graph also connects the same pair through one intermediate skill. This leaves 304 links connecting 86 of the 100 skills; some retained links may still be implied by longer paths. Of these 86 skills, 18 have no incoming link and 13 have no outgoing link. Every link points from a more commonly mastered skill to a less commonly mastered one, because $P(A \mid B)$ can exceed $P(B \mid A)$ only when more LLMs master A than B. Its direction therefore follows from the construction and does not show that A must be learned before B. Repeating the construction and link removal with a mastery cut-off of 0.4 instead of 0.5 gives 391 links connecting 89 skills, and with 0.6 it gives 209 links connecting 68 skills. Only 29 direct links appear at all three cut-offs, so individual links are sensitive to the cut-off.

Checks. Two checks support these patterns. First, we construct a second graph using the same LLMs, items and skill assignments, but classify an LLM as mastering a skill if it answers more than half of that skill's items correctly. This graph orders 1,325 pairs. The co-mastery graph orders 983 in the same direction and one in the opposite direction, leaving the rest unordered. Wherever both graphs order a pair, they agree in all but one case. Second, we independently shuffle each skill's mastery labels across LLMs, preserving the share mastering each skill while disrupting the observed associations. Before link removal, the rule finds at most 51 links in any of 200 shuffles, compared with 1,031 in the original profiles. This supports associations beyond individual skill frequencies. Across the 304 retained links, the median of $P(A \mid B)$ is 0.87 and the median of $P(B \mid A)$ is 0.26. A benchmark average does not expose these relationships between named skills. For dataset design, the graph can suggest skills to assess within groups of LLMs sharing a strength, as dice-roll combinatorics distinguishes among the 167 LLMs that master divisibility rules.

Cross-family overlap. We also apply the link rule separately to each model family with at least 100 LLMs: Llama, Qwen, Mistral, Gemma and Phi. We compare the resulting link sets before removing any links. For two families, the Jaccard index is the number of links shared by both divided by the number found in at least one. Across the ten family pairs, it ranges from 0.09 to 0.31. The detected links are therefore only partly shared across the sampled families; whether these patterns persist in other LLM populations remains to be tested.

N Benchmark Redundancy Details

Figure 7 shows skill coverage per benchmark: dark segments mark skills unique to that benchmark, light segments mark skills shared with at least one other benchmark. IFEval contributes the most unique skills (13 of 62), while BBH contributes only 1 unique skill of 42. MATH and IFEval together cover 87 of 100 curated skills.

Figure 8 shows the pairwise coverage matrix: cell (i, j) reports the fraction of benchmark i's skills also covered by benchmark j. MuSR is the most redundant row: 82% of its skills are covered by BBH, and 82% by IFEval. IFEval is the most independent row, with the lowest average overlap with the other benchmarks.

O Differences Between Base and Instruction-Tuned Models: Details

Figure 9 reports estimated mastery differences by model family and benchmark, with each skill assigned to the benchmark that contributes most of its items. Mean differences over IFEval skills are positive for every evaluated family. The MuSR column is uniformly negative, and the MATH/GPQA columns are near-zero or slightly negative for most families. Llama shows the broadest gains; Qwen is essentially flat.

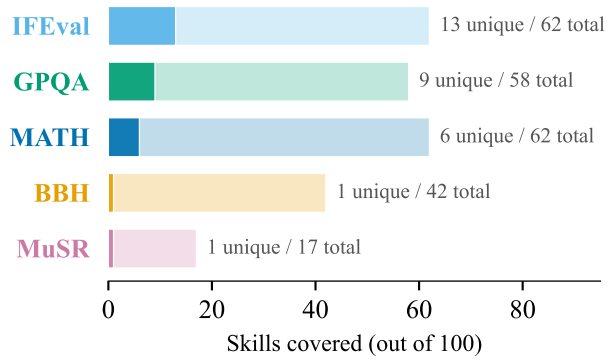

Figure 7: Skill coverage per benchmark. Dark segments show skills unique to that benchmark; light segments show shared skills. IFEval contributes 13 unique skills, while BBH has only 1. MATH + IFEval together cover $87/100$ skills.

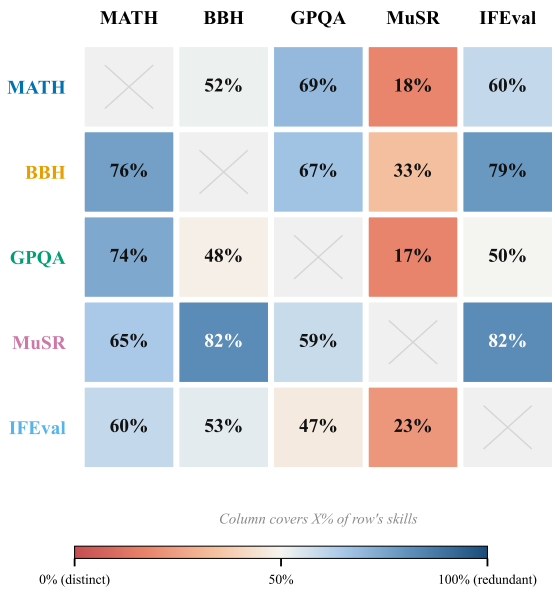

Figure 8: Pairwise skill coverage matrix. Cell (i, j): fraction of row benchmark's skills also covered by column benchmark. Red = low overlap (distinct), blue = high overlap (redundant). MuSR's skills are 82% covered by BBH and IFEval.

P LIMITATIONS

Scope of the new-LLM experiments. The new-LLM results use 200 of the 763 held-out LLMs and one trained network. We evaluate one adaptive selection rule at answer budgets up to 500. The prior is estimated from profiles produced by that network, while observed scores for the smaller benchmarks are based on relatively few test items. The results describe the evaluated LLMs, network and item pool; they do not establish an advantage for adaptive selection in every setting.

Co-mastery vs. causal prerequisites. The co-mastery graph in Appendix M is constructed from statistical co-mastery ($\mathrm{OR} > 10$), not causal dependency. The observed structure likely reflects a mix of skill-difficulty ordering and shared training-data signal rather than experimentally verifiable cognitive prerequisites. Establishing causal prerequisite relations would require intervention experi-

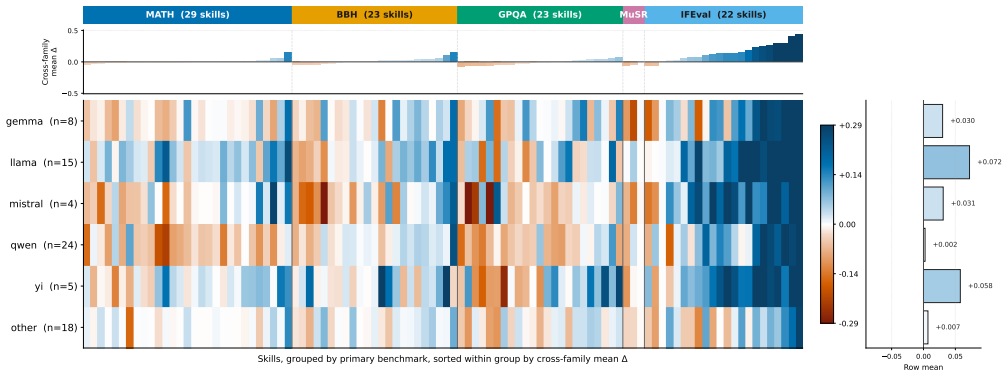

Figure 9: Mean estimated mastery difference (instruction-tuned−base) per model family and benchmark. Blue = improved, red = degraded. The IFEval column is positive for every evaluated family; the MuSR column is negative for every evaluated family.

ments (skill-specific fine-tuning) beyond our scope; we therefore phrase all structural claims in terms of co-mastery.

Q-matrix inter-annotator agreement. The Q-matrix is constructed semi-automatically via LLM-proposed skill phrases clustered by SBERT and HAC. Cluster choice is validated by downstream NCDM AUC rather than by human adjudication. A human validation of a sample of skill–question assignments is reported in Section 3.1. It tests whether an assigned skill is needed, not whether needed skills are missing. Items near cluster boundaries should be interpreted with additional caution. We sweep the number of skill clusters K and report sensitivity to K, but we do not separately vary the clustering algorithm (HAC), the skill-extraction prompt, or the sentence-embedding source (SBERT); we leave a joint pipeline-robustness study to future work.

Evaluation scope. Our item-wise OOD setting holds out all responses to the test items while retaining the same benchmarks and LLM population. We treat the Q-matrix as part of the test design: the skill taxonomy and Q-matrix are constructed before this split, using question text and benchmark/subtask metadata from the complete item bank, including test items. Skill extraction and clustering use item content rather than the evaluated LLMs' responses. However, the clustering method and the number of clusters were chosen with reference to downstream performance (Appendices D and F). Item difficulty and discrimination for test items are predicted from their text by projections trained on training items only. The evaluation is therefore transductive with respect to the skill representation, while test-item responses are excluded from parameter fitting: it measures prediction of held-out responses within a fixed, item-bank-derived taxonomy. Generalization remains limited. IRTNet outperforms SKILLEVAL on held-out-item prediction (Table 2). When MuSR and IFEval are jointly excluded from response-based training, observed routing accuracy on MuSR is similar to that of the single-model baseline selected by training accuracy (0.470 versus 0.467), whereas IFEval routing falls below the random-routing baseline (0.410 versus 0.437; Appendix J). These results limit the scope of our routing and profiling claims; reliable transfer to new benchmark domains or skill families remains to be established.

Benchmark coverage. SkillEval covers five English-language benchmarks (MATH, BBH, GPQA, MuSR, IFEval). Notable gaps include multilingual reasoning, tool use, long-context retrieval, safety/red-team benchmarks, and open-ended generation. The Q-matrix pipeline is designed to extend to new benchmarks without retraining existing skill clusters: adding a new benchmark requires running the LLM skill extraction pipeline on its items, assigning each extracted skill to its nearest existing cluster (or expanding K), and retraining NCDM.