PROBING MEMES IN LLMS: A PARADIGM FOR THE ENTANGLED EVALUATION WORLD

Anonymous authors
Paper under double-blind review

ABSTRACT

Current evaluations of large language models (LLMs) often treat datasets and models in isolation, obscuring phenomena that only emerge from their collective interaction. Items in datasets are reduced to labeled entries, disregarding the multidimensional properties they reveal when examined across model populations. Models, in turn, are summarized by overall scores such as accuracy, neglecting performance patterns that can only be captured through diverse data item interactions. To address this gap, this paper conceptualizes LLMs as composed of invisible memes, understood as cultural genes in the sense of Dawkins that function as replicating units of knowledge and behavior. Building on this perspective, the Probing Memes paradigm reconceptualizes evaluation as an entangled world of models and data. At its core lies the perception matrix, which captures interaction patterns and enables two complementary abstractions: probe properties, extending dataset characterization beyond labels, and phemotypes, revealing fine-grained capability structures of models. Applied to 9 datasets and 4,507 LLMs, Probing Memes reveals hidden capability structures and reveals phenomena invisible under traditional paradigms (e.g., elite models failing on problems that most models answer easily). This paradigm not only supports more informative, extensible, and fair benchmarks but also lays the foundation for population-based evaluation of LLMs.

1 INTRODUCTION

To advance the development and understanding of large language models (LLMs), researchers have devoted sustained efforts to improving benchmark design (Hendrycks et al., 2020; 2021; Srivastava et al., 2023). On one axis, increasingly challenging or cost-efficient datasets have been introduced (Phan et al., 2025; Maia Polo et al., 2024; Schilling-Wilhelmi et al.); on another, evaluation metrics have been expanded beyond simple accuracy to capture richer dimensions of performance (Ribeiro et al., 2020; Bommasani et al., 2023; Guo et al., 2025a). These efforts aim to enhance the effectiveness of evaluation. Further improvement efforts and limitations are detailed in Appendix A. However, persistent limitations remain: current approaches typically treat models and datasets in isolation, resulting in overly coarse descriptions. As a result, evaluations often lack depth and struggle to reveal phenomena that only emerge when data and models are analyzed in a population context (Figure 1 and 2).

On the data side, individual items are usually defined only by pre-assigned labels, without further characterization of their latent properties or their ability to differentiate model capabilities. This limits the explanatory power of datasets. For example, some items exhibit riskiness, where failing them strongly correlates with broader error patterns across the dataset. On the model side, although many new evaluation metrics have been proposed, they largely broaden the range of overall evaluation scores rather than revealing the deeper structure of model capabilities. Fine-grained differences are often obscured within overall scores, yet such differences typically surface only through population-level comparisons. For instance, certain elite models that excel in overall metrics nevertheless display anomalous errors on questions that most other models solve with ease.

These phenomena highlight the inadequacy of existing evaluation paradigms. To address this gap, this paper introduces the Probing Memes paradigm. As shown in Figure 3, the paradigm situates evaluation within an entangled world jointly shaped by interactions between data and models. Here,

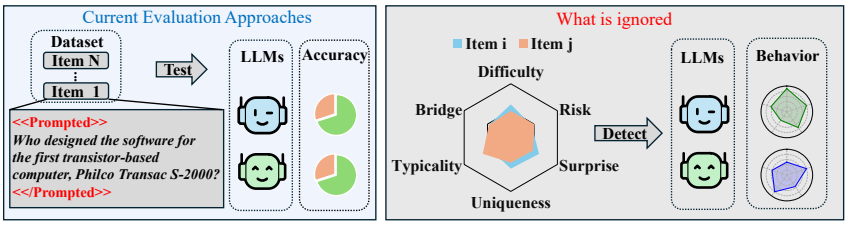

Figure 1: **Limitations of the current evaluation.** Current evaluation reveals only dataset-level accuracy across models. It neglects fine-grained attributes on both data and model sides, which are observable only through population-level interactions and thus remain hidden under accuracy-based evaluation.

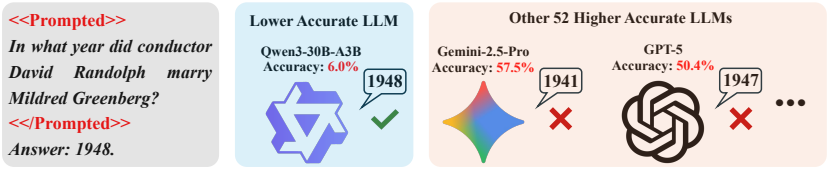

Figure 2: **A surprising case across LLMs.** Qwen3-30B-A3B, despite lower overall accuracy, succeeds on this item, whereas higher-accuracy LLMs (Gemini-2.5-Pro, GPT-5) fail.

the notion of meme is borrowed[1] and metaphorically extended to the context of LLM evaluation, denoting latent units of model capability that can be revealed through probing. From this perspective, the abilities of LLMs are conceptualized as composed of latent memes. At the same time, each data item is treated as a Meme Probe (MP) designed to elicit and expose particular aspects of these capabilities. See Appendix A.3 for information about memetics.

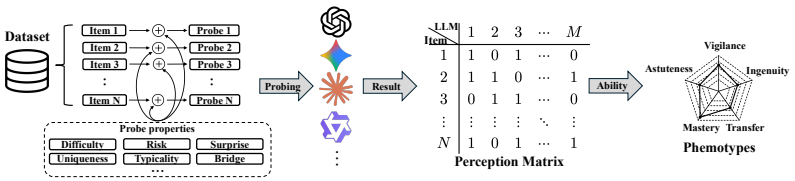

Figure 3: **Phemotype-based LLM probing framework.** Unlike traditional accuracy-focused approaches, this framework uses probes with diverse properties. By analyzing the resulting perception matrix, it better captures subtle LLM behaviors and reveals underlying abilities.

Interactions between probes and models yield a perception matrix. Analyzing this matrix enables two complementary abstractions. On the model side, latent memes can be organized into phenotypic memes (phemotypes), making structural differences in capabilities across models explicit and interpretable. On the data side, the ability of an item to elicit specific memes is captured by its Meme Probe Properties (MPPs). These properties are derived by generalizing across models and data contexts, revealing deeper characteristics of data and enabling more principled dataset optimization.

Crucially, both phemotypes and probe properties are designed to be extensible, allowing researchers to flexibly define new properties or phenotypes to meet diverse evaluation needs. In summary, Probing Memes enriches evaluation along two complementary axes: on the model side, it organizes latent capabilities into interpretable phemotypes; on the data side, it attributes probing power through MPPs. This dual abstraction moves beyond conventional reliance on overall metrics, enabling evaluation that is more flexible, fine-grained, and extensible.

[1] In *The Selfish Gene* (Dawkins, 1976), memes are described as "tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches," highlighting cultural units replicated through imitation.

The paradigm is validated through applications to 9 datasets and 4,507 LLMs. First, analyses are conducted on 28 models from 11 institutions across MATH-500, MMLU-Redux, and SimpleQA, focusing separately on probe-level and phemotype-level perspectives. At the probe level, the analysis illustrates how individual items in the entangled evaluation world can reveal fine-grained insights, such as the fact that datasets like MMLU-Redux contain a large number of seemingly simple questions that are nevertheless answered incorrectly by some elite models. At the phenotype level, the analysis reveals differences invisible to conventional evaluations, for example, models with similar accuracy may succeed on very different types of items. Second, by applying the paradigm to the Open LLM Leaderboard (Fourrier et al., 2024), which includes six datasets and 4,479 models, scalability is further demonstrated. This large-scale instantiation shows that Probing Memes sustains interpretability and flexibility at the population level. Taken together, these experiments validate the paradigm and reveal phenomena that remain hidden under conventional evaluations. Through both probe-level and phemotype-level analyses, such phenomena become explicit, underscoring the necessity of moving toward population-based, entangled evaluation.

In conclusion, the contributions of this work are threefold:

* It introduces the **Probing Memes** paradigm, which places evaluation within an entangled world shaped by data and model interactions;
* It formalizes two complementary abstractions, namely **phemotypes** and **meme probe properties**, enabling structured and extensible characterization of models and data;
* It validates the paradigm via large-scale experiments on 9 datasets and 4,507 LLMs, revealing fine-grained phenomena and insights remaining hidden under conventional evaluations.

2 The Probing Memes Paradigm

Building on the motivation outlined in Section 1, this section introduces the Probing Memes paradigm in detail. The exposition proceeds in three steps: first, by formalizing the paradigm as an evaluation paradigm within the entangled world shaped by the interaction between models and data; second, by characterizing the meme probe properties, as defined in this paradigm, that enable the detection of latent memes; and third, by defining phemotypes as structured representations of model capabilities.

2.1 Formalization of the Paradigm

The Probing Memes paradigm can be formalized by specifying data, models, and their interaction. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ denote a dataset of paired data items, where each pair consists of an input x_i and a reference output y_i. Let $\mathcal{M} = \{M_j\}_{j=1}^{m}$ be a collection of LLMs, each viewed as a mapping $M_j : \mathcal{X} \to \mathcal{O}$. For any (x_i, y_i), model M_j produces an output $o_{ij} = M_j(x_i) \in \mathcal{O}$.

A judging function $g : \mathcal{O} \times \mathcal{Y} \to \{0, 1\}$, applied to the paired outputs (o_{ij}, y_i), returns 1 if o_{ij} is judged correct with respect to y_i and 0 otherwise, yielding a *perception unit*

$$P_{ij} = g\big(M_j(x_i), y_i\big). \tag{1}$$

Collecting all results gives the *perception matrix* $P \in \{0, 1\}^{n \times m}$, where rows correspond to data items and columns to models. Specifically, each probe i is associated with a perception span P_i (i.e., the row of the perception matrix corresponding to that probe), which serves as the basis for defining higher-level probe properties and characterizes how the probe interacts with the population of models.

The perception matrix preserves the full structure of data-model interactions and further serves as the basis for deriving probe properties, which characterize the role of individual data items in revealing latent model capabilities.

2.2 Meme Probe Properties

The degree to which a probe reveals distinct facets of model capability depends on its intrinsic properties. These properties, termed *Meme Probe Properties* (MPPs), offer a structured lens for

characterizing the probing capacity of individual data items within the joint interaction of model and data populations. Statistically, the perception matrix can be viewed as a sample of the model population. Treating models as random variables drawn from a broader ensemble increases their diversity or number, enhancing the reliability of MPP estimation. In this sense, MPPs are defined at the nexus of data and models: they represent stable characteristics of data items that become increasingly precise as the model population expands. The following outlines, for each probe property, what the probe should do, followed by its definitions and notation.

Difficulty. A probe should dynamically provide a difficulty baseline based on the performance of the model population. Formally, the difficulty of the ith data item can be quantified as

$$d_i = 1 - \frac{1}{m} \sum_{j=1}^{m} P_{ij}, \tag{2}$$

where P_{ij} denotes the perception unit of model M_j on probe i as defined in Equation 1, and $m = |\mathcal{M}|$ is the number of models in the population $\mathcal{M}$. Intuitively, d_i measures the proportion of models that fail on probe i, so a higher value indicates greater difficulty relative to the population baseline.

Risk. A probe should reveal high-risk failure modes: failure on this probe is associated with elevated co-failure across many other probes. Formally, the risk of probe i is defined as

$$r_i = \frac{1}{n-1} \sum_{k \neq i} \mathrm{WJ}(i,k), \tag{3}$$

where $\mathrm{WJ}(i,k)$ denotes the weighted Jaccard similarity between the perception spans of probes i and k, given by $\mathrm{WJ}(i,k) = \sum_{j=1}^{m} I_j \mathbb{1}_{\{(1-P_{ij}) \wedge (1-P_{kj})\}} / \sum_{j=1}^{m} I_j \mathbb{1}_{\{(1-P_{ij}) \vee (1-P_{kj})\}}$, and the weight I_j of model M_j is defined as $I_j = -\ln\big(1 - \frac{1}{n}\sum_{i=1}^{n} P_{ij}\big)$.

Intuitively, $\mathrm{WJ}(i,k)$ measures how often two probes fail together relative to how often either one fails, so high risk corresponds to errors that co-occur broadly across probes. The weight I_j reduces the influence of weak models while emphasizing the contribution of stronger models, ensuring that risk is driven by informative rather than trivial failure patterns. A detailed discussion of the role of I_j and its statistical interpretation is provided in Appendix B.1.1.

Surprise. A probe should expose anomalies in which high-ability models fail on relatively easy probes, or conversely, low-ability models succeed on difficult probes. Formally, for the easy-side case, the surprise of probe i is

$$s_i^{\text{easy}} \;=\; \big(-\ln d_i\big) \cdot \frac{1}{|W_i|} \sum_{j \in W_i} a_j,$$

where d_i is the difficulty of probe i as defined in Equation 2, $W_i = \{j \mid P_{ij} = 0\}$ is the set of model indices such that M_j fails probe i, and a_j denotes the normalized accuracy of model M_j across all probes (see Appendix B.1.2).

Intuitively, s_i^{easy} becomes large when a probe is solved by most models but disproportionately failed by stronger ones, while s_i^{hard} highlights the reverse case. The formal definition of s_i^{hard} is provided in Appendix B.1.2. Finally, the overall surprise of probe i is given by

$$s_i = \tfrac{1}{2}\big(s_i^{\text{easy}} + s_i^{\text{hard}}\big). \tag{4}$$

Uniqueness. If a probe's response pattern does not materially reduce uncertainty about the responses to other probes, it should be flagged as highly unique. For consistency with the information-theoretic formulation, each probe i is not only represented by its perception span $(P_{i1}, \ldots, P_{im})$, but also viewed as a binary random variable P_i over the model population, where $P_i = 1$ indicates a correct response and $P_i = 0$ an incorrect one, and the vector entries serve as empirical samples of this variable. The uniqueness of probe i is then defined as

$$u_i = \frac{1}{n-1} \sum_{k \neq i}^{n} H(P_k \mid P_i), \tag{5}$$

where $H(P_k \mid P_i)$ is the conditional entropy of random variable P_k given P_i, estimated empirically from the samples.

Intuitively, a low u_i means that the model responses to probe i substantially reduce the uncertainty about other probes, indicating stronger representativeness; conversely, a high u_i implies that probe i provides little predictive information about others, indicating stronger uniqueness. The detailed formal definition of $H(P_k \mid P_i)$ is provided in Appendix B.1.3

To characterize the distinctiveness and commonality among probes' perception spans, this paper constructs a similarity graph from the perception matrix of all probes and applies Leiden community detection (Traag et al., 2019), yielding perception span clusters (i.e., sets of probes with highly similar perception spans).

Cluster Construction. Given two probes i and k, their similarity $\text{sim}(P_i, P_k)$ is measured by the ϕ-coefficient (see Appendix B.1.4). Here, each perception span is interpreted as a sample value of a Bernoulli random variable, whose expectation corresponds to the average difficulty of the probe. The ϕ-coefficient thus measures the correlation between two such random variables. An undirected weighted graph $G = (V, E)$ is then defined, where each node corresponds to a probe, and an edge (i, k) is included if $\text{sim}(P_i, P_k) > \tau$, with the edge weight set as the similarity value; here τ is a threshold controlling the sparsity of the graph. Applying Leiden community detection on this graph produces a partition $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ of probes into clusters of highly similar difficulty patterns. Building on this cluster structure, this paper defines the *typicality* and *bridge* properties.

Typicality. A probe should be considered a prototype if its difficulty vector shows high average similarity to other probes in the cluster. Formally, for probe $i \in C_l$, let $\mathcal{N}_i^{\text{intra}} = \{k \in C_l \mid (i, k) \in E\}$ denote the set of neighbors of i within its own cluster. The typicality of probe i is defined as

$$t_i \;=\; \frac{\sum_{k \in \mathcal{N}_i^{\text{intra}}} \text{sim}(P_i, P_k)}{|\mathcal{N}_i^{\text{intra}}|}. \tag{6}$$

Bridge. A probe should be considered a connector if its difficulty vector shows substantive similarity to probes in multiple distinct clusters. Formally, for probe $i \in C_l$, let $\mathcal{N}_i^{\text{inter}} = \{k \notin C_l \mid (i, k) \in E\}$ denote the set of neighbors of i in other clusters, and define $\kappa_i \;=\; |\{\, C_\ell \mid \exists k \in \mathcal{N}_i^{\text{inter}} \cap C_\ell \,\}|$ the number of distinct clusters spanned by probe i. Then the bridge property of probe i is defined as the product of *participation* and *strength*:

$$b_i \;=\; \underbrace{\frac{\kappa_i}{\kappa_i + \text{median}_{r \in V}\, \kappa_r}}_{\text{Participation}} \;\times\; \underbrace{\frac{1}{|\mathcal{N}_i^{\text{inter}}|} \sum\nolimits_{k \in \mathcal{N}_i^{\text{inter}}} \text{sim}(P_i, P_k)}_{\text{Strength}}. \tag{7}$$

Here, participation quantifies the extent to which a probe connects to multiple clusters, normalized by the population median, while strength captures the average cross-cluster similarity.

2.3 Phemotypes of LLMs

This subsection introduces model phemotypes, summarizing how memes are expressed across probe properties. In a nutshell, one phenotype dimension (e.g., vigilance) is constructed by combining each probe's corresponse properties into a numeric score, which the model obtains whenever it answers that probe correctly. Let $\tilde{d}_i, \tilde{r}_i, \tilde{s}_i, \tilde{t}_i, \tilde{b}_i, \tilde{u}_i \in (0, 1)$ denote normalized probe attributes, obtained via a generic normalization operator $\text{Norm}(\cdot)$. Given probe weights w, the phemotype score of model j is defined as

$$\text{Phemotype}(M_j; w) \;=\; \text{Score}\big(w;\, P_{\cdot j}\big), \tag{8}$$

where $\text{Score}(\cdot)$ denotes the weight-aggregated model score (see Appendix C.1 for details). The five concrete phemotypes differ only in their weighting schemes, summarized in Table 1.

3 Experiments and Analysis

This section introduces the derivation of probe properties and the resulting characterization of model phemotypes, as showcased within an Entangled Evaluation World.

Table 1: **Definitions of LLM phemotypes with semantic interpretation.**

Phemotype	Interpretation	Definition
Vigilance	Resist high-risk and counter-intuitive traps; maintain correctness where many models co-fail.	$w_i^{\mathrm{Vig}} = \tilde{r}_i\, \tilde{s}_i$
Mastery	Proficiency on typical yet difficult cluster-core motifs; s_i^{shr} denotes the cluster-shrink factor.	$w_i^{\mathrm{Mas}} = \tilde{t}_i\, \tilde{d}_i\, s_i^{\mathrm{shr}}$
Transfer	Generalization across clusters or prompts; success on bridging and difficult probes.	$w_i^{\mathrm{Trf}} = \tilde{b}_i\, \tilde{d}_i$
Ingenuity	Flexibility on unique and difficult probes; success on rare or non-canonical cases.	$w_i^{\mathrm{Ing}} = \tilde{u}_i\, \tilde{d}_i$
Astuteness	Avoid elite traps; identify key cues on surprising probes where common priors mislead.	$w_i^{\mathrm{Ast}} = \tilde{s}_i$

3.1 Experimental Setup

Under the proposed paradigm, this study evaluates 28 large language models from 11 providers, where models span small to large sizes. The study analyzes three reasoning modes: **default prompting (Base)**, **chain-of-thought prompting (CoT)**, and **internal reasoning (IR)**, definitions and prompts settings can be seen in Appendix F.1. These abbreviations are used consistently throughout the paper, including in figures and tables. Three widely used datasets across distinct tasks (mathematics, general knowledge, and question answering) are selected: MATH-500 (Lightman et al., 2023), MMLU-Redux (Gema et al., 2025), and SimpleQA (Wei et al., 2024). Further details and special cases appear in the Appendix F.

3.2 Probe-level Analysis

Within this evaluation paradigm, this paper performs probe-level analysis on the perception span matrix to derive well-designed probe properties for meme detection. To improve the quality of probe properties, probes whose perception spans are all ones or all zeros in the perception span matrix are excluded. For further details, refer to the Appendix F.2.

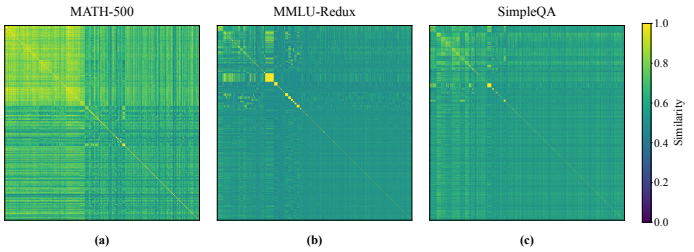

Figure 4: **Probe similarity heatmaps across datasets.**

Distributions of Probes. By calculating the perception span similarity of each pair of probes, as shown in Figure 4, the perception span similarity distribution in different datasets is significantly diverse. Concretely, MATH-500 shows higher probe similarity with clear blocks and repeated bands; by contrast, MMLU-Redux and SimpleQA show lower, more fragmented similarity with small clusters. See Appendix B.2 for more visualizations of alternative property combinations.

More than Correctness: A Unified Property Space for Probes. In the proposed paradigm, questions are treated as more than right or wrong. They are thus called probes. Each probe is represented by an expandable attribute vector and embedded in a unified property space. Figure 5 uses difficulty, uniqueness, and surprise as three axes and plots probes from each dataset. The distribution forms a funnel that narrows from easy to hard and shows a long tail of negative surprise at the hard end. The low-difficulty region is more dispersed. This suggests that easy probes can still produce unexpected

behavior in both positive and negative directions at the group level. Specifically, MMLU-Redux contains many easy probes with high surprise, indicating that many top models fail on them.

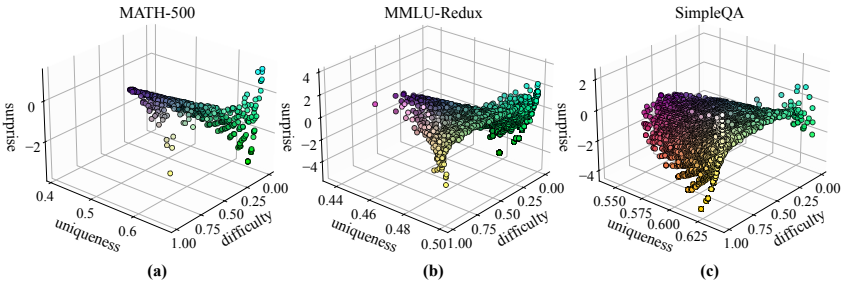

Figure 5: **Probe property distribution.** Axes show difficulty, uniqueness, and vertical surprise.

3.3 PHEMOTYPES OF LLMS

Building on the analysis of probes, six probe properties were derived to characterize the behavioral attributes of individual items. By combining the perception span matrix with these probe properties, five model-level phemotypes were computed. This construction gives rise to **A Memetic Landscape for Scrutinizing LLMs**.

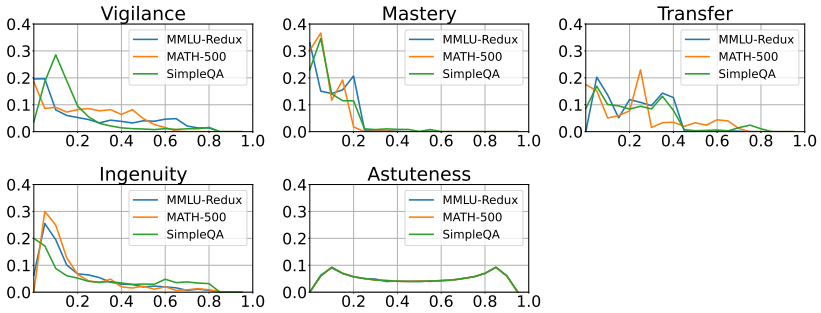

Figure 6: **Distribution of probe-level contributions to each phemotype.** This figure consists of five sub-graphs, one for each of the phemotype weight. Each sub-graph shows the distribution of a specific weight type across three different datasets. For each line, the x-axis represents 20 equal intervals within the range [0, 1]. The y-axis shows the proportion of weights that fall into that specific range. The x-coordinate of each point represents the midpoint of the interval, while the y-coordinate represents the proportion of weights in that interval.

From Probe to Phemotypes. Based on the design introduced in Section 2.3, each probe's properties are combined to detect memes in LLMs, thereby yielding the corresponding phemotypes. For each phemotype, Figure 6 plots the distributions of probe-level contribution weights across datasets, which in turn characterize a model's phemotype from its responses across probes.

Accuracy versus phemotypes. Figure 7 presents a comparison of accuracy and the five phemotype dimensions across all models under the three reasoning paradigms. Unlike the smooth trajectory of the accuracy curve, the phemotype curves exhibit nonparallel patterns with abrupt changes, crossings, and occasional reordering. These phenomena show how accuracy can alienate distinct behavioral characteristics, while phemotypes recover the latent diversity of memetic traits, providing evidence that models with the same accuracy may in fact display different behavioral patterns. It can be seen that even with comparable accuracy, gpt-4o-2024-11-20 (CoT) exhibits consistently lower phemotypes than qwen3-235b-a22b (IR), suggesting that the high accuracy of gpt-4o-2024-11-20 (CoT) relies more on routine or straightforward items, while its abilities in vigilance to traps and cross-cluster transfer are relatively weaker. The full tabular summaries and the per-dataset results are reported in Appendix C.2.

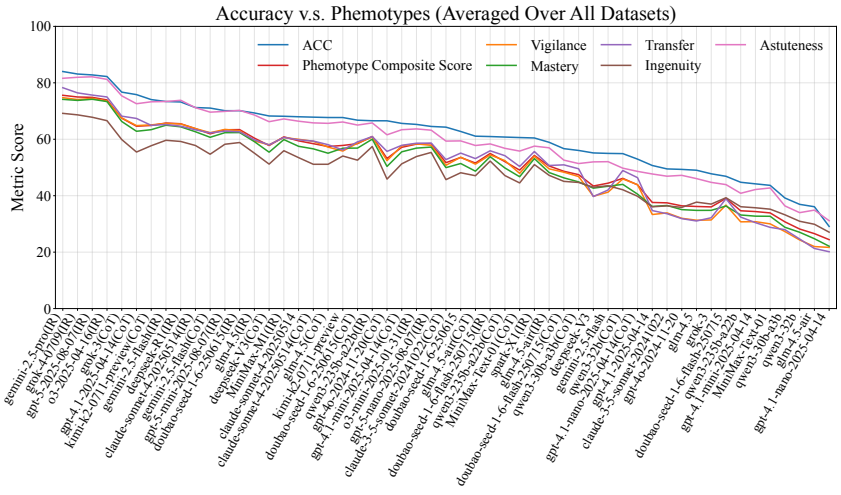

Figure 7: **Accuracy vs. phemotype scores.** Line plots show accuracy and phemotypes for all models under different reasoning modes, sorted by accuracy. The Phemotype Composite Score is the average of the five phemotypes.

4 Toward a Large Entangled Evaluation World at Scale: Open LLM Leaderboard

This section applies the Probing Memes paradigm to the Open LLM Leaderboard. Valid results from 4,479 models across six datasets are collected to construct a high-dimensional perception matrix. This matrix supports meme-level characterization of models, revealing shared and divergent behaviors. Details on the models and datasets appear in Appendix D.1.

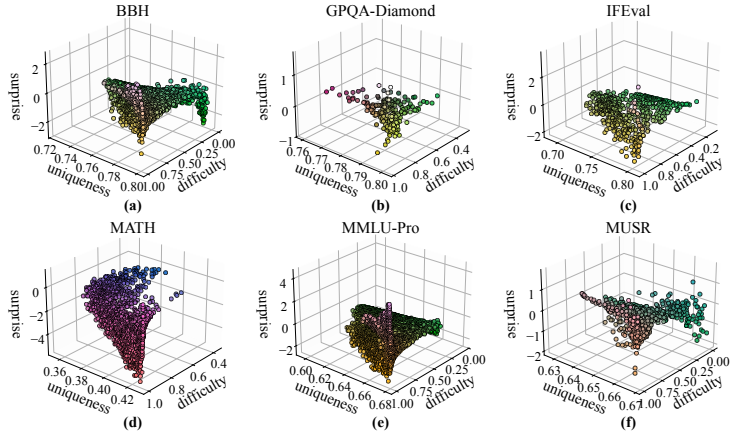

Figure 8: **Probe properties distributions across datasets of the Open LLM Leaderboard.**

Landscapes of Probe Properties. By applying open evaluation results from over 4,479 models across six datasets, the properties of each probe within each dataset are well-characterized, whose distributions are shown in Figure 8. Overall, the distributions vary across different datasets. Among these 4,479 models, the MATH and MUSR datasets contain a relatively high proportion of difficult probes. On the difficulty side of MMLU-Pro, there are many questions with high surprise scores, suggesting that a large number of models with lower performance can correctly answer these difficult probes. Moreover, the probes in IFEval, GPQA-Diamond, and BBH exhibit relatively high uniqueness. The visualization of probe similarity can be seen in Appendix D.2.

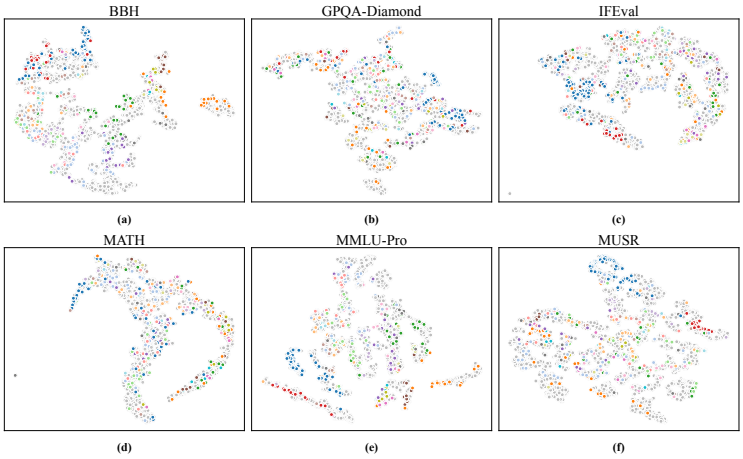

Figure 9: **Commonality and divergence among models revealed by phemotypes.** Each model is embedded with t-SNE from its five-dimensional phemotype representation. Model families are defined by the shared base model, and the 20 families with the largest numbers of models are color-coded (others are shown in gray). Nearby points indicate more similar phemotypic profiles.

Commonality and Divergence Among Models Revealed by Phemotypes. Figure 9 shows that models in the phemotype space are not uniformly distributed but instead form several clear clusters. Some datasets (e.g., MMLU-Pro) exhibit tightly packed and well-separated groups, indicating pronounced behavioral commonality and divergence. Using each model's reported base model on Hugging Face, models are organized into families. Colors indicate the top-20 families by size; unlabeled or other models are shown in gray. Notably, models from the same family tend to lie closer together in the visualization. Overall, these results demonstrate that phemotypes can uncover both similarities and differences among models. In other words, this paradigm can serve as a powerful tool to reveal similarities and differences between models, thereby **helping to investigate potential relationships in their training data, base models, and training strategies.**

5 Limitations

This paper proposes an innovative and effective evaluation paradigm. However, it still has limitations. First, the selected datasets do not comprehensively cover task types such as coding, retrieval-augmented generation (RAG), and agent workflows. Moreover, although the six properties help characterize phemotypes, more revealing property designs may exist that detect a wider range of memes. Finally, due to cost constraints, each question is queried only once per model; even with temperature set to 0 for non-reasoning models, full reproducibility is not guaranteed.

6 Conclusion

This paper reveals that the evaluation of large models is essentially an entangled world between data and models. To better explore the diverse characteristics of large models, the paper introduces the Probing Memes paradigm. It conceptualizes LLMs, drawing on memetics, as collections of invisible memes. Through interactions between data and models, calibrated probe properties detect these memes and infer each model's phemotype, thereby revealing hidden behavioral traits. Evidence comes first from 28 models tested under three reasoning modes across three datasets, revealing the diversity of models and probes that is obscured by traditional evaluation paradigms. The framework is then applied to the larger entangled world of the Open LLM Leaderboard, demonstrating behavioral similarities and divergences among thousands of models. The Probing Memes paradigm offers a scalable, extensible way to evaluate LLMs: calibrated probes yield interpretable phemotype profiles, enable cross-model comparisons, and expose failure modes that accuracy alone obscures.

REFERENCES

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

Asunción Álvarez. Three memetic theories of technology. 2005.

Anthropic. Claude 3.5 sonnet model card addendum, 2024. Accessed: 2024-06-23.

Anthropic. Introducing claude 4, 2025. Accessed: 2025-05-23.

Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Frank B. Baker and Seock-Ho Kim (eds.). *Item Response Theory: Parameter Estimation Techniques*. CRC Press, 2 edition. doi: 10.1201/9781482276725.

Aldous Birchall. Parrots are all you need: A memetic framework for apparent reasoning in llms.

Susan J Blackmore. *The meme machine*, volume 25. Oxford Paperbacks, 2000.

Rishi Bommasani, Percy Liang, and Tony Lee. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525(1):140–146, 2023.

William F Bradley. Llms and the madness of crowds. *arXiv preprint arXiv:2411.01539*, 2024.

ByteDance. doubao-seed-1-6-250615, 2025a. Accessed: 2025-06-15.

ByteDance. doubao-seed-1-6-flash-250715, 2025b. Accessed: 2025-07-15.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Richard Dawkins. The selfish gene, 1976.

Harriet Farlow, Claudio Ceruti, Matthew Garratt, Gavin Mount, and Timothy Lynar. Memes in the machine: Ideological propagation in large language models. In *2024 IEEE 15th International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 000017–000024, 2024. doi: 10.1109/CogInfoCom63007.2024.10894714.

Ivan Fomin. Memes, genes, and signs: Semiotics in the conceptual interface of evolutionary biology and memetics. *Semiotica*, 2019(230):327–340, 2019.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`, 2024.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. Are we done with MMLU? In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.262. URL `https://aclanthology.org/2025.naacl-long.262/`.

Glenn Grant, A Sandberg, and D McFadzean. Memetic lexicon, 1990.

Dadi Guo, Jiayu Liu, Zhiyuan Fan, Zhitao He, Haoran Li, Yumeng Wang, and Yi R Fung. Mathematical proof as a litmus test: Revealing failure modes of advanced large reasoning models. *arXiv preprint arXiv:2506.17114*, 2025a.

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

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

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. *NeurIPS*, 2021.

iFlytek. spark-X1, 2025. Accessed: 2025-01-15.

Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated errors in large language models. *arXiv preprint arXiv:2506.07962*, 2025.

Alex Kipnis, Konstantinos Voudouris, Luca M Schulze Buschoff, and Eric Schulz. metabench–a sparse benchmark of reasoning and knowledge in large language models. *arXiv preprint arXiv:2407.12844*, 2024.

Hynek Kydlíček. Math-Verify: Math Verification Library. URL `https://github.com/huggingface/math-verify`.

Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, et al. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.

OpenAI. Introducing gpt-5, 2025a. Accessed: 2025-08-07.

OpenAI. Introducing openai o3 and o4-mini, 2025b. Accessed: 2025-04-16.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

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. arxiv 2023. *arXiv preprint arXiv:2311.12022*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Association for Computational Linguistics (ACL)*, 2020.

Mara Schilling-Wilhelmi, Nawaf Alampara, and Kevin Maik Jablonka. Lifting the benchmark iceberg with item-response theory. In *AI for Accelerated Materials Design-ICLR 2025*.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *arXiv preprint arXiv:2310.16049*, 2023.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adri Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.

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/.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

xAI. Grok 3 beta — the age of reasoning agents, 2025a. Accessed: 2025-02-21.

xAI. Grok 4, 2025b. Accessed: 2025-07-09.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Extended Related Work

A.1 Result Driven IRT Analysis

Item Response Theory (IRT) (Baker & Kim) estimates a respondent's latent ability together with item difficulty and discrimination from observed responses. By migrating IRT from psychology and education, (Kipnis et al., 2024; Maia Polo et al., 2024) fit latent model ability alongside item difficulty and discrimination, then select high-information items and enable adaptive testing; as a result, they produce compact subsets that preserve full-benchmark scores while substantially reducing evaluation cost. Furthermore, (Schilling-Wilhelmi et al.) employs a Bayesian two-parameter IRT model that yields calibrated ability estimates with uncertainty, revealing how model rankings shift when viewed through the lens of IRT. However, these attempts do not adequately capture the heterogeneity across items or the behavioral similarities and differences across models.

Table 2: **Benchmark scores on phemotypes**

Model	Acc	PCS	Vig	Mas	Tra	Ing	Ast
gemini-2.5-pro(IR)	84.0	75.6	74.7	74.1	78.3	69.2	81.6
grok-4-0709(IR)	83.1	75.0	74.1	73.8	76.4	68.6	82.0
gpt-5-2025-08-07(IR)	82.8	74.8	74.5	74.2	75.7	67.8	82.1
o3-2025-04-16(IR)	82.2	73.9	73.6	73.4	75.0	66.6	81.2
grok-3(CoT)	76.7	67.5	67.6	66.3	68.2	59.9	75.4
gpt-4.1-2025-04-14(CoT)	75.8	64.6	64.8	62.8	67.4	55.5	72.6
kimi-k2-0711-preview(CoT)	74.1	64.9	65.1	63.4	64.9	57.7	73.3
gemini-2.5-flash(IR)	73.4	65.8	65.6	65.0	65.2	59.6	73.4
deepseek-R1(IR)	73.2	65.5	65.4	64.4	64.6	59.2	73.8
claude-sonnet-4-20250514(IR)	71.2	63.8	63.8	62.7	63.3	57.8	71.2
gemini-2.5-flash(CoT)	71.0	61.9	62.3	60.7	62.3	54.7	69.6
gpt-5-mini-2025-08-07(IR)	70.1	63.4	63.5	62.3	62.9	58.2	69.9
doubao-seed-1-6-250615(IR)	70.1	63.4	63.0	62.4	62.7	58.8	70.2
glm-4.5(IR)	69.3	60.4	59.4	59.2	59.8	55.1	68.6
deepseek-V3(CoT)	68.2	57.8	58.0	55.4	58.0	51.2	66.2
MiniMax-M1(IR)	68.1	60.9	60.6	59.9	60.6	55.9	67.2
claude-sonnet-4-20250514	68.0	59.4	60.0	57.5	59.8	53.5	66.4
claude-sonnet-4-20250514(CoT)	67.8	58.4	59.3	56.6	59.4	51.1	65.8
glm-4.5(CoT)	67.7	57.4	57.2	55.1	58.1	51.1	65.6
kimi-k2-0711-preview	67.7	57.8	55.8	56.8	56.2	54.1	66.1
doubao-seed-1-6-250615(CoT)	66.8	58.4	58.6	56.9	59.1	52.6	65.0
qwen3-235b-a22b(IR)	66.6	61.0	60.8	60.0	60.9	57.4	65.8
gpt-4o-2024-11-20(CoT)	66.5	53.2	52.4	50.4	55.7	45.9	61.6
gpt-4.1-mini-2025-04-14(CoT)	65.6	57.1	57.7	55.6	57.9	51.3	63.4
o3-mini-2025-01-31(IR)	65.3	58.3	58.4	56.9	58.6	53.9	63.7
gpt-5-nano-2025-08-07(IR)	64.5	58.5	57.8	57.2	58.7	55.3	63.2
claude-3-5-sonnet-20241022(CoT)	64.3	51.7	50.8	49.9	52.8	45.7	59.4
doubao-seed-1-6-250615	62.8	53.5	53.7	51.4	55.1	48.1	59.4
glm-4.5-air(CoT)	61.1	51.6	51.2	48.7	53.1	47.1	57.8
doubao-seed-1-6-flash-250715(IR)	61.0	55.0	54.4	53.8	55.9	52.3	58.3
qwen3-235b-a22b(CoT)	60.8	52.1	52.4	49.8	54.2	47.1	56.8
MiniMax-Text-01(CoT)	60.6	49.1	47.9	46.7	50.3	44.5	55.8
spark-X1(IR)	60.5	54.3	54.0	53.2	55.7	51.0	57.6
glm-4.5-air(IR)	58.9	50.5	49.5	48.2	50.7	47.2	57.0
doubao-seed-1-6-flash-250715(CoT)	56.7	48.7	48.3	46.3	50.9	45.1	52.6
qwen3-30b-a3b(CoT)	56.0	47.5	46.8	44.9	49.5	44.7	51.4
deepseek-V3	55.2	43.4	39.8	42.7	39.7	43.0	52.0
gemini-2.5-flash	55.0	44.4	41.3	43.3	42.0	43.5	52.1
qwen3-32b(CoT)	54.9	46.1	45.9	44.0	48.9	42.1	49.8
gpt-4.1-nano-2025-04-14(CoT)	53.0	43.9	43.9	40.6	46.4	39.8	48.6
gpt-4.1-2025-04-14	50.7	37.6	33.3	36.2	34.7	36.1	47.7
claude-3-5-sonnet-20241022	49.5	37.5	33.9	36.5	33.6	36.4	46.9
gpt-4o-2024-11-20	49.3	36.4	32.0	35.1	31.9	35.9	47.2
glm-4.5	49.0	36.2	31.3	34.8	31.0	37.7	46.1
grok-3	47.8	36.0	31.4	34.8	32.2	37.0	44.7
doubao-seed-1-6-flash-250715	46.9	39.0	36.7	36.3	38.9	39.3	43.9
qwen3-235b-a22b	44.8	34.7	30.7	33.2	32.4	36.1	40.9
gpt-4.1-mini-2025-04-14	44.2	34.4	30.8	32.7	30.4	35.7	42.2
MiniMax-Text-01	43.7	33.9	30.0	32.7	28.8	35.2	42.7
qwen3-30b-a3b	39.2	30.7	27.3	28.8	28.0	33.2	36.4
qwen3-32b	36.9	28.2	24.3	27.0	24.7	30.9	34.0
glm-4.5-air	36.1	26.6	22.0	24.8	21.3	29.9	34.9
gpt-4.1-nano-2025-04-14	29.1	24.4	21.7	22.1	20.1	27.1	31.1

A.2 Correlated Errors in LLMs

Prior studies have documented that large language models do not fail independently: incorrect model outputs are highly correlated across models, and often structured enough to reveal model families and shared failure modes. (Bradley, 2024) revealed that model errors are strongly correlated, manifested as high agreement on incorrect options in multiple-choice questions. He introduced a model classification approach based on error correlations, employing z-scores and hierarchical clustering to uncover model families. (Kim et al., 2025) conducted a study across hundreds of models and multiple benchmarks. They found that models sharing a developer, base architecture, and comparable size consistently exhibit higher agreement rates. While these studies indicate the presence of inherent similarities among different models, they do not provide a quantitative analysis of these attributes. Therefore, they fail to provide a mechanistic explanation of this similarity.

A.3 Memetics and Its Applications to LLMs

Dawkins first proposed the concept of *memes* (Dawkins, 1976), drawing an analogy to genes in cultural transmission. Building on this analogy, memetics introduced memotype and phemotype, paralleling genotype and phenotype (Grant et al., 1990; Blackmore, 2000; Álvarez, 2005; Fomin, 2019). In memetics, the *memotype* refers to the actual information content of a meme, while the *phemotype* denotes its concrete manifestation as produced by the memotype under specific conditions. Within LLM research, memetics has been applied to model ideological propagation (Farlow et al., 2024) and to explain reasoning behaviors (Birchall). These works fail to show how different memes shape similarities and differences across model populations.

B Probe Properties

B.1 Additional Discussion on MPPs

B.1.1 Risk

The weighting factor I_j can be viewed as an information weight derived from the overall error rate of model M_j. Its form resembles the notion of self-information $-\ln p$, assigning higher values when errors are rarer and lower values when errors are common. In this weighting scheme, models with extremely high error rates yield values close to zero, thereby diminishing their impact on the risk estimate. This prevents low-quality models, which fail almost universally, from artificially inflating co-failure statistics. Conversely, models that are generally accurate but occasionally fail on specific probes receive larger weights, highlighting their role in identifying probes that induce genuinely high-risk failure modes. Thus, weighting by I_j not only incorporates an information-theoretic perspective on model behavior but also mitigates distortions caused by extreme outlier models.

B.1.2 Surprise

Normalization To ensure comparability across models with different overall ability levels, model accuracy a_j is normalized by z-score:

$$a_j^z = \frac{a_j - \mu}{\sigma},$$

where μ and σ are the mean and standard deviation of $\{a_j\}_{j=1}^m$. This normalization guarantees that the contribution of each model is measured relative to the population, and it is consistently applied in all computations of Surprise.

Calculation of the Hard-side Surprise For the hard-side case, let

$$R_i = \{j \mid P_{ij} = 1\},$$

that is, R_i is the set of model indices such that M_j succeeds on probe i. The hard-side surprise of probe i is then defined as

$$s_i^{\text{hard}} \;=\; \big(-\ln(1-d_i)\big) \cdot \frac{1}{|R_i|} \sum_{j \in R_i} (1 - a_j^z),$$

where d_i is the difficulty of probe i. This formulation mirrors the easy-side case: while s_i^{easy} emphasizes probes that are generally easy yet unexpectedly failed by stronger models, s_i^{hard} emphasizes probes that are generally difficult yet unexpectedly solved by weaker models.

B.1.3 UNIQUENESS

For each probe i, the perception span $(P_{i1}, \ldots, P_{im})$ is a binary row vector recording the responses of m models. For the purpose of information-theoretic analysis, probe i is also viewed as a binary random variable P_i over the model population, where $P_i = 1$ indicates a correct response and $P_i = 0$ an incorrect one, and the vector entries $(P_{i1}, \ldots, P_{im})$ are regarded as empirical samples of this variable.

The uniqueness of probe i is defined as

$$u_i = \frac{1}{n-1} \sum_{k \neq i}^{n} H(P_k \mid P_i),$$

where P_k denotes the random variable associated with probe k.

The conditional entropy term is expanded as

$$H(P_k \mid P_i) = \Pr(P_i = 1)\, H(P_k \mid P_i = 1) + \Pr(P_i = 0)\, H(P_k \mid P_i = 0),$$

with

$$H(P_k \mid P_i = x) = -p_x \log_2 p_x - (1 - p_x) \log_2 (1 - p_x), \quad p_x = \Pr(P_k = 1 \mid P_i = x),$$

where probabilities are estimated empirically from the model population.

Thus, u_i measures the average conditional entropy of other probes given probe i, capturing how much information the responses to probe i contribute about the rest of the probes.

B.1.4 ϕ-COEFFICIENT

For each probe i, $P_i = (P_{i1}, P_{i2}, \ldots, P_{im})$, where $P_{ij} \in \{0, 1\}$ indicates whether probe i is answered incorrectly by model M_j. Each P_{ij} can be interpreted as the observed samples of a Bernoulli random variable, whose expectation corresponds to the empirical difficulty d_i of probe i.

The similarity between probes i and k is computed using the ϕ-coefficient, defined as

$$\phi(i, k) = \frac{n_{11} n_{00} - n_{10} n_{01}}{\sqrt{(n_{1\cdot} n_{0\cdot} n_{\cdot 1} n_{\cdot 0})}}, \tag{9}$$

where $n_{\alpha\beta}$ denotes the number of models for which $P_{im} = \alpha$ and $P_{km} = \beta$ with $\alpha, \beta \in \{0, 1\}$. Although the ϕ-coefficient is formally a correlation measure between two Bernoulli random variables, in this work, it is employed as a similarity score quantifying the extent to which two probes exhibit consistent difficulty patterns across the model population.

B.2 EXPANDED VISUALIZATIONS

This section presents the visualization results of various property combinations across different datasets.

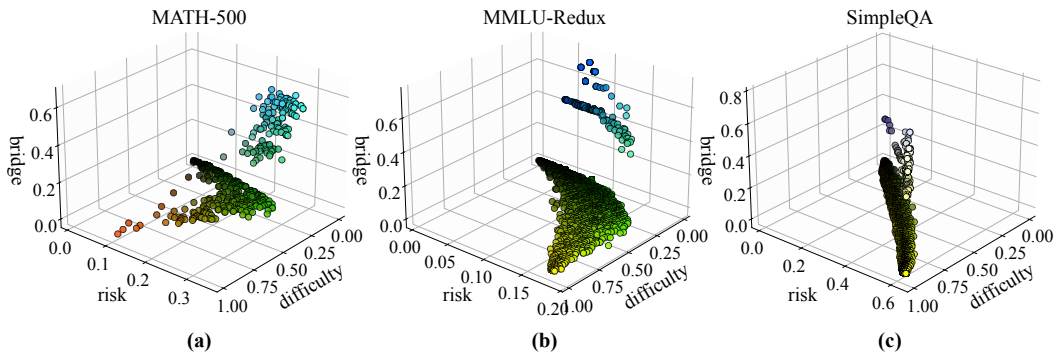

Figure 10: **Probe properties distributions across datasets.** Axes depict difficulty, risk, and bridge.

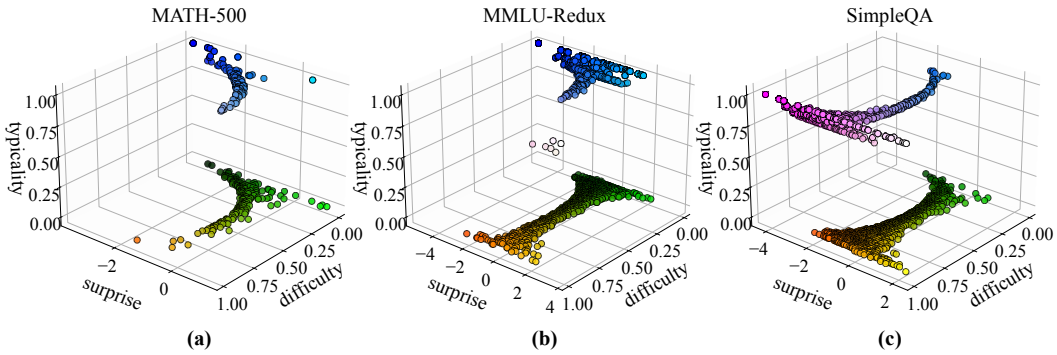

Figure 11: **Probe properties distributions across datasets.** Axes depict difficulty, surprise, and typicality. A typicality value of 0 indicates that the probe does not belong to any cluster during the clustering process.

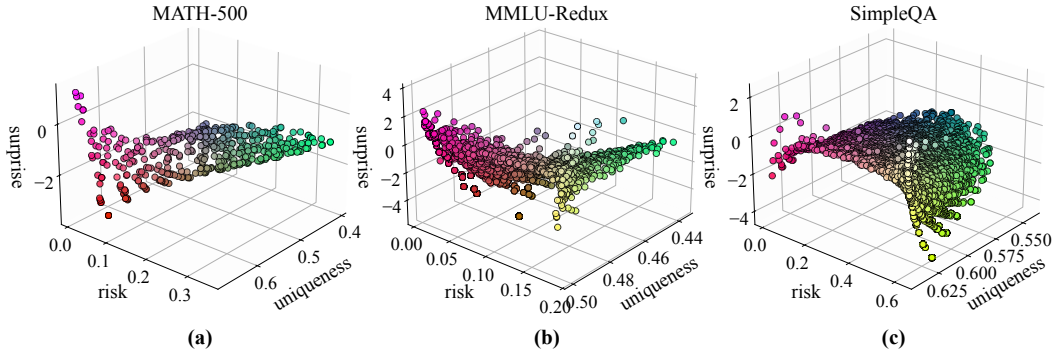

Figure 12: **Probe properties distributions across datasets.** Axes depict uniqueness, risk, and surprise.

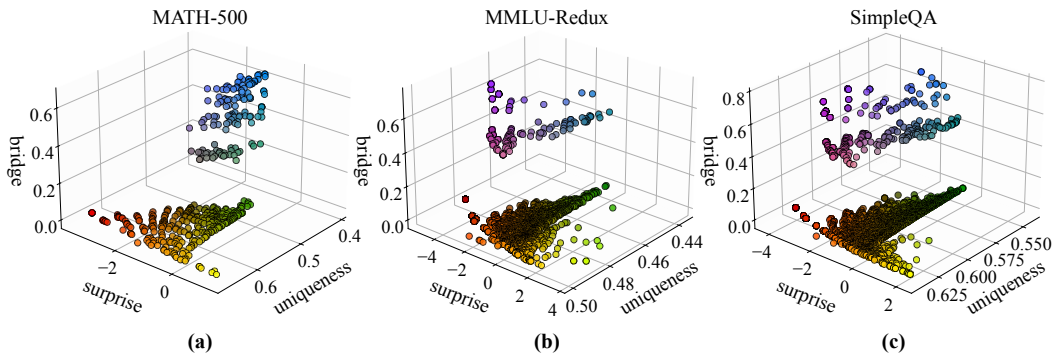

Figure 13: **Probe properties distributions across datasets.** Axes depict uniqueness, surprise, and bridge.

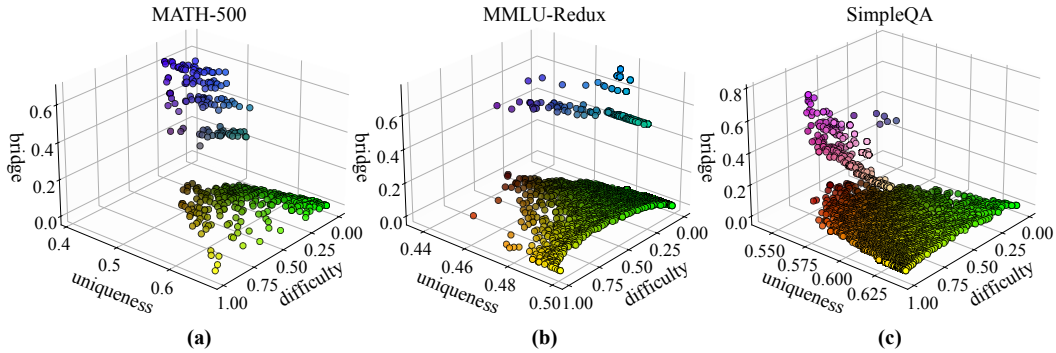

Figure 14: **Probe properties distributions across datasets.** Axes depict difficulty, uniqueness, and bridge.

C PHEMOTYPES

C.1 ADDITIONAL DISCUSSION ON PHEMOTYPES

Notation. $P \in \{0,1\}^{n\times m}$ is the perception matrix, with entries P_{ij} (probe i, model j), where n is the number of probes and m is the number of models. Probe raw attributes are denoted by $\rho_i \in \mathbb{R}$ (e.g., risk, surprise, difficulty, typicality, bridge, uniqueness). Weights are $w = (w_1, \ldots, w_n)^\top \in \mathbb{R}^n_{\geq 0}$.

C.1.1 NORMALIZATION

Given scores $\{\rho_i\}_{i=1}^n$, define the average-tie rank

$$\mathrm{rk}_i \;=\; 1 \;+\; \sum_{k=1}^{n} \mathbf{1}\{\rho_k < \rho_i\} \;+\; \tfrac{1}{2}\left(\sum_{k=1}^{n} \mathbf{1}\{\rho_k = \rho_i\} - 1\right). \tag{10}$$

Normalize ranks to unit interval

$$\mathrm{frac}_i \;=\; \frac{\mathrm{rk}_i - \frac{1}{2}}{n} \;\in\; \left(\tfrac{1}{2n},\, 1 - \tfrac{1}{2n}\right). \tag{11}$$

Here, n denotes the number of probes.

With temperature $\tau > 0$ and output range $[\ell, h]$, define

$$\tilde{\rho}_i \;=\; \mathrm{Norm}(\rho_i) \;=\; \sigma\!\left(\frac{\mathrm{frac}_i - \frac{1}{2}}{\tau}\right), \qquad \hat{\rho}_i \;=\; \ell + (h - \ell)\,\tilde{\rho}_i, \tag{12}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$.

Shorthand:

$$\begin{aligned} \tilde{r}_i &= \mathrm{Norm}(\rho_i^{\mathrm{risk}}), & \tilde{s}_i &= \mathrm{Norm}(\rho_i^{\mathrm{surprise}}), \\ \tilde{d}_i &= \mathrm{Norm}(\rho_i^{\mathrm{difficulty}}), & \tilde{t}_i &= \mathrm{Norm}(\rho_i^{\mathrm{typicality}}), \\ \tilde{b}_i &= \mathrm{Norm}(\rho_i^{\mathrm{bridge}}), & \tilde{u}_i &= \mathrm{Norm}(\rho_i^{\mathrm{uniqueness}}). \end{aligned} \tag{13}$$

C.1.2 Score

For weights $w \geq 0$ with $\sum_{i=1}^{n} w_i = 1$:

$$\mathrm{Score}(w;\, P_{\cdot j}) \;=\; \sum_{i=1}^{n} w_i\, P_{ij}, \qquad \sum_{i=1}^{n} w_i = 1. \tag{14}$$

Range: $\mathrm{Score} \in [0, 1]$.

C.1.3 Cluster shrink

Let clusters $\{C_c\}_{c=1}^{K}$ partition probes; c_i is probe i's cluster, size $|C_{c_i}|$.

$$s_i^{\mathrm{shr}} \;=\; \big(|C_{c_i}|\big)^{-\beta}, \qquad \beta \in [0, 1], \tag{15}$$

optionally clipped:

$$s_i^{\mathrm{shr}} \;\leftarrow\; \min\!\big\{\max\{s_i^{\mathrm{shr}},\, \ell\},\, h\big\}, \quad \ell \leq h. \tag{16}$$

Defaults: $\beta = 0.5$, $[\ell, h] = [0, 1]$.

C.1.4 Phenotype weights and scores

$$w_i^{\mathrm{Vig}} = \tilde{r}_i\, \tilde{s}_i, \quad w_i^{\mathrm{Mas}} = \tilde{t}_i\, \tilde{d}_i\, s_i^{\mathrm{shr}}, \quad w_i^{\mathrm{Trf}} = \tilde{b}_i\, \tilde{d}_i, \quad w_i^{\mathrm{Ing}} = \tilde{u}_i\, \tilde{d}_i, \quad w_i^{\mathrm{Ast}} = \tilde{s}_i. \tag{17}$$

Phenotype score for model M_j: $\mathrm{Score}(w^{\cdot};\, P_{\cdot j})$.

C.2 More Results

Figure 15 shows the results of phemotypes compared with the accuracy for each dataset separately. The models are sorted from high to low according to their accuracy. It can be seen that the scores of the five phemotypes do not change synchronously with the accuracy.

Table 2 presents the phemotype benchmarks of all models across three datasets (MMLU-Redux, MATH-500, and SimpleQA), with models sorted in descending order of accuracy. All scores are averaged over the three datasets. PCS refers to the Phemotype Composite Score, which represents the average of the five phemotypes, with scores scaled to 0–100 with one decimal. Column abbreviations are as follows: Acc for Accuracy, PCS for the composite, Vig for Vigilance, Mas for Mastery, Tra for Transfer, Ing for Ingenuity, and Ast for Astuteness. The table enables side-by-side inspection of aggregate accuracy and phemotype dimensions, making it possible to identify cases where accuracy-similar models exhibit divergent phemotype profiles.

C.2.1 Phemotype Composition Across Reasoning Modes

As shown in Figure 16, the three reasoning modes, produce noticeably different phemotype compositions. Across reasoning modes, the total capability rises from Base, through CoT, to IR. The pies show composition, not magnitude: how each mode distributes its "effort" across five phemotypes. These different workflows naturally reweight the phemotype mix even when headline accuracy moves in the same direction.

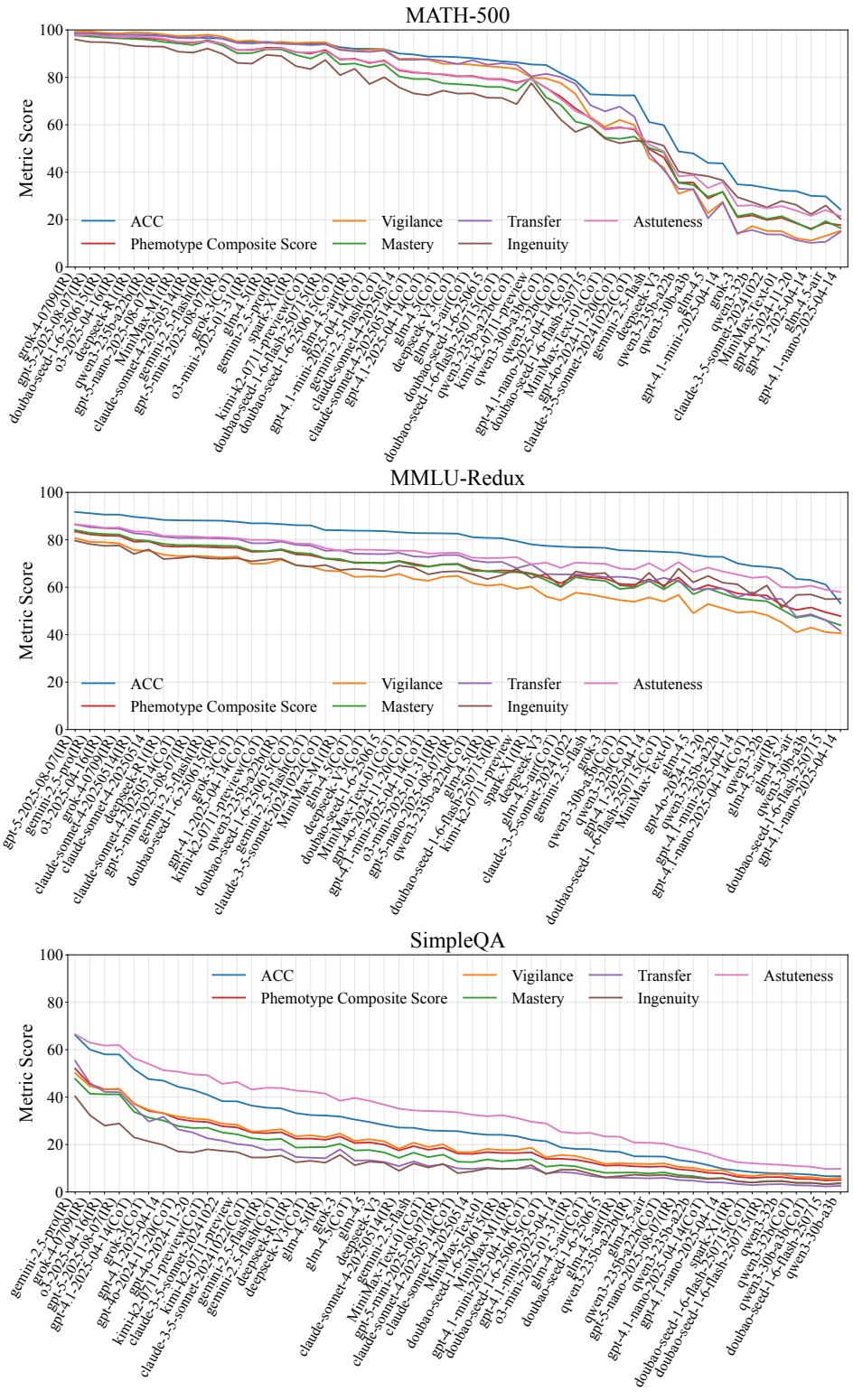

Figure 15: **The scores between accuracy and phemotypes.** This figure shows a line plot of the phemotypes and accuracy rates for all models under different reasoning modes (sorted in descending order of accuracy for each model).

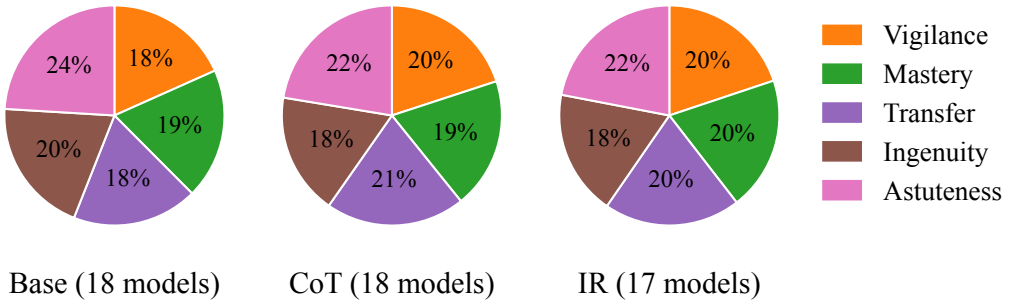

Figure 16: **Phemotype composition across reasoning modes.** Each pie chart shows the relative distribution of the five phemotype dimensions: Vigilance, Mastery, Transfer, Ingenuity, and Astuteness.

D More Details of Open LLM Leaderboard Application

D.1 Details About Models and Datasets

Results from the Open LLM Leaderboard v2 (Fourrier et al., 2024) on Hugging Face were used to assess the validity of the probing-memes paradigm. This work collects the publicly available response data of models from the Open LLM Leaderboard on Hugging Face, and removes models with missing records as well as items with incomplete information, ensuring that all models have complete results on the same set of items. Six datasets were contained in the leaderboard: IFEval (Zhou et al., 2023), MATH (Hendrycks et al., 2021), MUSR (Sprague et al., 2023), MMLU-Pro (Wang et al., 2024), BBH (Suzgun et al., 2023) and GPQA (Rein et al.). The three primary GPQA subsets (Diamond, Main and Extended) were available for all models. Therefore, GPQA-Diamond was used as a substitute.

D.2 More Results from Open LLM Leaderboard

Table 3: **Benchmark scores on phemotype**

Model	Acc	PCS	Vig	Mas	Tra	Ing	Ast
calme-3.2-instruct-78b	60.3	47.7	44.8	46.9	47.1	43.4	56.2
CalmeRys-78B-Orpo-v0.1	60.0	47.3	44.5	46.6	46.8	43.0	55.8
calme-3.1-instruct-78b	59.6	46.7	43.7	46.0	46.1	42.5	55.4
calme-2.4-rys-78b	59.5	46.5	43.5	45.8	45.9	42.0	55.3
FluentlyLM-Prinum	58.2	42.5	39.2	42.3	42.4	35.9	53.0
Homer-v1.0-Qwen2.5-72B	57.9	42.5	38.4	42.1	42.2	37.4	52.2
ultiima-72B	57.1	41.4	37.2	41.1	41.2	36.3	51.1
Gilgamesh-72B	57.0	42.9	39.3	42.8	42.9	37.7	51.9
shuttle-3	56.6	40.4	37.0	40.1	40.2	33.0	51.5
T3Q-qwen2.5-14b-v1.0-e3	56.2	41.9	38.0	42.5	42.6	36.9	49.7
T3Q-Qwen2.5-14B-Instruct-1M-e3	56.2	41.9	38.0	42.5	42.6	36.9	49.7
test-2.5-72B	56.1	41.4	37.9	41.6	41.7	35.5	50.3
Qwen2.5-72B-Instruct-abliterated	55.4	40.1	36.2	39.6	39.7	35.1	49.8
sky-t1-coder-32b-flash	55.3	41.0	38.4	40.1	40.2	34.9	51.6
RYS-XLarge	55.3	39.8	36.5	39.5	39.6	33.4	49.9
calme-2.1-rys-78b	55.2	40.3	37.3	39.8	39.9	34.4	50.3
tempmotacilla-cinerea-0308	55.2	38.4	34.3	38.8	39.0	32.1	48.0
ultiima-72B-v1.5	54.9	39.1	34.7	38.7	38.8	34.2	49.2
Rombos-LLM-V2.5-Qwen-72b	54.9	39.9	36.0	39.4	39.5	35.2	49.2
li-14b-v0.4	54.7	38.5	35.4	38.1	38.3	30.9	49.7

Table 3 presents the phemotype benchmarks of models the from Open LLM Leaderboard. All scores are averaged over the six datasets. PCS refers to the Phemotype Composite Score, which represents the average of the five phemotypes, with scores scaled to 0–100 with one decimal. Column abbreviations are as follows: Acc for Accuracy, PCS for the composite, Vig for Vigilance, Mas for Mastery, Tra for Transfer, Ing for Ingenuity, and Ast for Astuteness.

D.3 Extended Visualizations

Figure 17 shows the heatmaps of the perception span similarity of probes in the Open LLM Leaderboard dataset.

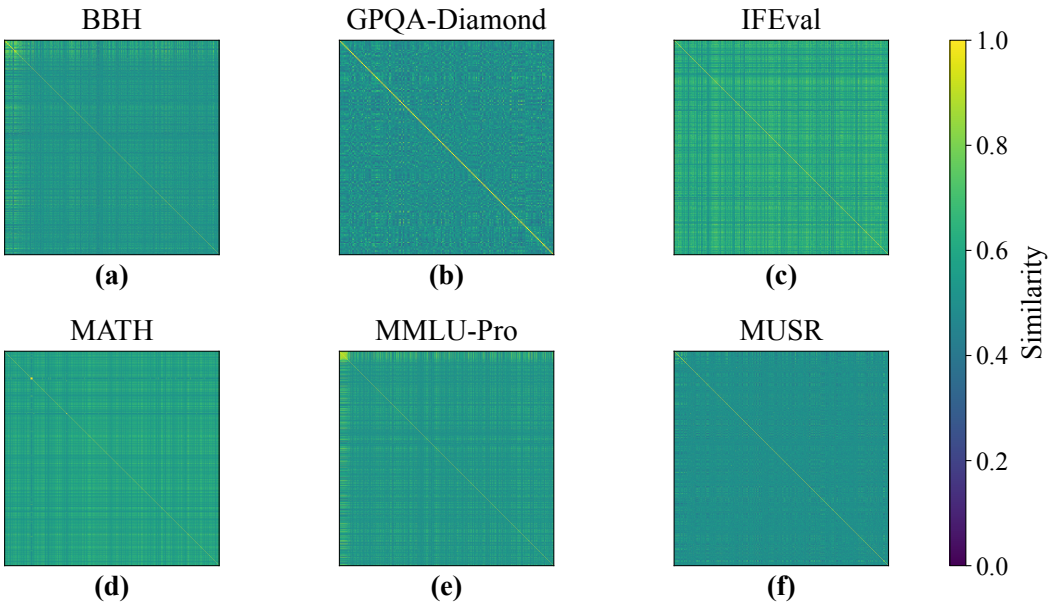

Figure 17: **Probe similarity heatmaps across datasets.** Probes are reordered by cluster to reveal blocks; similarities are computed from each probe's perception span.

E Models and Datasets

E.1 Models

The following 28 models are included in this work. OpenAI (9 models): gpt-4.1-2025-04-14, gpt-4.1-mini-2025-04-14, gpt-4.1-nano-2025-04-14, gpt-4o-2024-11-20, o3-2025-04-16, o3-mini-2025-01-31, gpt-5-2025-08-07, gpt-5-mini-2025-08-07, gpt-5-nano-2025-08-07 (Achiam et al., 2023; OpenAI, 2025b;a); Anthropic (2 models): claude-3-5-sonnet-20241022, claude-sonnet-4-20250514 (Anthropic, 2024; 2025); Google (2 models): gemini-2.5-flash, gemini-2.5-pro (Comanici et al., 2025); DeepSeek (2 models): deepseek-V3, deepseek-R1 (Liu et al., 2024; Guo et al., 2025b); Alibaba (3 models): qwen3-235b-a22b, qwen3-30b-a3b, qwen3-32b (Yang et al., 2025); xAI (2 models): grok-3, grok-4-0709(xAI, 2025a;b); MiniMax (2 models): MiniMax-Text-01, MiniMax-M1 (Li et al., 2025; Chen et al., 2025); ByteDance (2 models): doubao-seed-1-6-250615, doubao-seed-1-6-flash-250715(ByteDance, 2025a;b); Zhipu AI (2 models): glm-4.5, glm-4.5-air (Zeng et al., 2025); Moonshot AI (1 model): kimi-k2-0711-preview (Bai et al., 2025); iFlytek (1 model): spark-X1 (iFlytek, 2025).

E.2 DATASETS

There are three datasets involved in the experiment part of this work, including MATH-500 (Lightman et al., 2023), MMLU-Redux (Gema et al., 2025), and SimpleQA (Wei et al., 2024).

MATH-500 is a 500-problem subset of MATH (Hendrycks et al., 2021) curated by the OpenAI team. Its items come from high-school mathematics competitions, and many are challenging for humans. The high difficulty increases the discriminative power, which is crucial for revealing the distinct phemotypes of different models in the research. Another point is that all the problems in MATH-500 can be solved with step-by-step reasoning. Therefore, it is an excellent dataset for evaluating a model's stepwise reasoning ability.

MMLU-Redux is a revised version of MMLU (Hendrycks et al., 2020) dataset. It comprises 5,399 choice questions spanning 57 subject areas, including fields such as mathematics, physics, chemistry, political science, economics, law, and philosophy. Its broad subject coverage enables a comprehensive assessment of general knowledge across disciplines. In addition, MMLU-Redux is composed entirely of multiple-choice questions. This format allows for straightforward and highly accurate evaluation of model answers.

SimpleQA is a challenging dataset comprising 4,326 commonsense question-answer pairs. A distinctive feature of this dataset is its ternary answer evaluation scheme ("Correct", "Incorrect", or "Not Attempted"); for consistency, this study treated "Not Attempted" answers as "Incorrect". Due to its high difficulty, few models performed well in our experiments. The dataset's broad topical coverage enables a comprehensive evaluation of model capabilities across diverse domains. Moreover, all questions are phrased precisely and unambiguously. Their high reliability is ensured through a rigorous validation process involving multiple annotators who independently provided and cross-verified answers. This procedure guarantees a unique gold answer for each question.

F DETAILED EXPERIMENTAL SETTINGS

F.1 REASONING MODES

Two prompting templates are used in this paper: a default prompt and a chain of thought prompt. Models with internal reasoning (so-called reasoning models, like deepseek-R1 (Guo et al., 2025b)) use the default template, with the internal reasoning executed by the model; several of these models allow internal reasoning to be disabled, which permits use of both templates. Models without internal reasoning use both templates.

F.2 PROBE FILTERING

As shown in Table 4, most dataset contains a number of probes that are either answered correctly or incorrectly by all models.

Table 4: **Summary of unanimous probes and counts after filtering.**

Dataset	Total Probes	Unanimous (Correct)	Unanimous (Incorrect)	Remaining Probes
MMLU-Redux	5,399	2,698	35	2,666
Math-500	500	47	0	453
SimpleQA	4,326	10	569	3,747
BBH	5,759	0	0	5,759
GPQA-Diamond	198	0	0	198
IFEval	536	0	5	536
MATH	1,297	0	27	1,297
MUSR	756	0	0	756
MMLU-Pro	12,032	0	0	12,032

F.3 HYPERPARAMETERS

For models without internal reasoning, temperature is set to 0, top-p to 1, and max tokens to 8192. Internal reasoning models often output a large amount of reasoning content. For models with internal reasoning, max tokens is set to 28672, and the remaining parameters follow the providers' defaults because these models often do not support very low temperature, settings and defaults are recommended. Internal reasoning models from the Qwen family max tokens does not include the number of reasoning tokens, so set max tokens to 8192 and thinking budget to 20480 (for limiting max reasoning output).

F.4 PROMPT TEMPLATES

The boxes below present the prompts used for each dataset and reasoning mode, where "<question text>" denotes the text of the question. These prompts follow certain conventions. For instance, all prompts specify the required answer format, which varies across datasets. Furthermore, when testing under the Chain-of-Thought (CoT) setting, the phrase "Please reason step by step" is included to enable CoT reasoning, and models are instructed to output their reasoning process separately.

In terms of formatting, models are required to provide their final answers in the form "Answer:" followed by the answer, with no further explanation permitted. In addition, the MATH-500 prompt requires answers to be enclosed in \boxed{} to facilitate the extraction of mathematical expressions; the MMLU-Redux prompt requires the answer to be a single letter corresponding to the selected option; and the SimpleQA prompt imposes no additional requirements beyond the "Answer:" format.

MATH-500 *CoT Prompting (CoT).*

```
Answer the following question.
Question: ``<question text>''
Please reason step by step.
Your response must strictly follow the format below:
Reasoning Process: {Explain your reasoning step by step}
Answer: \boxed{Your final result without any explanation}
```

MATH-500 *Default Prompting (Base) and Internal Reasoning (IR).*

```
Answer the following question.
Question: ``<question text>''
Your response must strictly follow the format below:
Answer: \boxed{Your final result without any explanation}
```

MMLU-Redux *CoT Prompting (CoT).*

```
Answer the following question.
Question: ``<question text>''
Please reason step by step.
Your response must strictly follow the format below:
Reasoning Process: {Explain your reasoning step by step}
Answer: {Your final choice letter without any explanation}
```

MMLU-Redux *Default Prompting (Base) and Internal Reasoning (IR).*

```
Answer the following question.
Question: ``<question text>''
Your response must strictly follow the format below:
Answer: {Your final choice letter without any explanation}
```

SimpleQA *CoT Prompting (CoT).*

```
Answer the following question.
Question: ``<question text>''
Please reason step by step.
Your response must strictly follow the format below:
Reasoning Process: {Explain your reasoning step by step}
Answer: {Your final answer without any explanation}
```

SimpleQA *Default Prompting (Base) and Internal Reasoning (IR).*

```
Answer the following question.
Question: ``<question text>''
Your response must strictly follow the format below:
Answer: {Your final answer without any explanation}
```

F.5 Details and Methods of Answer Verification

Several issues arose during the extraction and evaluation of model responses. These included non-compliant output formats (despite explicit instructions), responses exceeding token limits, and model refusals to answer sensitive questions. Tables 7, 5, and 6 present the statistics for these respective issues. The token limits are specified in Appendix F.3.

The experiment applied two rounds of verification. The first round enforced the prompt's formatting requirements strictly: any response that failed to comply with the required format was treated as incorrect. The second round attempted to match and extract answers using a variety of possible formats, which did not conform to the prompt. Therefore, a purely formatting error was always regarded as incorrect in the first round verification, but could be viewed as correct in the second round verification if the model's output contained the correct answer. Responses that exceeded the token number limits and responses in which the model refused to answer were treated as incorrect in both verification rounds. The data presented in the experiments and analyses were obtained from the second round of verification.

Each round of answer verification comprises two steps: answer extraction and answer evaluation. The answer extractor extracts the model's answer (without any explanation) from the model's response, while the answer evaluator compares the extracted answer with the golden answer. Different datasets used different methods to extract and evaluate answers, and the methods are presented below.

MATH-500. MATH-500 dataset uses Math-Verify (Kydlíček) library to extract and evaluate answers. The extractor first attempts to extract the content enclosed by $\backslash$boxed$\{\}$ from the model response using regular expressions. If the attempt fails, the response will be sent directly to Math-Verify. Math-Verify is capable of extracting answers in LaTeX format as well as numeric/expression formats from the model response. It uses the following formats to extract answers in descending priority:

- Explicit final answer (e.g., "Final answer is 3. I hope");
- General final answer (e.g., "final answer is 3") and boxed expressions (e.g., $\backslash$boxed$\{3\}$) at the same priority;
- Answer with a colon (e.g., "answer: 3");
- Answer without a colon (e.g., "answer is 3");
- Unanchored matches (e.g., "3").

Unanchored matches carry some risk of extracting numbers/expressions that appear in the response but are not the model's perceived answer; however, manual per-item inspection found no such errors. After extraction, Math-Verify normalizes the answer format and then parses it with SymPy. The golden answer is likewise converted to SymPy, and Math-Verify judges correctness by comparing the two SymPy expressions.

MMLU-Redux. Regular expressions are used to extract the one-letter answer in MMLU-Redux. There are three modes in the answer extraction as follows:

Table 5: **Numbers of refusals to answer**

Model	MMLU-Redux	SimpleQA
qwen3-235b-a22b(IR)	6	34
spark-x1(IR)	12	7
MiniMax-M1(IR)	10	8
qwen3-30b-a3b(CoT)	5	8
qwen3-235b-a22b(CoT)	5	7
glm-4.5(CoT)	7	5
qwen3-30b-a3b	5	5
qwen3-235b-a22b	4	6
qwen3-32b(CoT)	5	5
qwen3-32b	2	5
glm-4.5-air(CoT)	3	3
glm-4.5(IR)	5	0
glm-4.5-air(IR)	3	1
glm-4.5	1	0
glm-4.5-air	1	0
others	0	0

Table 6: **Numbers of responses exceeding token limit**

Model	MATH-500	MMLU-Redux	SimpleQA
glm-4.5-air(IR)	31	506	452
glm-4.5(IR)	15	209	447
gemini-2.5-flash(CoT)	23	13	12
doubao-seed-1-6-flash-250715(CoT)	5	2	31
glm-4.5-air(CoT)	10	15	12
glm-4.5(CoT)	8	4	10
MiniMax-M1(IR)	0	2	18
gpt-4.1-mini-2025-04-14(CoT)	0	0	14
gpt-4.1-2025-04-14(CoT)	6	2	2
kimi-k2-0711-preview(CoT)	7	1	2
kimi-k2-0711-preview	6	0	0
grok-4-0709(IR)	2	1	3
doubao-seed-1-6-250615(CoT)	2	2	2
doubao-seed-1-6-flash-250715(IR)	2	0	3
gpt-4.1-nano-2025-04-14(CoT)	0	0	5
gemini-2.5-flash	2	0	2
doubao-seed-1-6-flash-250715	1	0	2
qwen3-30b-a3b(CoT)	0	0	3
deepseek-reasoner(IR)	2	0	0
gpt-4.1-nano-2025-04-14	0	0	2
deepseek-chat(CoT)	1	0	0
o3-2025-04-16(IR)	1	0	0
gpt-5-2025-08-07(IR)	1	0	0
claude-sonnet-4-20250514(CoT)	1	0	0
doubao-seed-1-6-250615	0	1	0
qwen3-30b-a3b	0	0	1
doubao-seed-1-6-250615(IR)	0	0	1
others	0	0	0

Table 7: **Numbers of malformed answers**

Model	MATH-500	MMLU-Redux	SimpleQA
doubao-seed-1-6-flash-250715(IR)	100	956	838
doubao-seed-1-6-flash-250715	19	707	912
MiniMax-M1(IR)	497	1119	21
doubao-seed-1-6-250615	5	21	693
spark-x1(IR)	88	399	16
gemini-2.5-flash(CoT)	219	220	0
gemini-2.5-flash(IR)	249	122	4
doubao-seed-1-6-flash-250715(CoT)	39	26	292
grok-3(CoT)	282	24	9
deepseek-reasoner(IR)	299	4	0
glm-4.5-air(IR)	168	84	22
grok-4-0709(IR)	234	1	0
qwen3-235b-a22b(IR)	191	6	2
glm-4.5(IR)	163	3	3
gemini-2.5-flash	152	3	0
qwen3-235b-a22b	18	97	0
doubao-seed-1-6-250615(CoT)	3	12	100
doubao-seed-1-6-250615(IR)	2	12	101
gpt-4o-2024-11-20(CoT)	12	77	2
gemini-2.5-pro(IR)	61	9	0
qwen3-235b-a22b(CoT)	14	42	0
kimi-k2-0711-preview(CoT)	46	4	4
gpt-4.1-2025-04-14(CoT)	0	42	0
MiniMax-Text-01(CoT)	4	24	8
gpt-4.1-nano-2025-04-14(CoT)	5	31	0
qwen3-32b(CoT)	15	20	0
o3-mini-2025-01-31(IR)	1	11	21
qwen3-30b-a3b(CoT)	9	22	0
grok-3	2	24	2
qwen3-30b-a3b	19	2	0
deepseek-chat	12	7	0
kimi-k2-0711-preview	18	1	0
glm-4.5(CoT)	10	8	0
MiniMax-Text-01	0	14	4
gpt-5-mini-2025-08-07(IR)	0	11	5
gpt-4.1-nano-2025-04-14	9	5	0
qwen3-32b	10	2	0
gpt-4.1-mini-2025-04-14(CoT)	4	7	0
deepseek-chat(CoT)	4	6	0
claude-3-5-sonnet-20241022(CoT)	1	8	0
glm-4.5-air(CoT)	1	7	0
gpt-5-2025-08-07(IR)	0	5	1
gpt-4.1-2025-04-14	0	6	0
claude-sonnet-4-20250514(IR)	0	1	5
o3-2025-04-16(IR)	0	5	0
claude-sonnet-4-20250514(CoT)	0	4	0
gpt-4.1-mini-2025-04-14	1	3	0
gpt-5-nano-2025-08-07(IR)	0	2	1
glm-4.5	1	1	0
glm-4.5-air	0	2	0
gpt-4o-2024-11-20	0	0	1
others	0	0	0

- Searching answer using "Answer"/"answer" anchor. If multiple matches occur, then take the last match.
- Searching answer with other anchors, like "{}" and "**". These anchors do not mean the letter beside them is definitely the answer. Therefore, the extractor accepts a match as an answer only if exactly one match is found.
- Full-string match. Sometimes models give one-letter responses, with no anchors existing in these responses. However, it is risky to extract non-anchor answers in responses. To address this issue, the extractor applies full-string matches, matching responses like "A", "A." and so forth.

After the extraction, the evaluation step only requires a simple string comparison between the extracted answer and the golden answer.

SimpleQA. In SimpleQA, the extractor only extracts the content following "Answer:" as the answer, without any other format requirements. GPT-4.1 is employed as an LLM evaluator to evaluate the answers of the models under test by comparing their answers with the golden answer. The prompt for the LLM evaluator is the same as that in SimpleQA's official publication paper (Wei et al., 2024). Under this prompt, the LLM evaluator classifies the answer into three categories: Correct, Incorrect, and Not Attempted. A response will be classified into "Not Attempted" if the model recognizes its inability to solve the problem and refrains from providing an answer. As long as it gives an answer, it will be classified into "Correct" or "Incorrect". In this work, only answers classified into the "Correct" category were regarded as correct answers, and other answers were all deemed incorrect.

G USE OF LARGE LANGUAGE MODELS

This article was written with the moderate use of LLMs as polishing tools.