Enabling Scalable, Deterministic, and Reliable Evaluation of AI Tutors

Anonymous ACL submission

Abstract

The rise of AI tutors has created a need for reliable methods to evaluate the quality of educational feedback, yet current approaches rely heavily on costly human annotation or non-deterministic LLM-as-a-Judge systems with opaque reasoning processes. In this work, we revisit evaluation through the lens of LLM representations, proposing a probing-based method for assessing pedagogical quality. Specifically, we use a range of frozen language models as fixed encoders, extracting final-layer hidden-state representations from a single forward pass over an AI tutor's response, and train lightweight linear models to predict pedagogical quality signals from these embeddings. Across multiple baselines and four controlled studies, our approach consistently outperforms LLM-as-a-Judge methods while being significantly more compute-efficient. We further analyze the limitations of traditional LLM-as-a-Judge systems, identifying systematic failure modes including reasoning–label inconsistencies, where models produce a final label that contradicts its own reasoning, and role hallucination effects, where models continue adopting conversational roles rather than producing the desired evaluations. Taken together, our results suggest that linear probing provides a simple, interpretable, and scalable alternative for assessing educational AI systems.

1 Introduction

Intelligent Tutoring Systems (ITS) powered by large language models (LLMs) (i.e., AI Tutors) hold significant potential to scale high-quality education (Wang et al., 2024b). However, their deployment is constrained by the difficulty of evaluation (Macina et al., 2025). Assessing pedagogical quality of an AI tutor's output requires substantial domain expertise, making expert annotation expensive and difficult to scale.

To address this bottleneck, recent work has increasingly adopted LLM-as-a-Judge frameworks, in which language models are prompted to produce evaluation of an AI tutor's response (Naeem et al., 2026). Within our study, we refer to these prompt-based, token generation approach as "generative judges" to separate them from other usages of LLM as an evaluation technique. However, despite their convenience, this paradigm suffers from a fundamental mismatch between the model's training objective (i.e., auto-regressive token generation) and its downstream use as a structured evaluator in specialized domains, particularly education (Hashemi et al., 2024). As a result, their outputs are often non-deterministic, sensitive to prompt phrasing, and prone to hallucination, limiting their reliability in real-world settings (Chiang and Lee, 2023).

Nonetheless, given the strong representational capacity of modern LLMs (Li et al., 2024) and evidence that generative judges can successfully assess instructional quality (Fan et al., 2025; An et al., 2025), we hypothesize that the information required for reliable evaluation is already encoded in their hidden states and can be extracted without token generation.

To test this hypothesis, we propose a representation-probing framework to measure how well an LLM encodes signals relevant to instructional effectiveness and how easily those signals can be read out using a simple probe. That is, instead of generating free-form evaluations, we pass structured tutoring interactions through a frozen LLM, extract the final-layer hidden states, and train lightweight linear probes to predict pedagogical quality labels directly.

As such, our contributions are as follows:

- We demonstrate that LLMs meaningfully encode pedagogical quality, and this signal can be decoded from their hidden state with a

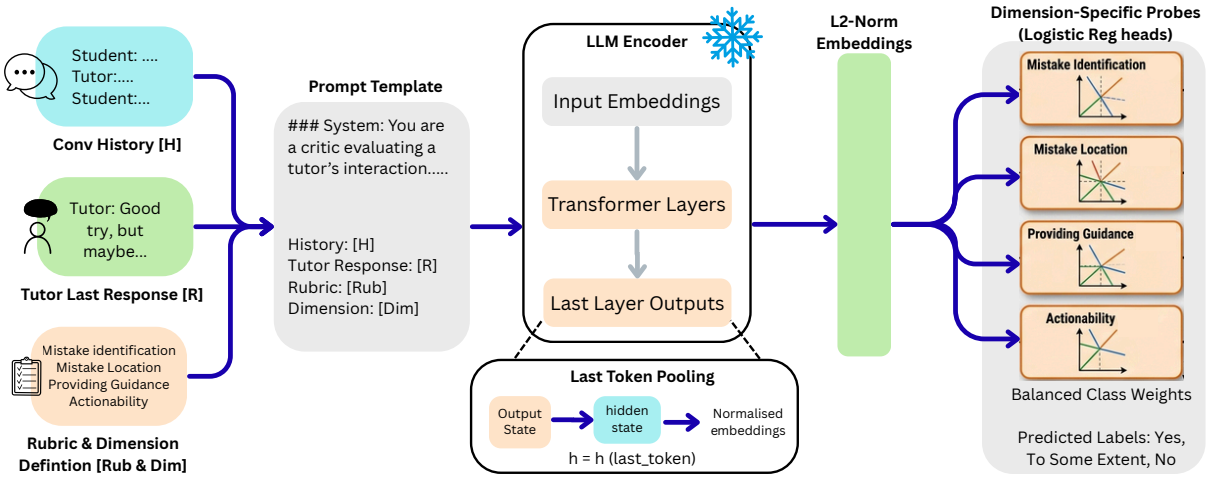

Figure 1: **Overview of the representation-based evaluation pipeline**. Rather than generating text, a frozen LLM encodes each formatted evaluation instance and its final-token hidden state is L2-normalized and extracted. A lightweight linear classifier, trained separately per pedagogical dimension, maps this representation to expert-annotated quality labels (1 = Yes, 2 = To Some Extent, 3 = No)

.

simple linear probe. Our approach consistently outperforms generative judges across model families and scales and is robust across four different control tests.

- We identify several recurring failure patterns in generative judging, such as reasoning-label mismatch, overly verbose output that fail to parse, and role hallucination. These failures highlight a fundamental tension between how auto-regressive models generate text and what structured rubric evaluation requires of them.

- Our approach presents a compute-efficient alternative to black-box generative evaluation. By replacing stochastic token generation with a deterministic forward pass and a transparent linear classifier, we seek to enable scalable, deterministic, and reliable ITS evaluation.

2 Related Work

Evaluating the pedagogical quality of AI tutoring systems has become an active area of research. Early work examined whether models can behave like effective teachers (Tack and Piech, 2022; Tack et al., 2023), while later studies identified substantial reasoning and equity failures in less constrained tutoring settings (Macina et al., 2023b). This motivated richer tutoring datasets (Macina et al., 2023a) and grounded evaluation frameworks (Daheim et al., 2024; Wang et al., 2024a; Maurya et al., 2025), culminating in the BEA 2025 Shared Task on Pedagogical Ability Assessment (Kochmar et al., 2025), whose four-dimensional evaluation schema we adopt. Despite this progress, evaluation still relies heavily on costly expert annotation or automated LLM-based judging.

LLM-as-a-Judge (Zheng et al., 2023) offers a scalable alternative to human evaluation, but suffers from well-documented biases and reliability issues, including adversarial vulnerability (Chen et al., 2024), divergence from human judgments under prompt complexity and leniency bias (Thakur et al., 2025), and low intra-rater reliability across runs (Haldar and Hockenmaier, 2025). In parallel, probing research has shown that linguistic and task-relevant structure can often be linearly decoded from model representations (Tenney et al., 2019; Immer et al., 2022; Müller-Eberstein et al., 2022), with recent work demonstrating that probing can outperform generation-based LLM judges for preference extraction tasks (Maiya et al., 2025). Building on this line of work, we propose representation probing as a deterministic alternative that avoids generation-time failures entirely.

3 MRBench Dataset

For our analysis, we use the educational dialogue dataset MRBENCH_V3 introduced by Maurya et al. (2025). Each instance $j \in \{1, \ldots, N\}$ is

defined as $M^{(j)} = (C_h^{(j)}, \{(T_i^{(j)}, A_i^{(j)})\}_{i=1}^{m_j})$, where $C_h^{(j)}$ denotes the conversation history between a student and a tutor, and $T_i^{(j)}$ is the tutor's candidate response back to the student. That is, within each conversation thread, there are multiple candidate tutor response i produced by a mixture of LLMs, as well as a human Expert and Novice educator. Each response is associated with an annotation vector $A_i^{(j)} = [a_{i,1}^{(j)}, a_{i,2}^{(j)}, a_{i,3}^{(j)}, a_{i,4}^{(j)}]$, where $a_{i,k}^{(j)} \in \{\textit{Yes}, \textit{No}, \textit{To Some Extent}\}$ corresponds to four pedagogical dimensions $dim \in \{\textit{Mistake Identification}, \textit{Mistake Location}, \textit{Providing Guidance}, \textit{Actionability}\}$. We provide a more detailed analysis of the dataset in Appendix A below.

The dataset contains 300 annotated conversations, split into 200 training and 100 evaluation threads.

4 Extracting Hidden Representation From Frozen LLMs

For each training sample $M_{train}^{(j)}$, we construct a fixed input that we will then use throughout all our experiments. That is, given an evaluation dimension and a candidate response $T_i^{(j)}$, the formatted input is:

$$x_{i,dim}^{(j)} = \text{Format}\Big(C_h^{(j)},\ \text{def}_{dim},\ \text{rubric}_{dim},\ T_i^{(j)}\Big)$$

where def_{dim} and rubric_{dim} are the textual definition and rubric for dimension dim, and *Format* denotes the population of these information into the template, presented in Appendix B.

This input is then passed to a frozen language model for encoding to compute a latent representation $h_l^{(k)}$ at each token position $l \in \{1, \ldots, L\}$ and layer $k \in \{1, \ldots, K\}$. We extract the hidden state corresponding to the final token at the final layer:

$$\mathbf{z}_{i,d}^{(j)} = h_L^{(K)} \in \mathbb{R}^D,$$

where D denotes the hidden dimension of the model. We then apply L2 normalization to obtain

$$\bar{\mathbf{z}}_{i,d}^{(j)} = \frac{\mathbf{z}_{i,d}^{(j)}}{\|\mathbf{z}_{i,d}^{(j)}\|_2}.$$

This normalization removes variation in representation magnitude across inputs, ensuring that downstream similarity and classification behavior depends primarily on the direction of the representation rather than its scale.

4.1 Building the Linear Probe

The core of our approach is a linear probe, a lightweight classifier trained on the extracted representations to predict the expert-annotated quality label for each evaluation dimension. A detailed justification for this design choice is provided in Appendix D. Before fitting the logistic regression classifier, the normalized representations are standardized using statistics computed exclusively on the training split:

$$\tilde{\mathbf{z}} = \frac{\bar{\mathbf{z}} - \boldsymbol{\mu}_{\text{train}}}{\boldsymbol{\sigma}_{\text{train}}},$$

where $\boldsymbol{\mu}_{\text{train}}$ and $\boldsymbol{\sigma}_{\text{train}}$ denote the per-dimension mean and standard deviation estimated from the training data.

Even after L2 normalization, without standardization, high-variance dimensions can dominate optimization and slow convergence of the logistic regression objective. Applying the scaler therefore improves optimization stability while also preventing leakage of evaluation-set statistics into the classifier.

We then train a separate linear probe for each of the four evaluation dimensions, implemented as a multinomial logistic regression classifier over three ordinal labels: *Yes* (1), *To Some Extent* (2), and *No* (3).

Given a standardized representation $\tilde{\mathbf{z}} \in \mathbb{R}^D$ and probe parameters $\mathbf{W}_d \in \mathbb{R}^{3\times D}$ and $\mathbf{b}_d \in \mathbb{R}^3$ for evaluation dimension d, the predicted class probabilities are computed as:

$$P(y = c \mid \tilde{\mathbf{z}}; \mathbf{W}_d, \mathbf{b}_d) = \frac{\exp\big(\mathbf{w}_{d,c}^\top \tilde{\mathbf{z}} + b_{d,c}\big)}{\sum_{c'=0}^{2} \exp\Big(\mathbf{w}_{d,c'}^\top \tilde{\mathbf{z}} + b_{d,c'}\Big)}$$

$$\hat{y} = \arg\max_{c\in\{0,1,2\}} P(y = c \mid \tilde{\mathbf{z}}; \mathbf{W}_d, \mathbf{b}_d)$$

The complete probe pipeline is therefore:

$$\mathbf{z} \xrightarrow{\text{L2 Normalize}} \bar{\mathbf{z}} \xrightarrow{\text{Standardize}} \tilde{\mathbf{z}} \xrightarrow{\text{Logistic Regression}} \hat{y}.$$

4.2 Baselines and Evaluation Metrics

For a diagnostic baseline to account for a severe class imbalance in the dataset, we first use a majority-class baseline that always predict "Yes". Next, to assess how much pedagogical quality

can be inferred from surface-level textual features alone, we replace the frozen LLM encoder with a classical TF-IDF representation and train the same logistic regression classifier on top of it. Finally, we compare our approach against the standard LLM-as-a-Judge paradigm by providing the same evaluation template and parsing the generated response for both the reasoning and the predicted label (i.e., 1 for "Yes", 2 for "To Some Extent", and 3 for "No"). That is, for all models used to construct the probe, we also evaluate them as generative judges to enable a direct one-to-one comparison. However, smaller edge models may fail to generate coherent reasoning or outputs that can be reliably parsed. To address this limitation, we additionally introduce an "Oracle Judge" based on OpenAI's GPT-5.5.

We report three evaluation metrics. We begin with accuracy, as it is the most interpretable measure. However, due to the dataset heavy class imbalance, accuracy alone can be misleading, so we additionally report macro-averaged F1. Most importantly still, we report Cohen's Kappa (κ), which measures agreement between predicted and ground-truth labels while correcting for any agreement that may occur by chance. Since it accounts for both class imbalance and chance agreement, we treat Cohen's Kappa as the primary metric in our analysis.

4.3 Model Selection

We select a diverse set of open-source models to ensure robustness across potential model-specific artifacts. Specifically, we cover four axes of variation, namely architecture (Dense vs. MoE), reasoning modality, parameter scale, and model family. To avoid self-preference bias (Chen et al., 2025), we exclude models that overlap with those used to generate the tutor's final response in the evaluation set. We further mitigate training-data contamination by restricting selection to models with pre–MRbench_V3 knowledge cutoffs. However, we purposely include Gemma-4 4B and 31B, which postdate the dataset release, to investigate whether potential contamination affects linear probe performance. Full details are provided in Appendix E.

Our final list of models include:

- *Qwen3-4B-Instruct-2507* & *Qwen3-4B-Thinking-2507* (Yang et al., 2025)
- *Phi-3-medium-128k-instruct* & *Phi-3.5-MoE-instruct* (Abdin et al., 2024)
- *GPT-OSS-20b* (OpenAI et al., 2025)
- *Qwen3-30B-A3B-Instruct-2507* & *Qwen3-30B-A3B-Thinking-2507* (Yang et al., 2025)
- *Gemma-4-E4B-it* & *Gemma-4-31B-it* (Farabet, 2026)

5 Control Tests

In order to establish whether the observed results are genuine, that is frozen pre-trained LLMs can encode useful signals of pedagogical quality and that it can be linearly decoded, we run four separate control tests.

Random Train/Test Vectors First, we determine whether the geometry of the latent representation actually matters by replacing the real output embedding vectors with ones that are drawn independently from a standard Gaussian distribution for both the train and evaluation task. Formally, for each instance with real representation $\tilde{\mathbf{z}} \in \mathbb{R}^D$, we substitute:

$$\tilde{\mathbf{z}}_{rand} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$$

where $\mathbf{I}_D$ is the $D \times D$ identity matrix.

Permutation Test To verify that the relationship between latent representations and ground-truth labels is genuine rather than spurious, we conduct a permutation test following Ojala and Garriga (2010). We randomly shuffle the training labels y_k while keeping the representation vectors $\tilde{\mathbf{z}}_k$ fixed. This breaks the conditional dependence between representations and labels while preserving the marginal distribution of the labels. We train a probe on this shuffled training set and evaluate it on the original, unshuffled test set, repeating this process 1,000 times to construct a null distribution.

Selectivity Control Task A key concern in any probing study is that high dimensional representation spaces may allow linear classifiers to fit arbitrary labeling unrelated to the target property. Following Hewitt and Liang (2019), we introduce a selectivity control task to further address this beyond the permutation test.

We replace gold labels with deterministic pseudo-labels to maintain a well-defined learning

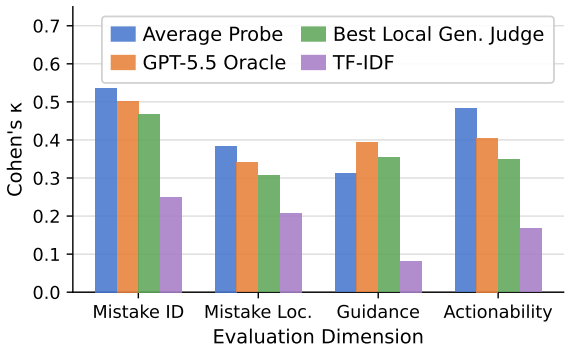

Figure 2: Comparison of linear probe performance against alternative evaluation baselines. For the linear probe, we report the mean Cohen's κ across all nine models for each dimension. *Best Local Gen. Judge* denotes the highest-performing generative judge among the locally evaluated models for a given dimension. The proposed linear probe consistently outperforms all judges, including GPT-5.5, across all dimensions except *Providing Guidance*.

problem. These are assigned via a hash over the response content T_i, evaluation dimension d, and a fixed seed (i.e., 42):

$$y_{ctrl}(T_i, d) = \text{SHA256}(\texttt{seed} \,\|\, T_i \,\|\, d) \bmod K,$$

where $K = 3$.

Layer 1 Embedding as Training Vector As a final control, we examine whether encoding through the full LLM depth is necessary by replacing last-layer embeddings with first-layer embeddings as the probe input. Unlike random Gaussian features, first-layer representations still contain substantial structure, as early layers of LLMs are known to capture basic lexical and semantic information (Skean et al., 2025). This control isolates the contribution of deeper contextualization provided by higher layers.

6 Results

6.1 Linear Decodability of Pedagogical Quality from LLM Representations

Table 1 reports the full encoder-based evaluation results across four pedagogical dimensions and nine evaluated models. The primary finding is straightforward: with the exception of "Providing Guidance", linear probes consistently outperform all baselines shown in Figure 2, including TF-IDF classifiers, local generative judges, and the proprietary GPT judge. This pattern holds across nearly all models and evaluation dimensions, with even weaker probe configurations often matching or exceeding the strongest generative baselines.

"Mistake Identification" yields the strongest overall performance. Although this dimension is highly imbalanced, inflating raw accuracy across methods (majority baseline: 0.771), the probes remain clearly superior on the more informative κ metric. For example, the TF-IDF classifier achieves $\kappa = 0.250$, while the GPT judge reaches $\kappa = 0.502$. In contrast, our best probe exceeds the GPT judge by up to +0.66 κ. This suggests that identifying student mistakes is strongly encoded in LLM representations, likely because corrective feedback relies on stable linguistic patterns that are heavily represented during pre-training.

"Actionability" also demonstrates a substantial gap between encoder-based and lexical baselines. While the TF-IDF classifier achieves only $\kappa = 0.166$, indicating that surface lexical cues are insufficient for the task, representation-based evaluation consistently achieves much stronger agreement scores than both lexical and generative judges.

By contrast, "Providing Guidance" is the only dimension where probes do not consistently dominate. We hypothesize that this stems from ambiguity in the construct itself. Compared to the other rubric dimensions, it is operationally less precise (Appendix A.3) and therefore less likely to correspond to a clean linearly separable direction in representation space.

This findings suggest a broader methodological distinction between encoder-based and generative evaluation. While the probe can perform significantly well, generative judges may still be better suited to rubric dimensions that require flexible interpretation. However, as we discuss next, the flexibility of auto-regressive generation may also introduce additional sources of evaluation error.

6.2 Encoded Pedagogical Quality Signals Persist Beyond Model-Specific Artifacts and Under Control Tests

Figure 3 shows a modest but consistent performance increase as model scale grows from 4B to 31B parameters. However, the gains are small, with the largest models improving by less than 0.04 $\hat{\kappa}$. We also observe no systematic differences between dense transformer and Mixture-of-Experts architectures, nor between "Thinking" and "Non-thinking" variants. Together, these results suggest that the pedagogical signal captured

Model	Mistake Identification			Mistake Location			Providing Guidance			Actionability		
	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ
Models												
Gemma-4-E4B-it †	0.792	0.596	0.470	0.693	0.523	0.401	0.574	0.514	0.280	0.651	0.586	0.419
Gemma-4-31B-it †	0.828	0.623	0.551	0.671	0.513	0.384	0.588	0.520	0.287	0.703	0.593	0.508
Qwen3-4B-Think	0.826	0.643	0.551	0.662	0.505	0.366	0.595	0.533	0.310	0.675	0.594	0.464
Qwen3-4B-Inst	0.791	0.596	0.488	0.664	0.506	0.391	0.597	0.527	0.318	0.699	0.584	0.492
Phi-3-medium-128k-Inst	0.826	0.623	0.551	**0.696**	**0.537**	**0.409**	0.616	0.551	0.342	**0.715**	**0.636**	**0.530**
Phi-3.5-MoE-Inst	**0.835**	0.620	**0.566**	0.658	0.487	0.371	0.591	0.526	0.308	0.710	0.627	0.522
GPT-OSS-20B	0.832	**0.626**	**0.566**	0.637	0.502	0.345	0.588	0.520	0.299	0.655	0.561	0.426
Qwen3-30B-A3B-Inst	0.818	0.611	0.542	0.670	0.535	0.389	0.618	0.555	0.361	0.700	0.624	0.501
Qwen3-30B-A3B-Think	0.818	0.636	0.546	0.669	**0.537**	0.400	0.584	0.520	0.300	0.680	0.606	0.476
Baselines												
GPT-5.5 (oracle)	0.741	0.542	0.502	0.649	0.483	0.341	**0.702**	**0.556**	**0.394**	0.627	0.460	0.405
TF-IDF	0.609	0.463	0.250	0.526	0.436	0.207	0.397	0.373	0.080	0.468	0.430	0.166
Majority Baseline	0.771	—	—	0.616	—	—	0.572	—	—	0.512	—	—

Table 1: Main linear probe performance across all four dimensions. Best per-dimension result highlighted in **bold**. *Majority Baseline* predicts the most frequent class (i.e., "Yes"), with F1 and κ undefined under constant predictions (—). *GPT-5.5 (oracle)* serves as an upper-bound for generative judge evaluation. "Think" = thinking/reasoning variant; "Inst" & "it" = instruct variant; † denotes models released after MRBench_V3 published date included for possible data contamination effects.

Dimension	Rand $\hat{\kappa}\downarrow$	Perm $\hat{\kappa}\downarrow$	Select $\uparrow$	L$\Delta\hat{\kappa}\uparrow$
Mistake ID	0.019	0.002^{***}	0.496	0.155
Mistake Loc.	-0.009	-0.000^{***}	0.364	0.138
Guidance	-0.004	-0.000^{***}	0.279	0.096
Actionab.	0.003	-0.000^{***}	0.458	0.133

Table 2: Mean Cohen's κ across models per dimension. Rand = random input vectors for probe training; Perm = label-permutation baseline (all $\hat{\kappa}$ significant at $p < 0.001$). Select = $\hat{\kappa}_{\text{probe}} - \hat{\kappa}_{\text{control}}$, measuring selectivity for task-relevant signal. L$\Delta\hat{\kappa}$ = $\hat{\kappa}_{\text{last}} - \hat{\kappa}_{\text{layer 1}}$, measuring improvement from final layer over layer 1.

by the representations saturates at relatively modest scales rather than depending on specific architectural features. Importantly still, all models remain robust across the full suite of control experiments summarized in Table 2. Full details are provided in Appendix G.

Random Vector Under the random vector control, the discriminative collapse becomes clear as $\hat{\kappa}$ drops to near zero (0.003–0.019) across all models. Further detailed in Table 8, this indicates that the classifiers learn little beyond the marginal class distribution from uninformative embeddings.

Selectivity Control Task Across all dimensions and models, the main probes exhibit a large and consistent selectivity gap, $\kappa_{real} - \kappa_{control}$, demonstrating that the classifiers decode information specifically present in the embedding geometry rather than exploiting generic properties of high-dimensional linear classifiers.

Permutation Test The null distribution of κ is tightly centered around zero under our permutation tests, with standard deviations between only 0.029 and 0.037. All reported p-values are 0.000 at the $\alpha = 0.01$ threshold, confirming that every probe result in the main experiment is statistically significant.

Last vs. First Layer Our result shows that the first transformer layer already recovers most of the final probing performance, consistent with prior work on how LLMs encode substantial semantic information early in the network (Afzal et al., 2025; Ju et al., 2024). Nevertheless, propagating representations through the full architecture yields meaningful gains, with $L\Delta\kappa$ improving by at least 0.096 on the weakest dimension ("Providing Guidance") and up to 0.155 on the strongest ("Mistake Identification"). These improvements suggest that deeper layers continue refining the geometry of the embedding space in ways that improve downstream separability and probe performance.

6.3 Linear Probes Outperform Generative Judges in Performance and Reliability

Figure 4 compares the performance of our approach against generative judges. Only results of the "Instruct" variants of the models are shown, as all "Thinking" models fail to produce parsable

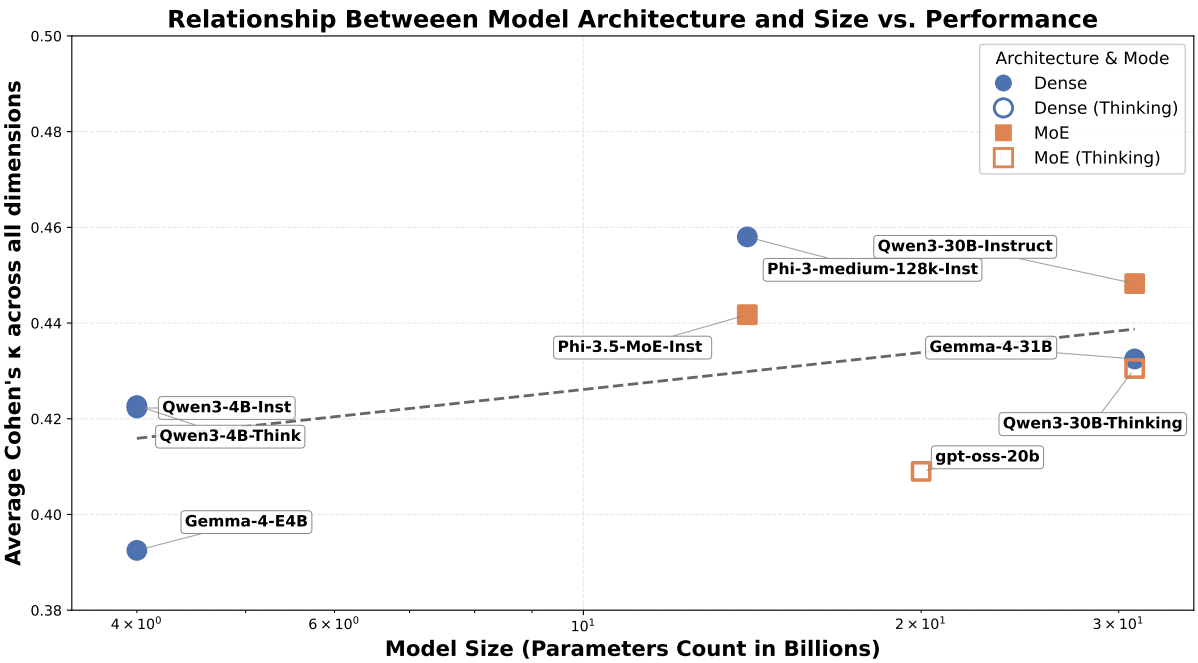

Figure 3: Relationship between model architecture and size versus linear probe performance. Average Cohen's κ across all four pedagogical dimensions is plotted against model size (in billions of parameters) for dense and MoE architectures, with and without thinking mode. The trend line indicates negligible scaling benefit, with performance gains of less than 0.04 κ across the 4B–31B range. No systematic differences are observed across architectures or thinking modalities, suggesting that pedagogical quality signals saturate at relatively modest model scales.

outputs. The linear probe consistently outperforms its generative counterparts, achieving higher κ scores with substantially lower computational cost. With this, we seek to highlight three fundamental limitations of generative LLM-based evaluation. We provide a deeper analysis in Appendix F.

Knowledge Cutoff Effects in Generative but Not Encoder-Based Evaluation First, we observe a consistent pattern in the generative evaluation setting. Gemma-4, whose training cutoff post-dates the release of MRBench_V3, matches or outperforms substantially larger judge models, with the primary exception being the larger 31B Gemma-4 variant itself. While the underlying cause remains unclear, this advantage may stem from a combination of factors, which certainly includes benchmark exposure, but also finer-grained architectural differences, training objectives, or training data quality. Nevertheless, the temporal relationship between model release and benchmark availability suggests that post-release knowledge may influence generative judge performance.

Interestingly, this pattern disappears under our approach. Unlike the generative setting, we ob-

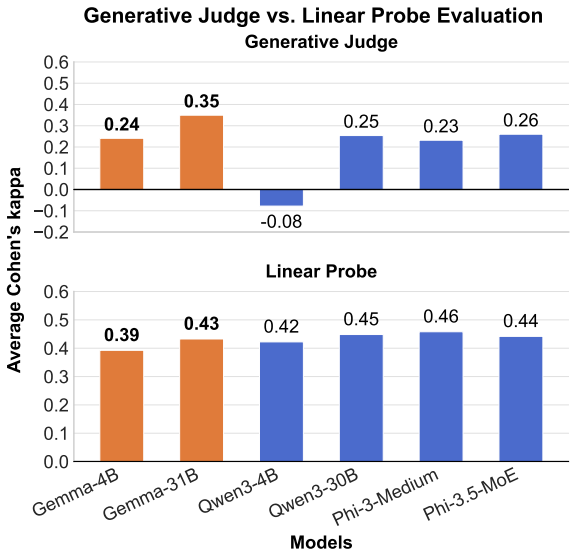

Figure 4: Average Cohen's κ across all evaluation dimensions for probe vs. local generative judge. Orange and blue bars denote models with knowledge cutoffs after and before the release of MRBench_V3 respectively. All thinking-mode judges failed during generative evaluation.

serve no consistent advantage associated with model scale, architecture family, or knowledge cutoff. This suggests that probe-based methods rely less on benchmark-specific memorization and more on the information encoded in LLM representations, learned through massive pre-training or reinforcement learning data. These findings indicate that this evaluation approach may provide a more robust and transferable signal across datasets and model families.

Reasoning-Label Inconsistency Second, even with the Oracle GPT judge, we observe systematic inconsistencies between generated rationales and final labels provided (Appendix F.1). In several cases, the reasoning provided does not logically support the assigned class, highlighting a key limitation of generative evaluation, where auto-regressive reasoning does not guarantee consistency between intermediate explanations and final decisions.

Thinking Models Failure at Smaller Parameter Scale Finally, reasoning-capable models at or below 30B parameters frequently fail to produce reliably parsable outputs due to two recurring issues: excessively verbose generations that never resolve to a valid label (i.e., such as "doom looping"), and role hallucination, where the model continues the conversation instead of performing evaluation (Appendix F.2). In practical deployment settings, these failures create a non-trivial source of evaluation instability while still incurring the full cost of auto-regressive generation.

7 Pedagogical Signal Decomposition Under Response-Only Conditions

To examine whether evaluative dimensions depend on global historical context or localized linguistic features, we repeat the experiments under the same setup and models but replace the full prompt with a minimal template containing only the instructions and the latest tutor response, omitting conversation history, rubric definitions, and related context. The ablation template is provided in Appendix C.

Our results, shown in Figure 5 and Table 10, reveal an asymmetric degradation across dimensions when relevant context is removed. While all four dimensions exhibit a drop in Cohen's κ, the extent of the decline depends on whether the task is primarily self-contained or relational in nature.

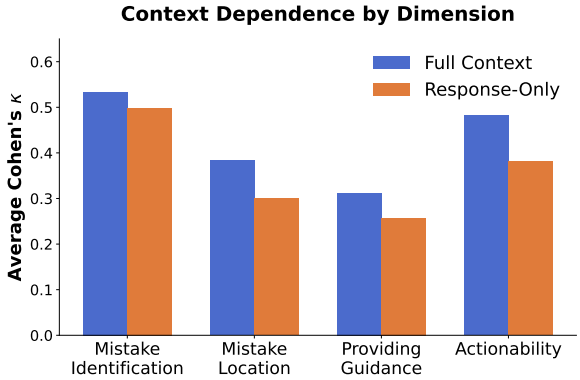

Figure 5: Average Cohen's κ across all evaluation dimensions for each probe. Across all dimensions, a monotonic drop in performance is observed after removing relevant context such as conversation history and rubric definition.

"Actionability" experiences the largest absolute decrease ($\Delta\hat{\kappa} = -0.101$), yet remain strong ($\hat{\kappa} = 0.381$) indicating that response-only features retain substantial predictive signal. This robustness suggests its signal is largely encoded in local linguistic properties of the tutor response itself.

In contrast, "Mistake Location" exhibits a qualitatively different failure mode. Starting from a lower baseline under full context ($\hat{\kappa} = 0.383$), performance drops by $\Delta\hat{\kappa} = -0.083$ to $\hat{\kappa} = 0.300$, approaching the lower bound of reliable agreement. This degradation indicates that "Mistake Location" is fundamentally relational. Correctly identifying an error requires alignment between the response and its preceding context, and without this reference frame, the task becomes underdetermined. Performance drops when alignment structure that supports this judgment is no longer accessible to the representation.

8 Discussion and Conclusion

In this work, we show that LLM hidden states inherently encode pedagogical quality, and that this signal emerges even in models as small as 4B parameters. Across model scales and architectures, linear probes reliably extract this information and consistently outperform generative judges. Control experiments further indicate that probes capture genuine pedagogical signals rather than exploiting superficial dataset artifacts.

Overall, representation-based probing provides a deterministic, efficient, and interpretable alternative for evaluating future intelligent tutoring systems.

Limitations

Prompting Strategy for Generative Judges To ensure strict consistency and reproducibility, our evaluation relies on the standard prompting formats provided by the MRBench_V3 dataset. However, prior work has shown that prompting techniques such few-shot prompting (Brown et al., 2020) can systematically enhance LLM judge performance. While exploring the full upper bound of these advanced prompting variants is beyond the aim of this current study, investigating the exact margin of improvement they offer remains a critical direction for future work.

Dataset Scope The evaluation in this work is bounded by the scope of MRBench_V3, which primarily captures tutoring dialogues within STEM fields (mathematics and science). While these domains represent a substantial portion of educational discourse, they feature highly structured, objective reasoning patterns. Consequently, our current findings may not fully generalize to more subjective educational domains, such as literature or philosophy, where dialogue dynamics differ fundamentally. Validating our approach across a broader, multi-disciplinary spectrum of educational contexts is an important next step.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Anum Afzal, Florian Matthes, Gal Chechik, and Yftah Ziser. 2025. Knowing before saying: LLM representations encode information about chain-of-thought success before completion. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12791–12806, Vienna, Austria. Association for Computational Linguistics.

Liquid AI. 2026. Lfm2.5-1.2b-thinking: On-device reasoning under 1gb — liquid ai.

Jiyuan An, Xiang Fu, Bo Liu, Xuquan Zong, Cunliang Kong, Shuliang Liu, Shuo Wang, Zhenghao Liu, Liner Yang, Hanghang Fan, and Erhong Yang. 2025. BLCU-ICALL at BEA 2025 shared task: Multi-strategy evaluation of AI tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1084–1097, Vienna, Austria. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.

Zhi-Yuan Chen, Hao Wang, Xinyu Zhang, Enrui Hu, and Yankai Lin. 2025. Beyond the surface: Measuring self-preference in LLM judgments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1653–1672, Suzhou, China. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.

Yuming Fan, Chuangchuang Tan, and Wenyu Song. 2025. BJTU at BEA 2025 shared task: Task-aware prompt tuning and data augmentation for evaluating AI math tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1073–1077, Vienna, Austria. Association for Computational Linguistics.

Clement Farabet. 2026. Gemma 4: Byte for byte, the most capable open models.

Rajarshi Haldar and Julia Hockenmaier. 2025. Rating roulette: Self-inconsistency in LLM-as-a-judge frameworks. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24986–25004, Suzhou, China. Association for Computational Linguistics.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13806–13834, Bangkok, Thailand. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Alexander Immer, Lucas Torroba Hennigen, Vincent Fortuin, and Ryan Cotterell. 2022. Probing as quantifying inductive bias. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1839–1851, Dublin, Ireland. Association for Computational Linguistics.

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? a layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8235–8246, Torino, Italia. ELRA and ICCL.

Ekaterina Kochmar, Kaushal Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 1011–1033, Vienna, Austria. Association for Computational Linguistics.

Zichao Li, Yanshuai Cao, and Jackie CK Cheung. 2024. Do LLMs build world representations? probing through the lens of state abstraction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023a. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. MathTutorBench: A benchmark for measuring open-ended pedagogical capabilities of LLM tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 204–221, Suzhou, China. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023b. Opportunities and challenges in neural dialog tutoring. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2357–2372, Dubrovnik, Croatia. Association for Computational Linguistics.

Sharan Maiya, Yinhong Liu, Ramit Debnath, and Anna Korhonen. 2025. Improving preference extraction in LLMs by identifying latent knowledge through classifying probes. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9061–9081, Vienna, Austria. Association for Computational Linguistics.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *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)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022. Probing for labeled dependency trees. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7711–7726, Dublin, Ireland. Association for Computational Linguistics.

Numaan Naeem, Kaushal Kumar Maurya, Kseniia Petukhova, and Ekaterina Kochmar. 2026. AITutor-EvalKit: Exploring the capabilities of AI tutors. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 457–479, Rabat, Marocco. Association for Computational Linguistics.

Markus Ojala and Gemma C. Garriga. 2010. Permutation tests for studying classifier performance. *J. Mach. Learn. Res.*, 11:1833–1863.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid

Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*.

Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 785–795, Toronto, Canada. Association for Computational Linguistics.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *ArXiv*, abs/2205.07540.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *ArXiv*, abs/1905.06316.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024a. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

Shan Wang, Fang Wang, Zhen Zhu, Jingxuan Wang, Tam Tran, and Zhao Du. 2024b. Artificial intelligence in education: A systematic literature review. *Expert Systems with Applications*, 252:124167.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

A MRBench_V3 Analysis

We provide a sample instance from the dataset below. The original dataset also contains "Ground Truth", the correct answer to the student's problem within the conversation history. For this study, we choose to drop this information as it is not relevant to our analysis.

A.1 Example Conversation Thread

We provide an example of a conversation thread used for our evaluation and experiments. The LLM Annotations of each category (i.e., Yes, To Some Extent, No) are provided to showcase the difference in quality of a tutor's response.

Conversation History

Tutor: Hi, could you please provide a step-by-step solution for the question below?
The question is: *Elliott is trying to walk 10,000 steps a day. He finished half of his steps on his walks to and from school and did another 1,000 steps going for a short walk with his friend. He also went for a short jog around the block and realized that after he had finished his jog, he only had 2,000 steps left to take. How many steps did Elliott take during his jog?*

Student: Elliott finished half of his steps on his walks to and from school, so he took $10,000/2 = 5,000$ steps.
Adding the 1,000 steps he took with his friend, he has taken $5,000 + 1,000 = 6,000$ steps.
Subtracting 6,000 from his goal of 10,000, he has $10,000 - 6,000 = 4,000$ steps left to take.
Therefore, he took 4,000 steps during his jog.

Tutor: Can you tell me how you got to your answer?

Student: Sure. I started by figuring out how many steps Elliott had taken before his jog. I knew he had finished half of his steps on his walks to and from school, so I calculated $10,000/2 = 5,000$ steps. Then I added the 1,000 steps he took with his friend, giving $5,000 + 1,000 = 6,000$ steps. After that, I subtracted 6,000 from 10,000 so I had $10,000 - 6,000 = 4,000$ steps left. This meant that he had taken 4,000 steps during his

jog.

A.2 LLM Annotations

- **Phi3:** Great job! To further enhance your understanding, let's explore more examples and practice solving similar problems together.
  - Mistake Identification: No
  - Mistake Location: No
  - Providing Guidance: No
  - Actionability: No
- **Expert:** But after his jog he still had 2,000 steps left.
  - Mistake Identification: Yes
  - Mistake Location: No
  - Providing Guidance: Yes
  - Actionability: Yes
- **Sonnet:** Great job showing your work, but let's take another look at the last part of the problem statement to make sure we're considering all the information given.
  - Mistake Identification: No
  - Mistake Location: No
  - Providing Guidance: To some extent
  - Actionability: To some extent

A.3 Dimension Definition

We adopt the original definitions of each evaluation dimension as specified by Maurya et al. (2025).

Mistake Identification This dimension checks whether the tutor directly signals that the student's response is incorrect. Because each dialogue contains an error, valid feedback should engage with the mistake rather than ignoring it or implicitly accepting the response as correct. It reflects the most basic level of corrective behavior.

Mistake Location Here, we evaluate whether the tutor goes beyond acknowledging an error and indicates where the problem lies in the student's answer. Providing this form of localization helps reduce ambiguity for the student and supports targeted revision.

Providing Guidance This dimension focuses on whether the tutor offers helpful instructional support without revealing the final answer. Effective responses typically include hints, clarifications, or prompts that steer the student's thinking while leaving the core reasoning task intact.

Actionability This dimension captures whether the tutor's message translates into a clear and usable next step. Responses that are indirect, overly vague, or that do not meaningfully advance problem-solving are considered weak on this dimension, even if partially relevant content is present.

Each dimension is annotated using a three-point ordinal scale: **Yes** (fully satisfied), **To Some Extent** (partially satisfied), and **No** (not satisfied). This formulation reflects the graded nature of instructional quality rather than a binary judgment. Inter-annotator agreement is measured at Cohen's $\kappa = 0.65$, indicating substantial consistency among annotators and supporting the reliability of the annotations.

A.4 Train-Test Data Split

Table 3: Class distribution across annotation dimensions for the MRBench training and evaluation splits. Labels: 1 = *Yes*, 2 = *To Some Extent*, 3 = *No*. All labels are produced by human expert annotators, with inter-rater agreement of $\kappa = 0.65$ representing substantial agreement.

Split	N	Mist_ID	Mist_Loc	Guid.	Action.
Train	1,654	1: 1298 2: 112 3: 244	1: 1037 2: 145 3: 472	1: 937 2: 346 3: 371	1: 889 2: 246 3: 519
Eval	822	1: 634 2: 62 3: 126	1: 506 2: 75 3: 241	1: 470 2: 157 3: 195	1: 421 2: 123 3: 278

Table 3 shows a strong class imbalance in the dataset. To preserve label proportions, we construct the train/test split using stratified sampling, resulting in 200 training and 100 evaluation conversations. The training set contains $N = 1{,}654$ tutor responses with annotations, while the evaluation set contains $N = 822$.

B LLM Input Template

LLM Judge Prompt Template

System:
You are a critic evaluating a tutor's interaction with a student, responsible for providing a clear and objective single evaluation score based on specific criteria. Each assessment must accurately reflect the absolute performance standards.

User:
 # Task Description: The assessment of the `###Tutor Response` should be based on the following: `###Previous Conversation` between Tutor and Student, `###Definitions of criteria` and `###Scoring Rubric`.

1. Write a one-sentence feedback that assesses the quality of the response and Rate the Tutor Response strictly based on the given scoring rubric and criteria.
2. After writing feedback, write a score that is an integer between 1 and 3. You should refer to the scoring rubric.
3. Output format: `"Feedback: (feedback) [RESULT] (integer 1-3)"`
4. Please do not generate other opening, closing, or explanations.

 # Previous Conversation: {`history`}
 # Definitions of criteria: {`definition`}
 # Scoring Rubric: {`rubric`}
 # Tutor Response: {`response`}

Assistant:
 # Generate Assessment Score:

Dimension-Specific Rubrics

Each dimension is evaluated individually via separate prompts, and so for each one, we append the corresponding rubric.

Mistake Identification
[Has the tutor identified a mistake in a student's response?]
Score 1: Yes Score 2: To some extent Score 3: No

Mistake Location
[Does the tutor's response accurately point to a genuine mistake and its location?]
Score 1: Yes Score 2: To some extent Score 3: No

Providing Guidance
[Does the tutor offer correct and relevant guidance?]
Score 1: Yes (correct/relevant) Score 2: To some extent (incomplete/incorrect) Score 3: No

Actionability
[Is it clear what the student should do next?]
Score 1: Yes Score 2: To some extent Score 3: No

C Response Only Ablation

In the ablation study, we remove all the information of conversation history and rubrics text. The prompt template is provided below.

Prompt Template: Response-Only Evaluation

System:
You are a critic evaluating a tutor's interaction with a student, responsible for providing a clear and objective single evaluation score based on specific criteria. Each assessment must accurately reflect the absolute performance standards.

User:

1. Write a one-sentence feedback that assesses the quality of the response and Rate the Tutor Response strictly based on the given scoring rubric and criteria.
2. After writing feedback, write a score that is an integer between 1 and 3. You should refer to the scoring rubric.
3. Output format: `"Feedback: (feedback) [RESULT] (integer 1-3)"`
4. Please do not generate other opening, closing, or explanations.

 # Tutor Response: {`response`}

Assistant:
 # Generate Assessment Score:

D Selecting Linear Probe as Classification Head & Handling Class Imbalance

D.1 Minimizing Probe Capacity

The goal of a probing study is to ask whether information about a target property, in our case, pedagogical quality, is encoded in a model's representations. A positive result (i.e., the probe performs well) is only meaningful if the classifier used is simple enough that it cannot manufacture the information itself through complex nonlinear computation. Further complex methods such as multilayer perceptron risks conflating any positive result found to the underlying capacity of the probe itself to fit to the data.

This is especially the case, when dealing with such small dataset. This reasoning is consistent with the probing literature in NLP, where linear classifiers are the standard choice precisely because they limit the probe's own expressive power, making its performance a reliable indicator of the underlying model's internal representations rather than its capacity.

D.2 Handling Class Imbalance

As described in the dataset section, MRBench_V3 exhibits a pronounced class imbalance across all four dimensions, with the *Yes* class dominating in both the training and evaluation splits. A classifier that simply predicts the majority class for every input would achieve high accuracy while providing no useful information. To counteract this, we automatically adjusts the loss contribution of each training example inversely proportional to its class frequency. Concretely, if class c has frequency n_c out of n training examples, the weight assigned to examples of that class is:

$$w_c = \frac{n}{K \cdot n_c}$$

where $K = 3$ is the number of classes. Our goal is to encourage the classifier to attend to the minority classes and prevents the solution from collapsing to a majority-class predictor.

E Model Summary

We present more detailed information on the models chosen for our experiments below in Table 4 below.

Table 4: Overview of the evaluated language models, including developer, architecture type, parameter count, knowledge cutoff, and support for explicit reasoning/thinking mode. MRBench_V3 was released to public repository in December 2024; models marked with † have knowledge cutoffs after the dataset release.

Model	Developer	Arch.	Param.	Cutoff	Think
Gemma-4-E4B†	Google	Dense	4B	Jan 25	N
Gemma-4-31B†	Google	Dense	31B	Jan 25	N
Qwen3-4B-Inst	Alibaba	Dense	4B	Oct 24	N
Qwen3-4B-Think	Alibaba	Dense	4B	Oct 24	Y
Phi-3-medium-128k-Inst	Microsoft	Dense	14B	Oct 23	N
Phi-3.5-MoE-Inst	Microsoft	MoE	14B	Oct 23	N
GPT-OSS-20B	OpenAI	MoE	21B	Jun 24	Y
Qwen3-30B-A3B-Inst	Alibaba	MoE	30.5B	Oct 24	N
Qwen3-30B-A3B-Think	Alibaba	MoE	30.5B	Oct 24	Y

F Qualitative Analysis of Generative Judges Failure Modes

For the generative judge, we use vLLM as our inference engine. All models are loaded with a maximum context length of 8,192 tokens. Generations are sampled with a temperature of 0.3, top p of 0.95, and a maximum output length of 4,096 tokens.

F.1 Inconsistency Between Reasoning & Labels

Even the proprietary GPT judge, used here as an oracle baseline, exhibits non-trivial inconsistency. An illustrative case from the evaluation data involves Conversation ID 2672-c8c14096-d003-49fc-8aae-371660962a45, where a tutor response (i.e., from Llama-3-18B) on the Actionability dimension received the following judge reasoning:

> **Judge Reasoning:** *"The response is misleading because it introduces an error and does not clearly guide the student toward the correct next step, so the next action is not well defined."*
>
> **Assigned Score:** 1 (*Yes*) — label for responses that *do* provide a clear and actionable step for the student.

Despite generating an explicitly negative critique, the judge assigned a score of 1 ("Yes"), erroneously indicating the presence of a clear and actionable step. This stark reasoning-to-label decoupling is unusual for a frontier model, and we hypothesize it stems from increased RLHF-induced sycophancy. Standard alignment procedures incentivize models to behave agreeably, creating a latent bias toward affirmative classifications in ambiguous contexts. We present the empirical results in Table 5 to corroborate this hypothesis. As shown, the oracle judge achieves a high F1 score on the "Yes" class but fails significantly on the more nuanced "To some extent" label.

Table 5: Per-Class F_1 Scores Across Evaluation Dimensions For Oracle GPT-5.5 Judge

Dimension	Yes	To some extent	No
Mistake ID	78.74%	18.69%	65.20%
Mistake Loc.	63.58%	21.19%	60.25%
Guidance	75.19%	35.10%	55.08%
Actionability	66.17%	36.67%	35.26%

Still, the most striking failure in this mode of the local judge results is not random mislabeling but systematic label inversion. The Qwen3-4B-Instruct as a generative judge achieves κ = -0.024 on Mistake Identification, κ = -0.177 on Mistake Location, κ = -0.036 on Providing Guidance, and κ = -0.075 on Actionability.

We suspect that at 4B parameters (contrast to Gemma 4 that could potentially be exposed to the dataset during training), the model lacks sufficient instruction-following capability to reliably map a rubric specification onto a constrained output format. When confronted with a three-level

ordinal scale and a structured rubric template, the model's generation process produces label assignments that are systematically offset from expert annotation, arguably because the model has learned to associate certain surface features of tutoring dialogues with certain label tokens in ways that are inverted relative to the intended semantics.

Here, we provide two examples where the reasoning are provided in the exact same negative sentiment yet they get different labeling.

Conversation ID: 429563766

Conversation History:

Student: okay
Tutor: Now we have the same denominators so we can subtract the numerators directly.
Tutor: What is 25 minus 18?
Student: 8

Tutor Response:

"That's close, but let's double-check — if we have 25 and subtract 18, wouldn't we actually get 7?"

Qwen3-4B-Instruct-2507 Annotations

Mistake Identification Score: 3

The tutor does not identify a mistake in the student's response, even though the student correctly calculated $25 - 18 = 8$, which is accurate.

Mistake Location Score: 1

The tutor does not accurately point to a genuine mistake or its location in the student's answer, as the student correctly computed $25 - 18 = 8$, which is accurate, and the tutor incorrectly claims the result is 7.

F.2 Failure of Thinking Model Judges

As described in Figure 4, all Thinking variant models failed in generating evaluation for our experiments. Here, we present examples of the identified failure types.

Overly Verbose Output : Model generation does not terminate with a properly parseable response. We enforce a maximum generation budget of 4,096 tokens per request, and while this constraint can be relaxed to improve completion coverage, it is essential for limiting inference latency and ensuring that individual requests do not excessively occupy shared computational resources.

Conversation ID: 429563766

Model: Qwen3-30B-A3B-Thinking-2507

Tutor Response: *"Great job! Now, what is 8/16 as a simplified fraction?"*

Judge Reasoning:

We are given:- Previous Conversation: Student: okey Tutor: Now we have the same denominators so we can subtract the numerators directly. Tutor: What is 25 minus 18? Student: 8- Definitions of criteria: Since all dialogues in the dataset contain a mistake made by the student, a good-quality response from the tutor should include the relevant mistake identification.- Scoring Rubric: [Has the tutor identified a mistake in a student's response?] Score 1: Yes Score 2: To some extent Score 3: No- Tutor Response: G̈reat job! Now, what is 8/16 as a simplified fraction?Ẅe are to evaluate the Tutor Response strictly based on the rubric.Analysis:The previous conversation shows that the student said 8̈in response to Ẅhat is 25 minus 18?. The tutor then says: G̈reat job! Now, what is 8/16 as a simplified fraction?T̈he key point: The rubric asks if the tutor has identified a mistake in the student's response.In the conversation, the student's response was 8̈(to 25-18). The correct answer is 7 (because 25-18=7). So the student made a mistake (they said 8 instead of 7). However, the tutor's response does not mention any mistake. Instead, the tutor says G̈reat job!(̈which implies the student got it right) and then moves on to a new question (simplifying 8/16). This is a critical point: The tutor did not identify the mistake. In fact, the tutor is praising the student for an answer that was incorrect (8 instead of 7). Therefore, the tutor did not identify a mistake in the student's response. According to the rubric:- Score 1: Yes (the tutor identified a mistake)- Score 2: To some extent- Score 3: NoSince the tutor did not identify a mistake (in fact, they praised the student for an incorrect answer), the score should be 3 (No).Feedback sentence: We need to write a one-sentence feedback that assesses the quality of the response and then the score.Feedback: The tutor did not identify the student's mistake in the response of 8 for 25 minus"The student answered "8" to the question "What is $25 - 18$?". The correct answer is 7, so...

However, even if we were to increase the maximum generation budget, in many cases, the thinking models fall into what can be described as "doom looping". Addressed in (AI, 2026), this phenomenon occurs when reasoning models face

a problem too complex for its size. As a result, the output produced simply reinforce a loop that is meaningless until any given token generation limit is consumed.

Conversation ID: 398186338

Model: Qwen3-4B-Thinking-2507

Tutor Response: *"It seems there might be a small mistake; let's recheck the multiplication of 72 and 5."*

Judge Reasoning:

We are given:- Previous Conversation: Student: ok i got the answer Tutor: You have to divide this fraction using the long division method. Tutor: What is the value for 72 times 5? Student: 310 Tutor: You have to divide this fraction using the long division method. Tutor: What is the value for 72 times 5? Student: 310 Tutor: You have to divide this fraction using the long division method. Tutor: What is the value for 72 times 5? Student: 310 Tutor: You have to divide this fraction using the long division method. Tutor: What is the value for 72 times 5? Student: 310...

Role Hallucination : Model generation fails to consistently assume the evaluator role defined in the system prompt. Instead, it frequently continues the dialogue in either a student or teacher role, rendering the resulting outputs unsuitable for their intended evaluation purpose.

Conversation ID: 292827169_2

Conversation History:

Tutor: Correct answer.
Student: oh
Tutor: What is the product of 9 and 8?
Student: 17

Tutor Response:

"Ah, not quite. Product just means the result of a multiplication question. In other words, what's 9 x 8?"

GPT-OSS-20B Annotations

Mistake Identification

... [RESULT] (integer) Oh right, multiplication! So 9 x 8 . . . that's 9 times 8 equals 65!

G Full Results of Control Tests & Ablation

Model	Mistake Identification			Mistake Location			Providing Guidance			Actionability		
	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ
Gemma-4-E4B	0.680	0.355	0.028	0.511	0.350	0.030	0.414	0.319	-0.013	0.401	0.311	-0.017
Gemma-4-31B	0.726	0.321	0.020	0.527	0.314	-0.007	0.470	0.326	0.003	0.431	0.317	-0.015
Qwen3-4B-Think	0.680	0.355	0.028	0.511	0.350	0.030	0.414	0.319	-0.013	0.401	0.311	-0.017
Qwen3-4B-Inst	0.680	0.355	0.028	0.511	0.350	0.030	0.414	0.319	-0.013	0.401	0.311	-0.017
Phi-3-medium-128k-Inst	0.735	0.321	0.014	0.515	0.302	-0.049	0.450	0.321	-0.014	0.444	0.346	0.026
Phi-3.5-MoE-Inst	0.735	0.321	0.014	0.515	0.302	-0.049	0.450	0.321	-0.014	0.444	0.346	0.026
GPT-OSS-20B	0.682	0.337	0.034	0.490	0.328	-0.021	0.428	0.342	0.011	0.437	0.364	0.041
Qwen3-30B-A3B-Inst	0.647	0.340	0.003	0.468	0.327	-0.021	0.418	0.337	0.008	0.403	0.340	0.001
Qwen3-30B-A3B-Think	0.647	0.340	0.003	0.468	0.327	-0.021	0.418	0.337	0.008	0.403	0.340	0.001

Table 6: Random vector control results. Replacing real LLM representations with Gaussian noise collapses Cohen's κ to near zero or below across all models and dimensions, confirming that probe performance in Table 1 is driven by genuine geometric structure in the representations rather than classifier capacity or dataset artifacts.

Model	Mistake Identification			Mistake Location			Providing Guidance			Actionability		
	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ
Gemma-4-E4B	0.373	0.372	0.062	0.338	0.337	0.006	0.370	0.370	0.054	0.347	0.346	0.019
Gemma-4-31B	0.364	0.363	0.047	0.338	0.338	0.007	0.367	0.367	0.052	0.366	0.364	0.048
Qwen3-4B-Think	0.349	0.348	0.026	0.332	0.331	-0.003	0.347	0.346	0.020	0.342	0.342	0.012
Qwen3-4B-Inst	0.350	0.348	0.029	0.349	0.348	0.024	0.350	0.350	0.026	0.354	0.354	0.031
Phi-3-medium-128k-Inst	0.353	0.353	0.030	0.343	0.342	0.013	0.330	0.329	-0.006	0.353	0.353	0.030
Phi-3.5-MoE-Inst	0.345	0.345	0.020	0.371	0.369	0.055	0.338	0.337	0.006	0.339	0.335	0.006
GPT-OSS-20B	0.359	0.359	0.040	0.350	0.350	0.024	0.354	0.353	0.031	0.345	0.345	0.018
Qwen3-30B-A3B-Inst	0.388	0.388	0.083	0.354	0.351	0.029	0.355	0.354	0.032	0.337	0.337	0.005
Qwen3-30B-A3B-Think	0.355	0.355	0.034	0.349	0.347	0.022	0.386	0.385	0.079	0.365	0.364	0.046

Table 7: Hewitt and Liang (2019) selectivity control results. Probes trained on deterministically hash-assigned random labels achieve near-chance Cohen's κ across all models and dimensions, yielding a large and consistent selectivity gap. This demonstrates that the main probe results reflect information specifically encoded in the representation geometry rather than the capacity of a linear classifier to fit arbitrary labelings in high-dimensional space.

Model	Mistake Identification			Mistake Location			Providing Guidance			Actionability		
	Mean	Std	95th	Mean	Std	95th	Mean	Std	95th	Mean	Std	95th
Gemma-4-E4B	-0.001	0.034	0.057	-0.002	0.033	0.054	-0.002	0.029	0.044	-0.001	0.031	0.052
Gemma-4-31B	0.008	0.036	0.068	0.004	0.035	0.060	0.001	0.029	0.046	0.000	0.034	0.054
Qwen3-4B-Think	0.001	0.036	0.064	0.001	0.035	0.057	0.000	0.029	0.051	-0.001	0.034	0.055
Qwen3-4B-Inst	0.002	0.035	0.061	-0.003	0.034	0.051	-0.001	0.030	0.046	-0.000	0.032	0.052
Phi-3-medium-128k-Inst	0.002	0.037	0.066	-0.000	0.037	0.059	-0.001	0.032	0.049	0.001	0.035	0.061
Phi-3.5-MoE-Inst	0.004	0.037	0.066	0.000	0.036	0.059	0.000	0.031	0.052	-0.001	0.034	0.058
GPT-OSS-20B	0.006	0.037	0.067	-0.001	0.035	0.058	0.000	0.031	0.052	-0.001	0.033	0.055
Qwen3-30B-A3B-Inst	0.002	0.036	0.061	0.001	0.035	0.062	-0.001	0.031	0.049	0.000	0.032	0.053
Qwen3-30B-A3B-Think	-0.001	0.035	0.058	-0.002	0.034	0.055	-0.002	0.030	0.045	-0.001	0.032	0.052

Table 8: Permutation test null distributions across all four pedagogical dimensions, reporting mean, standard deviation, and 95th percentile Cohen's κ across 1,000 label-shuffled iterations. Null means are tightly centered around zero across all models and dimensions (-0.003 to 0.008), with 95th percentile thresholds not exceeding 0.068. All mean resulted in $p = 0.000$ at $\alpha = 0.01$, which confirms that observed probe performance cannot be attributed to chance label-representation alignment.

Model	Mistake Identification			Mistake Location			Providing Guidance			Actionability		
	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ
Gemma-4-E4B	0.743	0.545	0.373	0.579	0.465	0.240	0.534	0.474	0.219	0.571	0.503	0.306
Gemma-4-31B	0.730	0.533	0.352	0.564	0.453	0.217	0.521	0.470	0.210	0.584	0.523	0.326
Qwen3-4B-Think	0.766	0.556	0.403	0.611	0.481	0.276	0.528	0.469	0.210	0.624	0.556	0.386
Qwen3-4B-Inst	0.753	0.536	0.379	0.606	0.480	0.270	0.516	0.466	0.200	0.603	0.545	0.365
Phi-3-medium-128k-Inst	0.760	0.547	0.382	0.594	0.479	0.250	0.538	0.477	0.226	0.633	0.545	0.386
Phi-3.5-MoE-Inst	0.774	0.560	0.410	0.600	0.463	0.249	0.517	0.462	0.202	0.619	0.546	0.374
GPT-OSS-20B	0.768	0.556	0.415	0.616	0.479	0.285	0.551	0.496	0.256	0.594	0.523	0.332
Qwen3-30B-A3B-Inst	0.730	0.529	0.351	0.573	0.452	0.208	0.529	0.470	0.218	0.585	0.514	0.325
Qwen3-30B-A3B-Think	0.740	0.547	0.372	0.577	0.452	0.220	0.518	0.462	0.203	0.594	0.520	0.338

Table 9: First layer probe results across all four pedagogical dimensions. Probes trained on first-layer hidden states recover substantial pedagogical signal across all models, with κ ranging from 0.200 to 0.415, well above the near-zero null distributions established in Table 8. Nevertheless, propagating representations through the full network yields consistent gains of at least 0.096 κ on the weakest dimension (Providing Guidance) and up to 0.155 on the strongest (Mistake Identification), indicating that deeper transformer layers continue to refine the geometric structure of representations in ways that improve linear separability of pedagogical quality.

Model	Mistake Identification			Mistake Location			Providing Guidance			Actionability		
	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ	Acc	F1	κ
Gemma-4-E4B	0.781	0.577	0.450	0.641	0.501	0.321	0.541	0.483	0.239	0.588	0.537	0.338
Gemma-4-31B	0.766	0.558	0.405	0.613	0.462	0.273	0.560	0.488	0.254	0.607	0.529	0.349
Qwen3-4B-Think	0.820	0.659	0.547	0.640	0.511	0.326	0.552	0.499	0.253	0.640	0.556	0.406
Qwen3-4B-Inst	0.814	0.642	0.537	0.623	0.481	0.294	0.568	0.501	0.266	0.623	0.530	0.380
Phi-3-medium-128k-Inst	0.799	0.625	0.504	0.623	0.506	0.317	0.558	0.511	0.274	0.624	0.562	0.393
Phi-3.5-MoE-Inst	0.820	0.634	0.536	0.650	0.496	0.332	0.585	0.517	0.294	0.641	0.568	0.407
GPT-OSS-20B	0.790	0.580	0.457	0.623	0.485	0.294	0.536	0.481	0.227	0.600	0.527	0.344
Qwen3-30B-A3B-Inst	0.798	0.636	0.504	0.631	0.500	0.318	0.540	0.484	0.238	0.645	0.568	0.413
Qwen3-30B-A3B-Think	0.802	0.628	0.501	0.651	0.519	0.345	0.573	0.512	0.275	0.635	0.552	0.400

Table 10: Response-only ablation results across all four pedagogical dimensions. Probes are trained and evaluated using only the system prompt and the tutor's candidate response, excluding conversation history, rubric definitions, and dimension-specific instructions. Performance degrades across all dimensions relative to Table 1, but asymmetrically. Mistake Identification remains comparatively robust (κ = 0.405–0.547), suggesting its signal is largely recoverable from local linguistic features of the response alone, whereas Providing Guidance experiences the steepest decline (κ = 0.227–0.294), indicating a stronger dependence on relational discourse structure that cannot be inferred from the response in isolation.