NEURAL INFORMATION PROCESSING SYSTEMS

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY | 统计与数据科学系 Department of Statistics and Data Science

Beyond the Surface: Enhancing LLM-as-a-Judge Alignment with Human via Internal Representations

Peng Lai[1] Jianjie Zheng[1] Sijie Cheng[2] Yun Chen[3]
Peng Li[2] Yang Liu[2] Guanhua Chen[1*]

[1]Southern University of Science and Technology, [2]Tsinghua University,
[3]Shanghai University of Finance and Economics

Background

Large language models (LLMs) are increasingly used as *judges*—to automatically evaluate text quality, rank responses, or score outputs across tasks.

This *LLM-as-a-Judge* paradigm provides a scalable and cost-efficient alternative to human evaluation, and has become a cornerstone in recent LLM benchmarking and alignment research.

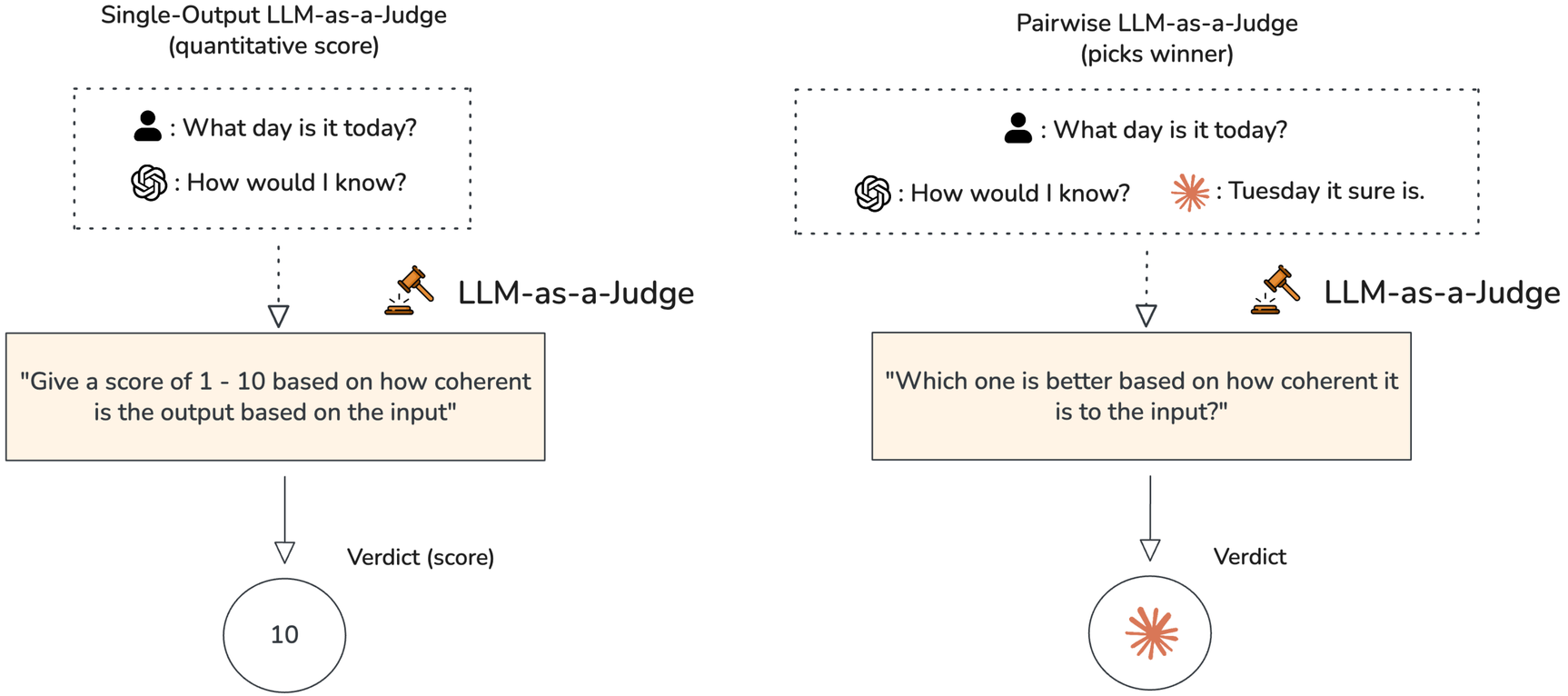


Image from LLM-as-a-Judge Simply Explained: The Complete Guide to Run LLM Evals at Scale - Confident AI

The Challenge: LLM-as-a-Judge Lacks Human Alignment

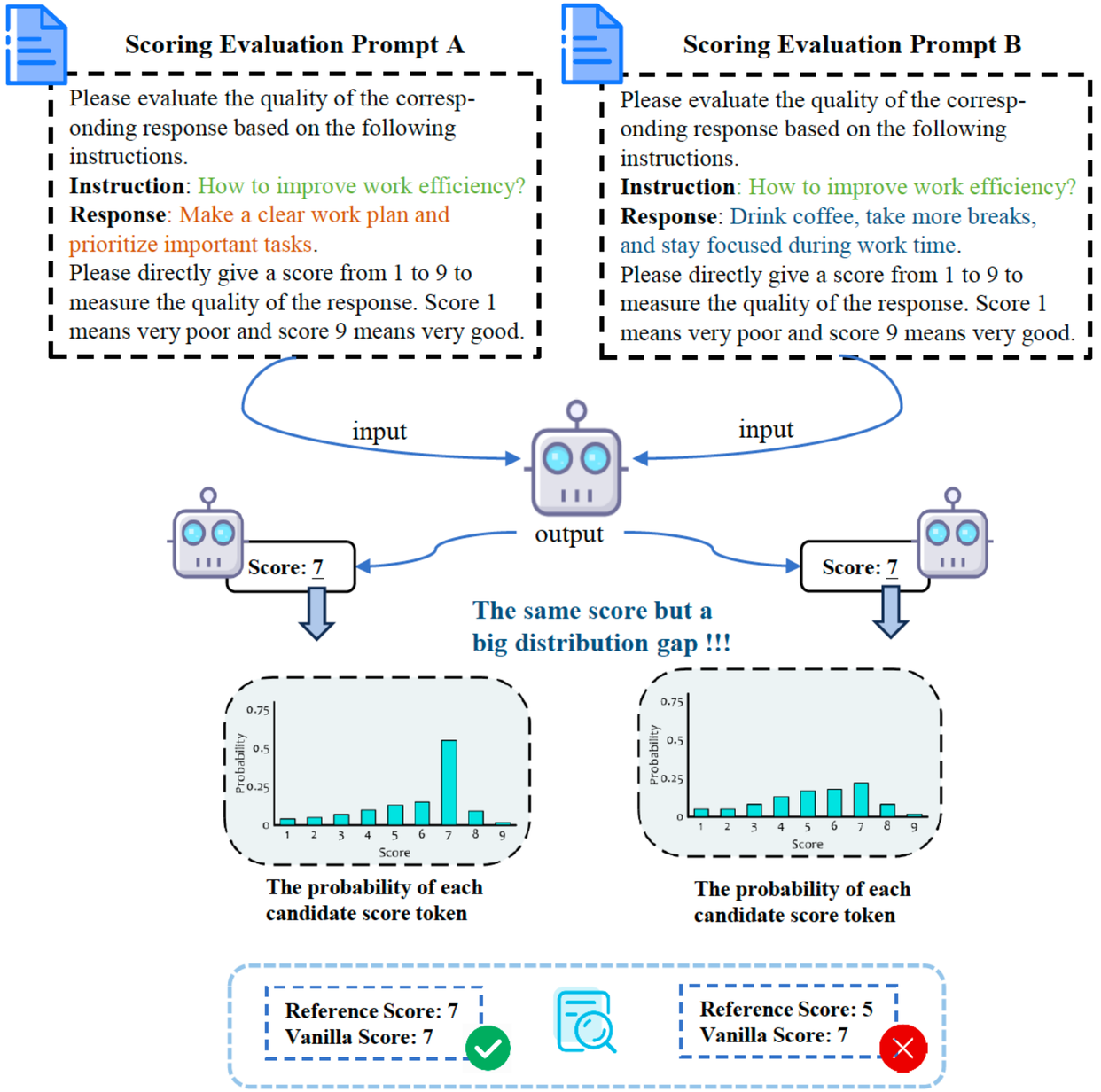
The Problem

However, despite its scalability, LLM-as-a-Judge often produces judgments that misalign with human preferences.

Existing methods to address this issue either depend solely on shallow signals (e.g., the final score token) or require costly task-specific fine-tuning, which limits generalization.

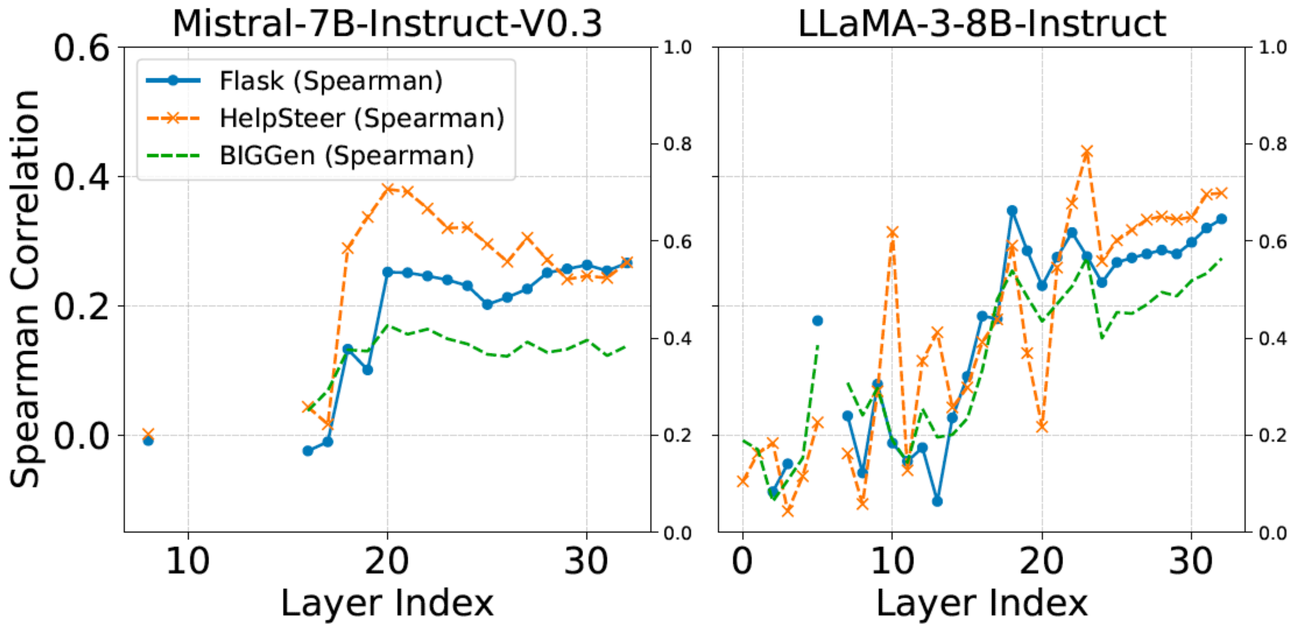
The Core Insight

Valuable semantic and task-relevant information is hidden within the model's internal layers, but it's often ignored.



Finding the Signal: Intermediate Layers Align Better

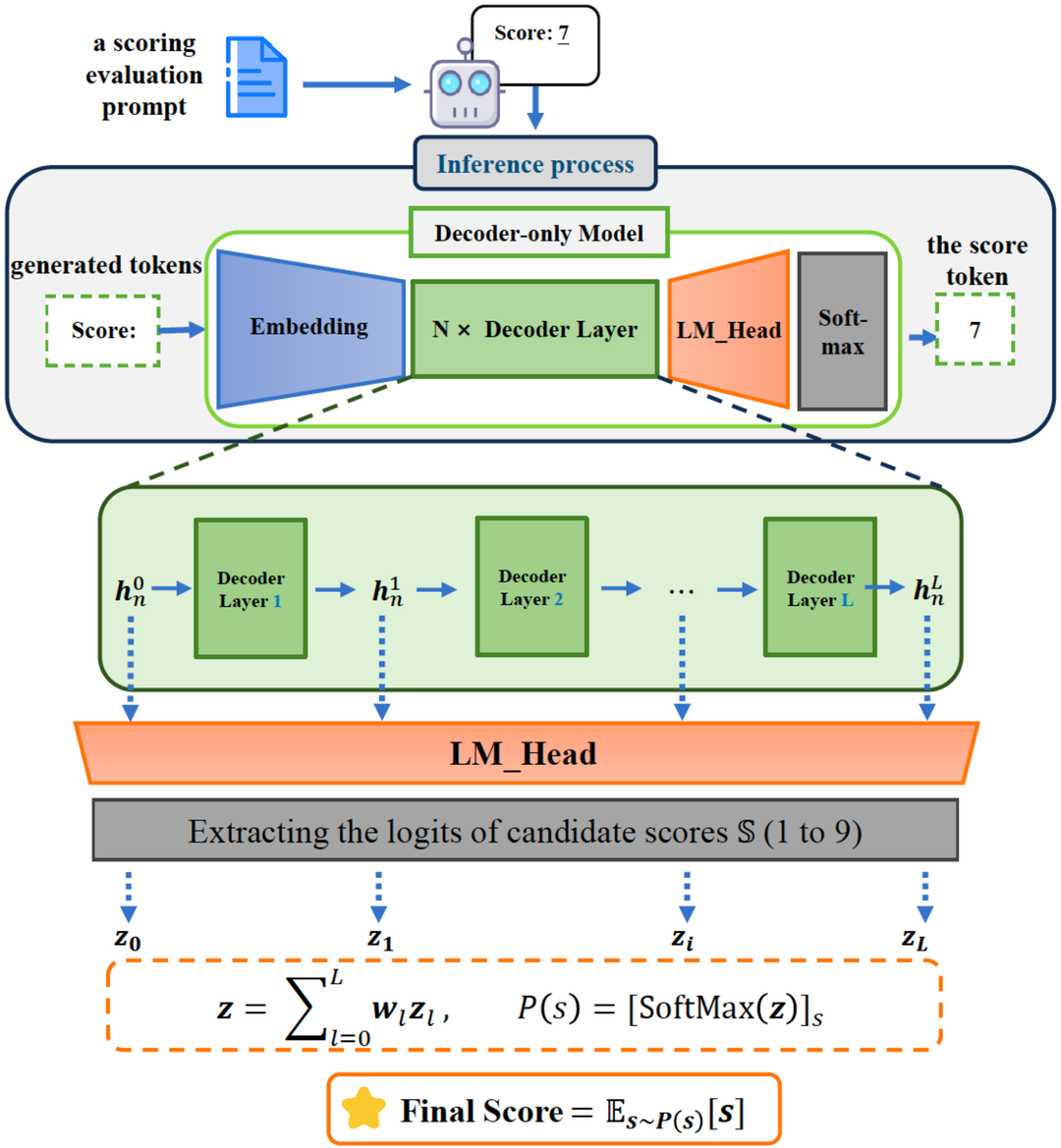
We investigated the alignment of scores derived from each layer's hidden states with human judgments. A consistent pattern emerged: middle-to-upper layers often outperform the final layer.



- Agreement Between Human Ratings and Internal Layer Scores of Different Models.

Relying only on the final layer is suboptimal, as it loses fine-grained evaluative signals during the final transformation focused on next-token prediction.

Our Solution: The LAGER Framework



Leveraging Aggregated Representations for enhancing LLM-as-a-Judge

- **Post-hoc & Plug-and-Play**: No changes to the LLM backbone.
- **Efficient**: Leverages existing internal states, no costly reasoning steps.
- **Lightweight**: An optional tuning step only adjusts a small set of layer weights.

How LAGER Works: Key Components

1. Weighted Logit Aggregation

Instead of using just the final layer, LAGER aggregates the score-token logits from multiple (or all) layers. This combines low-level lexical cues and high-level semantic signals into a single, richer representation.

$$\hat{\boldsymbol{z}}_l = (f^{(l)}_{\text{decoder}} \circ \cdots \circ f^{(1)}_{\text{decoder}} \circ f_{\text{embd}}(x_{<n}))\boldsymbol{W}_{\text{unembd}} = \mathbf{h}^{(l)}_n \boldsymbol{W}_{\text{unembd}}.$$

$$\hat{\boldsymbol{z}} = \sum_{i=0}^{L} w_i \hat{\boldsymbol{z}}_i = \sum_{i=0}^{L} w_i \mathbf{h}^{(i)}_n \boldsymbol{W}_{\text{unembd}},$$

$$\hat{\boldsymbol{z}}_{[\mathcal{M}]} = \sum_{i=0}^{L} w_i \hat{\boldsymbol{z}}_{i[\mathcal{M}]} = \sum_{i=0}^{L} w_i [\mathbf{h}^{(i)}_n \boldsymbol{W}_{\text{unembd}}]_{\mathcal{M}},$$

2. Expected Score Calculation

After aggregation, a softmax is applied to the candidate scores to create a probability distribution. The final score is the expected value of this distribution, yielding a continuous, fine-grained result that captures uncertainty.

$$P(s) = \frac{\exp(\hat{\mathbf{z}}[s])}{\sum_{s' \in \mathbb{S}} \exp(\hat{\mathbf{z}}[s'])}, \quad s \in \mathbb{S}$$

$$s^* = \mathbb{E}_{s \sim P(s)}[s] = \sum_{s \in \mathbb{S}} s \times P(s)$$

3.Lightweight Training of Layer Weights

$$\begin{aligned}\mathcal{L}_{\text{Final}} &= \alpha \cdot \mathcal{L}_{\text{CE}} + (1-\alpha) \cdot \mathcal{L}_{\text{MAE}} \\ &= \alpha \cdot \left(-\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{s=1}^{\mathbb{S}} \mathbb{I}(s = s^i_{\text{truth}}) \log P_i(s) \right) + (1-\alpha) \cdot \left(\frac{1}{2|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} (s^*_i - s^i_{\text{truth}})^2 \right)\end{aligned}$$

LAGER Achieves Superior Human Alignment Results

- Across three standard benchmarks (Flask, HelpSteer, BiGGen), LAGER consistently outperforms baseline methods, including vanilla scoring and expectation scoring, for various open-source models.

- LAGER improves alignment by up to 7.5% over the best baseline, sometimes matching or exceeding reasoning-based methods without the extra cost.

Model	Flask		HelpSteer		BIGGen Bench		Average
	Direct	Reasoning	Direct	Reasoning	Direct	Reasoning	
Fine-tuned Models							
TIGERscore-7B	-	0.175	-	0.118	-	0.171	0.155
Prometheus2-7B	-	0.413	-	0.514	-	0.367	0.431
Close-source Model via API							
GPT-4o-mini							
Vscore	0.526	0.535	0.482	0.535	0.534	0.509	0.520
E-Score	0.579	0.561	0.500	0.541	0.573	0.530	0.547
Open-source Models							
Mistral-7B-Instruct-v0.3							
GPTScore	0.258	-	0.209	-	0.183	-	0.217
Vscore	0.266	0.269	0.267	0.364	0.138	0.280	0.264
E-Score	0.239	0.279	0.296	**0.380**	0.185	0.283	0.277
LAGER (w.o tuning)	0.338	0.295	0.401	0.377	0.353	0.329	0.349
LAGER (w. tuning)	**0.347**	**0.298**	**0.403**	0.376	**0.357**	**0.333**	**0.352**
LLaMA3.1-8B-Instruct							
GPTScore	0.061	-	-0.022	-	-0.162	-	-0.041
Vscore	0.334	0.429	0.374	0.518	0.273	0.390	0.386
E-Score	0.386	0.446	0.464	**0.525**	0.352	0.403	0.429
LAGER (w.o tuning)	0.472	0.456	**0.520**	0.524	0.475	0.443	0.482
LAGER (w. tuning)	**0.477**	**0.460**	0.515	0.524	**0.482**	**0.444**	**0.484**
InternLM3-8B-Instruct							
GPTScore	-0.087	-	-0.062	-	-0.257	-	-0.135
Vscore	0.423	0.449	0.388	0.425	0.374	0.441	0.417
E-Score	0.515	0.472	0.453	0.430	0.470	0.470	0.468
LAGER (w.o tuning)	0.449	0.468	0.426	0.429	0.374	0.469	0.436
LAGER (w. tuning)	**0.545**	**0.489**	**0.515**	**0.474**	**0.507**	**0.490**	**0.501**
Qwen-2.5-14B-Instruct							
GPTScore	0.001	-	-0.014	-	-0.142	-	-0.052
Vscore	0.547	0.537	0.420	0.423	0.458	0.461	0.474
E-Score	0.579	0.555	**0.447**	0.452	0.502	0.457	0.499
LAGER (w.o tuning)	0.572	0.567	0.433	**0.473**	0.503	0.507	0.509
LAGER (w. tuning)	**0.612**	**0.572**	0.443	0.472	**0.567**	**0.524**	**0.531**
Mistral-Small-24B-Instruct							
GPTScore	0.016	-	-0.008	-	-0.147	-	-0.046
Vscore	0.528	0.505	0.420	0.459	0.542	0.533	0.498
E-Score	0.577	0.532	0.442	0.486	0.585	0.555	0.530
LAGER (w.o tuning)	0.591	**0.542**	0.452	0.485	0.589	0.562	0.537
LAGER (w. tuning)	**0.596**	**0.542**	**0.449**	**0.487**	**0.598**	**0.566**	**0.540**
LLAMA-3.3-70B-Instruct							
GPTScore	0.042	-	0.014	-	-0.193	-	-0.046
Vscore	0.518	0.567	0.435	0.494	0.559	0.539	0.519
E-Score	0.464	0.540	0.444	0.488	0.530	0.506	0.495
LAGER (w.o tuning)	0.610	0.598	**0.506**	**0.520**	0.597	**0.585**	0.569
LAGER (w. tuning)	**0.611**	**0.598**	0.504	0.519	**0.602**	0.584	**0.570**

Ablation Study

- **The full configuration (Exp. score + Logits agg. + Tuning) achieves the best performance**,
  → reaching a maximum Spearman correlation of **0.545**.

- **Fine-tuning shows significant improvement**,
  → yielding up to **+0.10** gain over the untuned version.

- **Expectation scoring outperforms maximum scoring**,
  → with improvements of up to **+0.17**.

- **Logits aggregation surpasses probability aggregation**,
  → with improvements of up to **+0.07**.

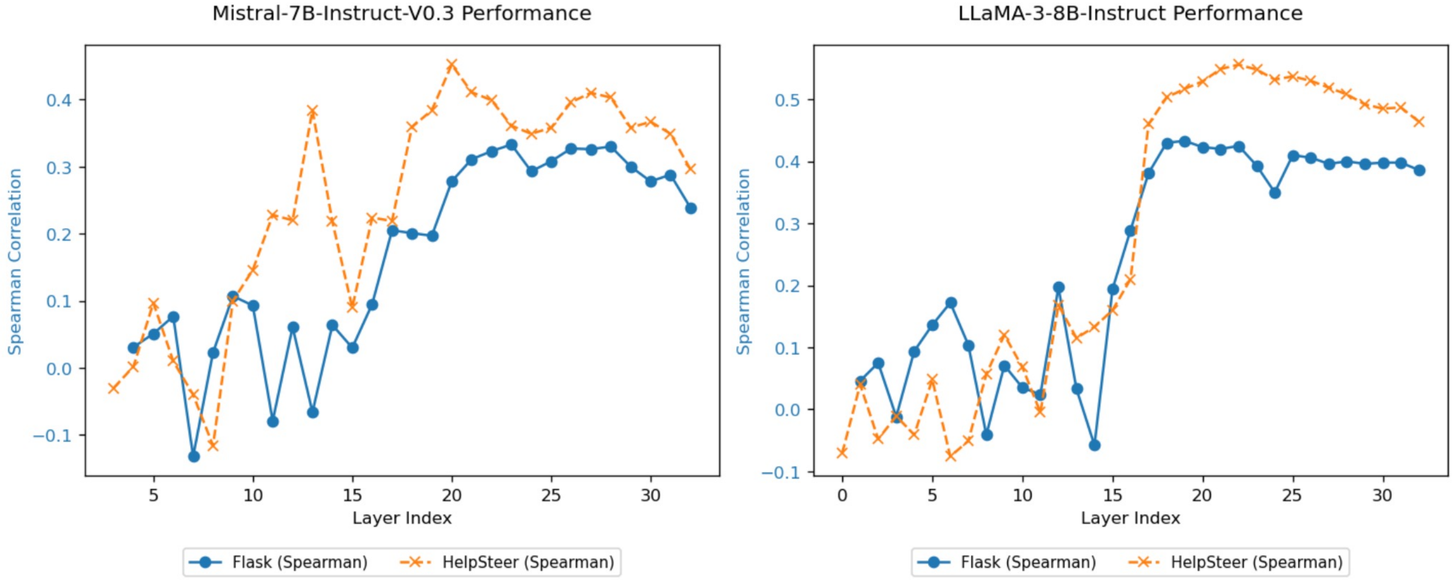
- **Multi-layer integration is generally effective**,
  → particularly for **Mistral**.

- **Model-specific differences are observed**,
  → for **InternLM**, in untuned settings, removing aggregation sometimes performs better.

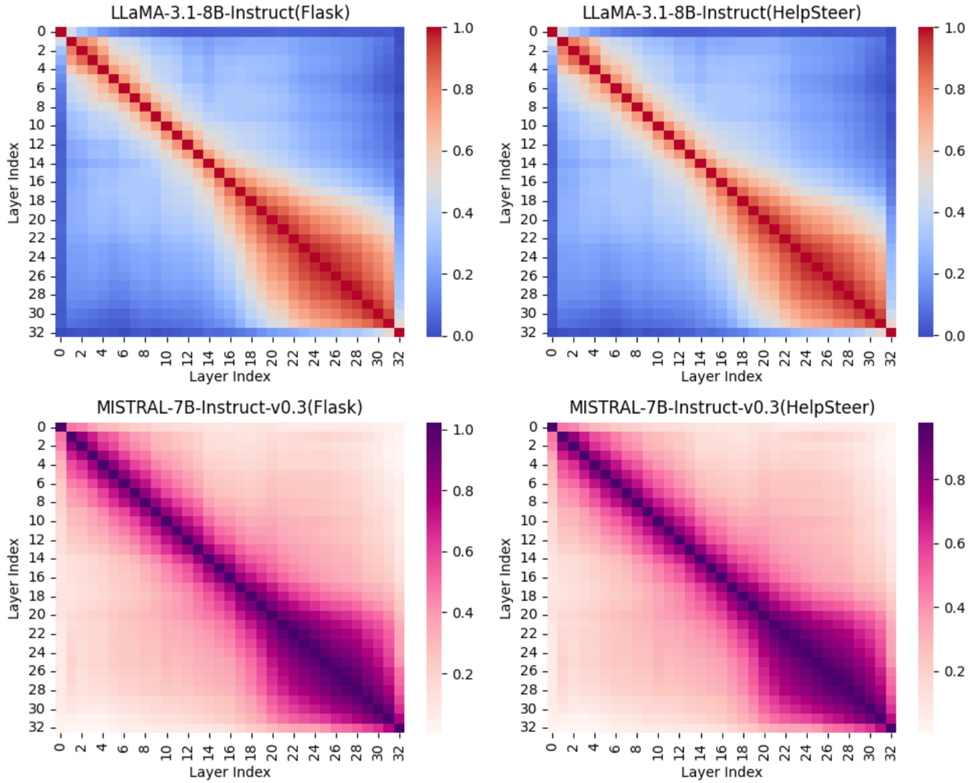
	Strategies For PalmScore					InternalLM		Mistral	
ID	Exp. score	Max. score	Logits agg.	Prob. agg.	Tuning	Flask	HelpSteer	Flask	HelpSteer
①	✓	×	✓	×	✓	**0.527**	**0.498**	**0.347**	**0.403**
②	✓	×	×	✓	✓	0.477	0.479	0.301	0.340
③	×	✓	✓	×	✓	0.443	0.394	0.269	0.264
④	×	✓	×	✓	✓	0.323	0.383	0.242	0.341
⑤	✓	×	✓	×	×	0.449	0.426	0.338	0.401
⑥	✓	×	×	✓	×	0.451	0.432	0.306	0.357
⑦	×	✓	✓	×	×	0.379	0.367	0.265	0.264
⑧	×	✓	×	✓	×	0.380	0.334	0.268	0.330
⑨	✓	×	×	×	×	0.515	0.453	0.239	0.296
⑩	×	✓	×	×	×	0.423	0.388	0.266	0.267

Analysis: Understanding Internal States

- Performance of PalmScore (w.o. tuning) with score distribution computed from different layers.



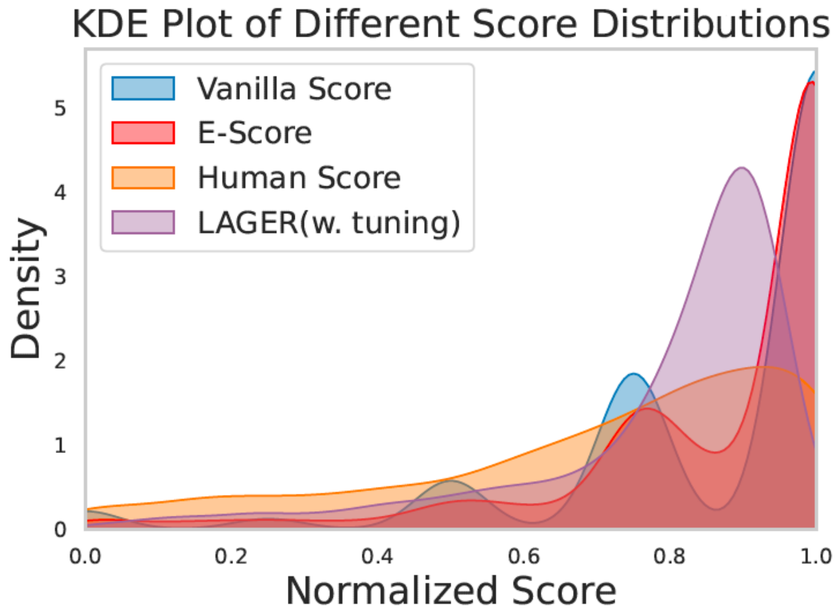
- The average cosine similarity heatmap of hidden states across different layers



1) The top layer has judgment ability but may miss fine details.

2) Middle-upper layers align well with human scores; aggregating them improves accuracy.

Analysis: Score Distribution More Closely Matches Humans



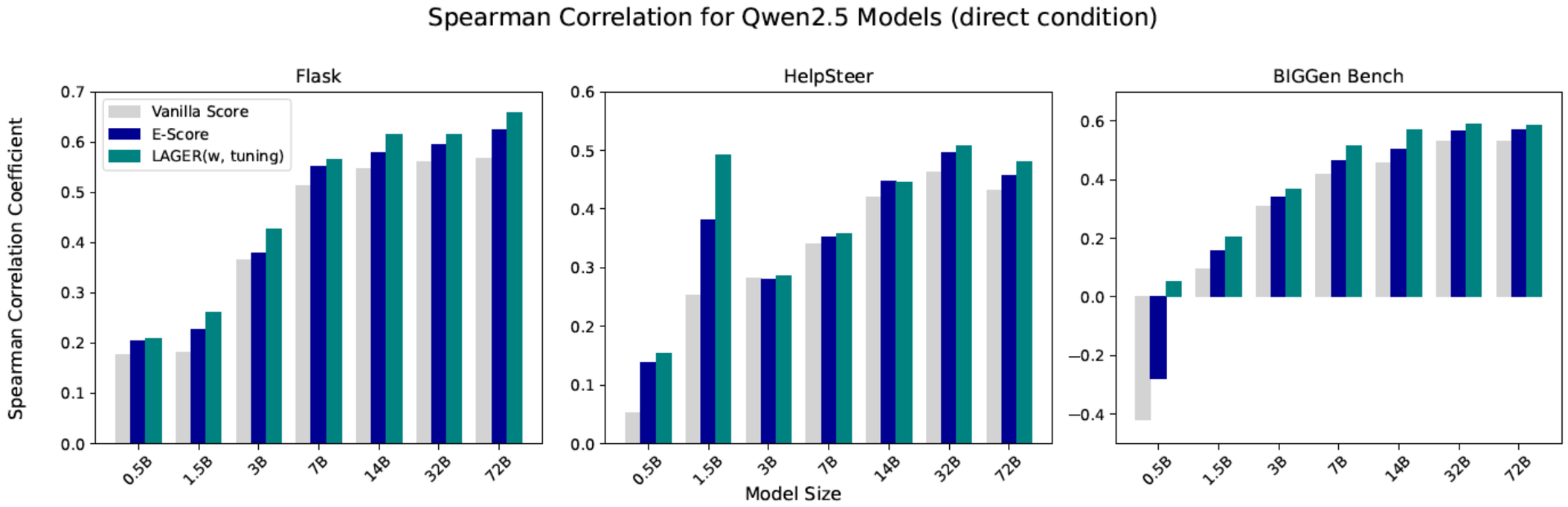
Closer to Ground Truth

LAGER's fine-grained scores produce a distribution that more accurately reflects the human score distribution, mitigating the high-score bias often seen in LLM judges.

LLaMA-3.1-8B-Instruct	$D_{KL}(\downarrow)$	MSE($\downarrow$)
VScore	0.312	0.112
E-Score	0.102	0.092
LAGER (w. tuning)	**0.087**	**0.060**

LAGER's Benefits Scale with Model Size

We tested LAGER on the Qwen2.5 model family, from 0.5B to 72B parameters. The performance gains from LAGER are not only consistent but also synergistically amplified as model capacity grows.



Conclusion & Broader Applications

Key Takeaways

- Final-layer judging is insufficient; internal layers hold rich evaluative signals.
- LAGER is a lightweight, plug-and-play framework that effectively aggregates these signals.
- It significantly improves human alignment across benchmarks and model scales.
- It provides a more robust and fine-grained evaluation without costly reasoning.

Applications

Instruction Data Selection

Using LAGER to score and filter data for SFT led to a +1.96 LC win rate over the SuperFiltering baseline on AlpacaEval 2.0.

Emotional Understanding

LAGER improves an LLM's ability to score emotional intensity, boosting correlation with human ratings by up to 9 points on EQ-Bench.

THANKS