Task-Aware Evaluation and Error-Overlap Analysis for Large Language Models

Pranava Madhyastha
City, University of London
The Alan Turing Insitute
pranava.madhyastha@city.ac.uk

Abstract

Public leaderboards for large language models often rely on aggregate scores that conceal critical information about model behaviour. In this paper, we present a methodology for task-aware evaluation that combines (i) correctness metrics aligned with task semantics compliance checks for measuring instruction-following and numeric equivalence for mathematics with (ii) pairwise error-overlap analysis for identifying complementary model pairs. We apply this methodology to 17 outputs of recent state of the art and frontier LLMs across multiple-choice QA, instruction-following, and mathematical reasoning tasks. Our analysis shows that task-aware metrics can reorder model rankings relative to generic lexical metrics, and that error-overlap patterns vary substantially across model pairs and scenarios. We finally conclude by discussing implications for model selection, routing strategies, and using LLMs in the context of judging and measuring outputs.

1 Introduction

Large language models (LLMs) are increasingly embedded in high-stakes pipelines (Tamkin et al., 2021), such as from triaging safety incidents and assessing student work (for e.g., Liu et al., 2023) to screening resumes and serving as automatic judges in evaluation (Zheng et al., 2023). While public leaderboards usually present a certain ordering of models (Liang et al., 2023; Hugging Face, 2023), real world deployments usually hinge on a set of different questions: what types of mistakes do models make, how often do models share those mistakes, and which metrics faithfully capture correctness for the task at hand? Previous research has observed that reported headline (aggregated) scores can conceal substantial error correlation across models (see for instance Kim et al., 2025), and that generic text similarity metrics are often ill-suited to instruction-following or mathematical reasoning (Zheng et al., 2023; Liang et al., 2023).

These questions have significant operational (or contextual utilisation) relevance. When models appear similar on aggregate leaderboards but diverge on specific scenarios, practitioners (or the users of the models) may need finer-grained diagnostics to inform deployment choices. Previous research has documented substantial error correlation across models, particularly on multiple-choice tasks (Kim et al., 2025), and has shown that model outputs can be more similar to each other than to human responses (Jain et al., 2025). Correlated errors have implications, especially, for effectiveness of ensembling(Chen et al., 2025), or for LLM-as-judge reliability when judges share blind spots with candidates (Zheng et al., 2023; Panickssery et al., 2024), and broader concerns about algorithmic monoculture in decision-making systems (Kleinberg and Raghavan, 2021; Bommasani et al., 2023b). In this paper, we argue that combining task-aligned correctness criteria with per-scenario error-overlap analysis can provide complementary signals for model selection and evaluation design though validating the operational impact of these methods remains an important direction for future work.

A growing body of recent research in this direction quantifies correlated errors across LLMs and their downstream effects. Kim et al. (2025) demonstrate substantial error agreement across hundreds of models on multiple-choice QA (e.g., on MMLU (Hendrycks et al., 2021) within HELM in (Liang et al., 2023)) and show that correlation increases with individual accuracy and shared lineage (provider/architecture), with notable impacts on LLM-as-judge and hiring-market simulations. **?** propose accuracy adjusted similarity metrics that treat different wrong answers as disagreement and leverage predictive distributions when available. Other works analyse algorithmic monoculture and systemic exclusion in markets (Klein-

Proceedings of the 1st Workshop on Confabulation, Hallucinations and Overgeneration in Multilingual and Practical Settings (CHOMPS 2025), pages 1–10

berg and Raghavan, 2021; Creel et al., 2022), self-preferencing in judging, and ecosystem structure including component sharing across models (Bommasani et al., 2023b). Surveys of LLM-as-judge practices document both strengths and limitations, including bias when judges share error modes with candidates (Zheng et al., 2023; Xu et al., 2025). Broadly, these studies emphasize the prevalence and consequences of the inherent correlations.

Our contribution in this work is complementary to these directions. We extend correlation analysis beyond multiple-choice into instruction-following and mathematics with examples of task-aware scoring; introduce alignment-aware, per-scenario error overlap that localizes co-failures. Specifically, we:

- propose structured correctness checks for instruction-following (compliance with constraints on format, length, and content) and mathematics (numeric equivalence with tolerance for common representations), as alternatives to lexical overlap metrics where those may be misaligned with task semantics.
- compute per-scenario pairwise error overlap under explicit alignment modes, providing a basis for identifying where models fail on the same versus different instances.
- implement robust answer extraction for multiple-choice tasks and surface per-class confusion matrices to expose distribution-specific patterns.
- demonstrate how structured checks can serve as audit tools for LLM-as-judge pipelines, complementing rather than replacing human evaluation.

We present initial evidence that these methods reveal ranking differences and error patterns not visible in aggregate scores, and discuss their potential applications in model portfolios and evaluation design. Our analysis code and per-instance outputs are made available to support replication and extension.

2 Related work

Recent work has documented that different LLMs frequently *share* their mistakes. Kim et al. (2025) measure agreement when both models err across hundreds of systems on multiple-choice (MC) benchmarks (e.g., MMLU (Hendrycks et al., 2021) within HELM (Liang et al., 2023)), showing substantial correlation that increases with individual accuracy and with shared lineage (based on provider and architectures). Complementary analyses propose accuracy-adjusted similarity metrics that treat different wrong answers as disagreement and, when available, leverage predictive distributions (**?**); others find that on creative tasks, LLM outputs are more similar to each other than human responses are to one another (Xu et al., 2025). Our work builds directly on these findings by extending correlation analysis beyond multiple-choice tasks and by introducing per-scenario overlap measurement to localize patterns of agreement and complementarity.

While using LLMs to evaluate other LLMs is appealing but, this process has been shown to introduces bias when judges share blind spots with candidates. Zheng et al. (2023) provide evidence and guidance for LLM-as-judge pipelines; subsequent surveys catalogue strengths and limitations of judges in practice (Chang et al., 2024). Empirically, judges can over-inflate models with which they share error modes, including models from the same provider or family (see more focussed discussion in Kim et al., 2025), connecting to self-preferencing concerns (Panickssery et al., 2024). In this paper, we complement this direction of work by highlighting calibration of judges with non-LLM, structured checks (compliance and numeric equivalence), potentially helping reduction in over-rewarding of plausible but wrong outputs. Our work contributes towards a practical approach for using rule-based checks to audit judge outputs, acknowledging that such checks capture only certain dimensions of correctness and should complement rather than replace human judgment.

A parallel direction of literature examines the societal and market-level implications of model homogeneity. Theoretically, algorithmic monoculture can reduce firm performance and increase systemic exclusion, wherein applicants are rejected across many decision-makers using similar systems (Kleinberg and Raghavan, 2021; Creel et al., 2022). Follow-up work analyses trade offs between individual accuracy and diversity, showing contexts where diversity can yield *wisdom-of-crowds* gains and settings where monoculture affects applicant and firm welfare (Peng and Garg, 2024a,b). Our per-scenario error-overlap analysis operationalises diversity by identifying complementary model pairs that minimise co-failures in

specific scenarios.

The inherent correlation is plausibly driven by shared components (data, architectures, training regimes). Ecosystem studies map component sharing across models, supporting a component-sharing hypothesis (Bommasani et al., 2023b,a). Such structural commonalities help explain why models converge not only in accuracy but also in error (Kim et al., 2025). Mechanistic evidence of representational homogeneity (e.g., aligned embeddings or layered activations across networks) provides further context (Lin et al., 2025).

Within-model generative diversity remains an open concern (Chang et al., 2024; Panickssery et al., 2024). Empirical studies report reduced variance relative to training corpora and limited gains from inference-time perturbations. Our focus is complementary: we study *cross-model* error similarity and how to exploit residual diversity (low-overlap pairs) for routing and ensembling.

Holistic evaluation efforts (Liang et al., 2023) and widely used benchmarks such as MMLU (Hendrycks et al., 2021) have enabled broad cross-model comparisons. However, generic lexical metrics are poorly aligned with instruction-following correctness and mathematical validity. We therefore adopt task-aware measures: compliance scoring for instruction-following (e.g., highlight counts, punctuation constraints, word limits, checklist coverage) and numeric equivalence for mathematics (fractions and square-root forms). These measures reveal ranking reversals that headline scores obscure, and they localise failure modes when combined with per-scenario error overlap.

3 Methodology

We present a methodology for task-specific evaluation and error-overlap analysis designed to complement existing benchmark scores. Our approach is motivated by the observation that generic lexical metrics (token overlap, BLEU) may not align well with the semantic requirements of specialized tasks. However, we emphasize that the correctness criteria we propose compliance checks and numeric equivalence are proxy measures that capture certain aspects of task success but do not replace human evaluation or task-specific ground truth when available. Our goal is to provide additional diagnostic signals that can inform model selection and highlight areas for deeper investigation.

Data and scope. Our analysis covers three task families with distinct correctness notions: (i) multiple-choice (MC) QA (e.g., MMLU (Hendrycks et al., 2021) within HELM (Liang et al., 2023)); (ii) instruction-following (e.g., IFEval and WildBench type prompts); and (iii) mathematical problem solving (e.g., Omni-MATH-type items). We source scenario-state JSONs from HELM benchmark output files (Liang et al., 2023), which include per-instance model completions, inputs, and, when available, reference outputs and option mappings.

Instance alignment. For cross-model error-overlap, instances must be aligned across systems. We support multiple alignment keys: (a) *scenario-instance* (scenario identifier + instance id); (b) *prompt-hash* (hash of normalised input text) for robustness to id drift; and (c) *instance-id* alone for datasets with stable identifiers. All per-instance outputs include the chosen alignment key to ensure reproducibility.

3.1 Task-aware correctness metrics

Multiple-choice (MC). We detect MC via adapter specifications or the presence of an `output_mapping`. Predicted answers are extracted using contextual patterns (e.g., "Final answer: (C)", "Option A"), falling back to isolated-letter detection, and finally to mapping by option-text mentions, with all predictions filtered to the set of valid options. Gold answers are recovered from references tagged *correct* or from the mapping. We report:

- **Accuracy**: fraction of instances where the predicted letter set equals the gold set (single-label by default).
- **Confusion matrices**: counts over gold vs. predicted letters to expose distractor-specific errors and class imbalance.
- **Macro PRF**: per-class precision/recall/F1 averaged across labels (reported only with sufficient sample size to avoid instability).

Rationale: MC tasks require robust extraction and class-sensitive diagnostics; macro PRF complements accuracy under imbalance.

Instruction-following (compliance). Generic lexical metrics (e.g., BLEU, token F1) may poorly reflect adherence to explicit constraints when reference outputs are unavailable or when the task

requires specific formatting. We therefore compute *compliance scores* from structured rules that check for: (i) punctuation constraints, (ii) format constraints, (iii) length constraints, and (iv) checklist coverage. These checks capture surface-level adherence to instructions and may serve as a complement to human judgment of overall response quality, though they do not guarantee semantic correctness or utility.

Mathematics (numeric equivalence). For problems where answers are numeric expressions, exact string matching is overly strict while general text similarity is insufficiently precise. We parse predicted and reference answers into numeric values, handling common representations (fractions, square roots), and compute equivalence within a small tolerance. This approach aims to recognize mathematically equivalent answers while remaining conservative where some valid reformulations may not be detected, leading to underestimation of correctness in cases requiring symbolic manipulation.

3.2 Formal definitions

Let $\mathcal{D}$ denote a set of aligned instances and $\mathcal{M}$ a set of models. For $i \in \mathcal{D}$, let y_i be the gold label (MC) or reference text (free-form), and $\hat{y}_i^{(m)}$ the prediction of model $m \in \mathcal{M}$. We write A_i for the alignment key.

MC accuracy and macro PRF. For single-label MC with label set $\mathcal{L}$,

$$\mathrm{Acc}(m) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \mathbf{1}\{\hat{y}_i^{(m)} = y_i\}. \tag{1}$$

From the confusion matrix, for each class $\ell \in \mathcal{L}$ with true positives TP_ℓ, false positives FP_ℓ, and false negatives FN_ℓ,

$$\mathrm{P}_\ell = \frac{\mathrm{TP}_\ell}{\mathrm{TP}_\ell + \mathrm{FP}_\ell + \epsilon}, \tag{2}$$

$$\mathrm{R}_\ell = \frac{\mathrm{TP}_\ell}{\mathrm{TP}_\ell + \mathrm{FN}_\ell + \epsilon}, \tag{3}$$

$$\mathrm{F1}_\ell = \frac{2\,\mathrm{P}_\ell\,\mathrm{R}_\ell}{\mathrm{P}_\ell + \mathrm{R}_\ell + \epsilon}, \tag{4}$$

$$\mathrm{MacroF1} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \mathrm{F1}_\ell, \tag{5}$$

using a small $\epsilon > 0$ for numerical stability when reporting.

Token overlap (free-form). Let $t(\cdot)$ tokenise text at the word level. Define corpus-level precision, recall, and F1 as

$$\mathrm{P}(m) = \frac{\sum_i |t(\hat{y}_i^{(m)}) \cap t(y_i)|}{\sum_i |t(\hat{y}_i^{(m)})|}, \tag{6}$$

$$\mathrm{R}(m) = \frac{\sum_i |t(\hat{y}_i^{(m)}) \cap t(y_i)|}{\sum_i |t(y_i)|}, \tag{7}$$

$$\mathrm{F1}(m) = \frac{2\,\mathrm{P}(m)\,\mathrm{R}(m)}{\mathrm{P}(m) + \mathrm{R}(m)}. \tag{8}$$

We report these for completeness and ablation; they are not treated as correctness for instruction-following or mathematics.

BLEU (N-gram) (based on Papineni et al., 2002). With clipped n-gram precisions p_n and uniform weights $w_n = 1/N$, the BLEU score to order N is

$$\mathrm{BLEU}_N = \mathrm{BP} \cdot \exp\Big(\sum_{n=1}^{N} w_n \log p_n\Big), \tag{9}$$

$$\mathrm{BP} = \min\big(1,\, e^{1-r/c}\big), \tag{10}$$

where c is the candidate length and r is the effective reference length.

Numeric equivalence. When both y_i and $\hat{y}_i^{(m)}$ can be parsed into reals by a normaliser $\nu(\cdot)$ supporting forms such as a/b, $k\sqrt{n}/d$, and $\sqrt{n}$,

$$\mathrm{NumMatch}_i^{(m)} = \mathbf{1}\{\,|\nu(\hat{y}_i^{(m)}) - \nu(y_i)| \le \tau\,\}, \tag{11}$$

$$\mathrm{NumRate}(m) = \frac{1}{|\mathcal{D}_\nu|} \sum_{i \in \mathcal{D}_\nu} \mathrm{NumMatch}_i^{(m)}, \tag{12}$$

with tolerance τ and $\mathcal{D}_\nu = \{i \in \mathcal{D} : \nu(y_i), \nu(\hat{y}_i^{(m)})$ exist$\}$.

Compliance rate. Given instance-level constraints $\{c_j\}$ with Boolean checks $g_j(\hat{y}_i^{(m)}) \in \{0, 1\}$ and recognised set $\mathcal{C}_i$, define

$$\mathrm{CompRate}(m) = \frac{\sum_i \sum_{j \in \mathcal{C}_i} g_j(\hat{y}_i^{(m)})}{\sum_i |\mathcal{C}_i|}. \tag{13}$$

We also report per-instance compliance $\mathrm{Comp}_i^{(m)} = \frac{\sum_{j \in \mathcal{C}_i} g_j(\hat{y}_i^{(m)})}{|\mathcal{C}_i|}$.

Error-overlap (Jaccard). Let $E_m \subseteq \{A_i : i \in \mathcal{D}\}$ be the set of alignment keys where model m errs under the relevant criterion. The pairwise Jaccard similarity is

$$J(m_1, m_2) = \frac{|E_{m_1} \cap E_{m_2}|}{|E_{m_1} \cup E_{m_2}|}, \qquad (14)$$

reported both globally and per-scenario by restricting $\mathcal{D}$.

3.3 Error-overlap and complementarity

We quantify shared failures using *pairwise Jaccard similarity* over error sets, where each error is identified by the alignment key of an instance mispredicted (for MC) or failing the task-aware criterion (for free-form when applicable). We compute global Jaccard across all scenarios and per-scenario Jaccard to localise co-failures. Low-overlap pairs are candidates for routing or ensembling, while high-overlap pairs indicate similar failure modes.

3.4 LLM-as-judge calibration

Because judges can share blind spots with candidates (Zheng et al., 2023; Kim et al., 2025), we calibrate or audit judging pipelines with structured, non-LLM checks: compliance for instruction-following and numeric equivalence for mathematics. When judges are used to grade free-form generation, we report agreement with structured checks and surface cases of plausible-but-wrong outputs receiving undue credit. This mitigates inflation from correlated errors and supports fairer cross-model comparisons.

3.5 Reporting and reproducibility

For each system we report: (i) per-instance CSVs with predictions, rationales when available, alignment keys, and task-aware metrics; (ii) per-scenario summaries including accuracy/compliance/numeric rates; (iii) MC confusion matrices; and (iv) global and per-scenario Jaccard matrices. These artefacts are intended to support downstream decisions (model selection, routing, and guardrail design) and to facilitate replication.

3.6 Scope and Design Choices

Our pipeline operates on scenario-state JSONs from HELM benchmark outputs, which include per-instance requests, completions, and when available, reference outputs and option mappings. We make the following design choices:

a) We extract predicted answers using contextual patterns (e.g., "Answer: (C)"), falling back to isolated letter detection and option-text matching. Predictions are filtered to valid options only. This approach handles most common response formats but may miss edge cases with non-standard phrasing.

b) Compliance rules are derived from instance metadata when available (constraint identifiers and arguments from IFEval-style annotations). When such metadata are absent, we report lexical metrics for reference but do not interpret them as correctness scores.

c) Our numeric parser supports common representations: plain numbers, fractions (a/b), and square roots ($k\sqrt{n}/d$, $\sqrt{n}$). We apply unicode normalization and use a small absolute tolerance ($\tau = 10^{-6}$). We do not perform general symbolic manipulation, so expressions requiring algebraic simplification may not be recognized as equivalent.

d) We compute Jaccard similarity over error sets, where errors are identified by instance alignment keys. We support scenario–instance and prompt-hash alignment; hash collisions are unlikely but theoretically possible. For free-form tasks, overlap is computed only when a binary criterion (compliance or numeric match) is defined.

f) All metrics are deterministic and rule-based; no additional LLMs are invoked during scoring. We emit per-instance CSVs and per-scenario summaries with intermediate values (alignment keys, extracted predictions) to enable independent verification.

4 Experiments

Our goal is to demonstrate the methodology in practice and provide initial evidence regarding: (i) whether task-aware metrics produce different rankings than lexical metrics, (ii) whether error-overlap patterns vary meaningfully across model pairs, and (iii) what per-scenario diagnostics reveal about model behaviour. We emphasize that our results are descriptive and exploratory establishing causal relationships or operational impact would require controlled deployment studies beyond our current scope.

4.1 Setup

We evaluate across three task families with distinct correctness notions: (i) multiple-choice (MC) QA (e.g., MMLU within HELM); (ii)

System	Parameters	Architecture	Context
GPT Family			
GPT-5	Undisclosed	MoE	400K/128K
GPT-5 Mini	Undisclosed	MoE	400K/128K
GPT-5 Nano	Undisclosed	MoE	400K/128K
GPT-OSS (120B)	117B (5.1B active)	MoE	128K
GPT-OSS (20B)	∼20B	MoE	128K
Other Frontier Models			
Grok 4	∼1.7T	MoE	256K
Kimi K2	1T (32B active)	MoE	256K
Qwen3 (235B)	235B	MoE	32K
GLM 4.5 Air	106B (12B active)	MoE	128K
Nova Premier	Undisclosed	MoE	1M
Gemini 2.5 Flash Lite	Undisclosed	Sparse MoE	1M
OLMo Family			
OLMo 2 (32B)	32B	Dense	4K
OLMo 2 (13B)	13B	Dense	4K
OLMo 2 (7B)	7B	Dense	4K
OLMoE (7B)	7B (1B active)	MoE	4K
Small Open Models			
Granite 3.3 (8B)	8B	Dense	128K
Marin (8B)	8B	Dense	4K

Table 1: Technical specifications of the 17 evaluated systems. For MoE models, active parameters per forward pass are shown in parentheses. Context shows maximum input token length (input/output when specified separately).

instruction-following (e.g., IFEval and WildBench style prompts); and (iii) mathematics (e.g., Omni-MATH-style items). Scenario-state JSON files are sourced from HELM outputs and include per-instance inputs, completions, references, and MC option mappings when applicable. We adopt the alignment and metrics defined in Section §3.

Systems. We compare a representative set of systems spanning open and closed families and capacities. We apply our methodology to 17 systems spanning multiple model families and scales, across three task types. We briefly summarise the systems in Table 1 based on the openly available details for the models[1].

Implementation. Our analyser produces per-instance CSVs, per-scenario summaries, MC confusion matrices, and pairwise error-overlap (Jaccard) matrices. For instruction-following, we evaluate compliance via structured rules (punctuation, highlights, word-count, checklist). For mathematics, we compute numeric equivalence with a tolerance τ after normalising fractions and square-root forms.

Protocol. For each task family, we report the task-appropriate correctness metric and include lexical metrics as secondary references. We compute global and per-scenario error-overlap to surface complementary pairs. Scores are aggregated over aligned instances only.

4.2 Results

4.2.1 Overall summary across models

We report MC accuracy and macro F1, compliance (IF), numeric equivalence (Math), and token F1 (free-form; secondary). Columns are organized by task family, each measuring a different aspect of model capability. MC Acc and Macro F1 capture multiple-choice performance and per-class balance; Compliance measures adherence to explicit constraints (punctuation, format, length, checklist items) as a proxy for instruction-following; Numeric Eq. measures mathematical answer correctness via numeric normalization; and Token F1 provides lexical overlap for reference. We emphasize that Compliance and Numeric Eq. are rule-based proxies that capture certain dimensions of correctness but do not substitute for human evaluation of response quality or task success.

Three patterns emerge from these results. First, task-aware metrics can produce different rankings than lexical metrics. For instance, Compliance scores range from ≈69% to ≈86% across systems, differentiating instruction-following capability even when Token F1 values are uniformly low (often below ≈5%) due to absent references or min-

[1]We refer the reader to https://crfm.stanford.edu/helm/lite/latest/ for more details and the full extent of the outputs

	Multiple-Choice		Instruction	Math	Reference
System	**Acc**	**Macro F1**	**Compliance**	**Num. Eq.**	**Token F1**
GPT Family					
GPT-5	59.7	86.5	84.5	79.4	3.7
GPT-5 Mini	57.7	83.4	81.8	70.5	3.2
GPT-5 Nano	53.9	78.2	82.3	76.7	3.3
GPT-OSS (120B)	55.0	79.5	82.7	62.9	2.0
GPT-OSS (20B)	51.2	74.1	75.8	67.1	4.0
Other Frontier Models					
Grok 4	58.9	88.9	86.2	81.8	6.1
Kimi K2	56.6	82.4	85.1	64.5	0.7
Qwen3 (235B)	57.6	84.7	86.2	63.4	0.5
GLM 4.5 Air	53.5	83.3	84.4	80.0	1.6
Nova Premier	50.2	72.6	81.8	37.5	1.6
Gemini 2.5 Flash Lite	36.3	80.0	84.5	48.9	0.4
OLMo Family					
OLMo 2 (32B)	38.2	41.4	84.0	20.7	1.9
OLMo 2 (13B)	32.4	33.3	82.9	19.5	1.8
OLMo 2 (7B)	30.6	30.8	74.6	15.6	1.6
OLMoE (7B)	23.2	20.4	69.7	13.7	3.5
Small Open Models					
Granite 3.3 (8B)	24.6	36.5	77.3	23.6	1.4
Marin (8B)	26.6	27.7	71.8	18.8	1.6

Table 2: Overall performance across 17 systems, organized by model family. Metrics are aggregated over aligned instances across all tasks. MC Acc and Macro F1 measure multiple-choice performance; Compliance measures instruction-following constraint adherence; Numeric Eq. measures mathematical correctness; Token F1 provides lexical overlap as reference. Metrics measure different aspects of capability and are not directly comparable across columns. All values are percentages.

imal lexical overlap with valid responses. Similarly, Numeric Eq. scores span ≈13% to ≈81%, and systems with similar Token F1 can differ substantially in numeric correctness. These divergences suggest that task-aligned metrics may reveal capability differences that generic lexical measures obscure, though validating whether these differences predict real-world task success remains important future work.

Second, MC Macro F1 provides a complement to accuracy by accounting for per-class precision and recall. Systems with similar MC Acc scores can show notable differences in Macro F1 (e.g., Kimi K2 at 56.6%/82.4% versus Nova Premier at 50.2%/72.6%), potentially indicating different patterns of distractor sensitivity or class imbalance handling. Whether these differences are operationally significant depends on the downstream application and class distribution.

Third, no single system dominates across all task types. Some models score highly on Compliance but lower on Numeric Equations, while others show the reverse pattern. This variation suggests that model selection might benefit from considering workload composition though implementing task-specific routing or portfolios introduces engineering complexity (infrastructure, latency, cost) beyond the scope of our current analysis.

When interpreting these results for model selection, we recommend: (i) prioritizing the metric(s) most aligned with your task requirements (Compliance for instruction-following, Numeric Equations for math tasks, MC Acc/Macro F1 for multiple-choice); (ii) treating Token F1 as contextual information rather than a correctness criterion for instruction-following or mathematics; and (iii) considering both aggregate performance and error-overlap complementarity (discussed below) as inputs to selection decisions. However, we emphasize that these metrics provide diagnostic signals rather than definitive guidance which operational deployment requires broader consideration of cost, latency, safety, and task-specific validation.

4.2.2 Error-overlap patterns.

Table 3 shows pairwise Jaccard similarity of error sets for four OLMo variants on GPQA (multiple-choice). Error overlap for the proportion of instances where both models fail ranges from ≈56% to ≈62% within this model family. These moderate overlap values suggest that even architecturally related models exhibit some diversity in their failure

	OLMo 2 (32B)	OLMo 2 (13B)	OLMo 2 (7B)	OLMoE (7B)
OLMo 2 (32B)	–	59.7	61.0	62.0
OLMo 2 (13B)	59.7	–	56.8	57.1
OLMo 2 (7B)	61.0	56.8	–	60.3
OLMoE (7B)	62.0	57.1	60.3	–

Table 3: Pairwise error overlap (Jaccard similarity, %) on GPQA (multiple-choice) among four OLMo family models. Values indicate the proportion of instances where both models fail out of all instances where at least one model fails. Lower values suggest more complementary error patterns. For brevity, we show one representative model family.

patterns, though whether this diversity translates to practical gains in ensemble or routing scenarios would require explicit validation.

We note that this analysis is limited to one model family on a single multiple-choice benchmark. Cross-family patterns and behaviour on instruction-following or mathematical tasks may differ. Moreover, error overlap is a descriptive measure of co-failure frequency as it does not establish causality (e.g., whether shared errors result from common training data, architectural similarities, or inherent task difficulty) nor does it guarantee that low-overlap pairs will yield superior ensemble performance without empirical testing.

Across our full analysis, we observe three patterns. First, task-aware metrics can reorder systems relative to lexical metrics on instruction-following and mathematics. Second, error-overlap values vary across model pairs and scenarios some pairs exhibit higher overlap (potentially indicating redundant coverage), while others show lower overlap (potentially indicating complementarity), though the operational significance of these differences remains to be validated. Third, multiple-choice confusion matrices reveal per-class error patterns that aggregate accuracy obscures, such as systematic biases toward particular distractors.

These patterns suggest that combining task-aligned metrics with instance-level error analysis may provide diagnostic signals that complement aggregate benchmark scores. However, translating these signals into deployment decisions, such as constructing model portfolios, implementing routing strategies, or calibrating ensemble methods, requires additional work and empirical validation beyond the scope of our current analysis.

4.2.3 IFEval (Instruction-Following)

We compute pairwise error overlap (Jaccard similarity) separately for each task type to examine whether complementarity patterns differ across domains. For brevity, we present 4-system subsets. Table 4 shows error overlap on IFEval, where errors are instances failing compliance checks (punctuation, format, length, checklist constraints). Overlap ranges from ≈67% to 82%, indicating substantial but incomplete co-failure among these high-performing systems.

	Grok-4	Kimi K2	Qwen3 (235B)	GPT-5
Grok-4	–	82.6	78.3	73.1
Kimi K2	82.6	–	79.2	67.9
Qwen3 (235B)	78.3	79.2	–	76.9
GPT-5	73.1	67.9	76.9	–

Table 4: IFEval error overlap (Jaccard, %). Values indicate proportion of instances where both models fail compliance checks, out of instances where at least one fails. High overlap (68-83%) suggests these systems struggle with similar constraint types.

4.2.4 Omni-MATH (Mathematics)

Table 5 shows overlap on Omni-MATH, where errors are instances failing numeric equivalence checks. Overlap ranges from 54.5% to 62.5%, lower than IFEval but more stable than WildBench. This suggests moderate complementarity: these systems share roughly half their mathematical failures while differing on the remainder.

	Grok-4	GLM 4.5 Air	GPT-5	GPT-OSS (120B)
Grok-4	–	62.5	60.6	61.8
GLM 4.5 Air	62.5	–	55.6	54.5
GPT-5	60.6	55.6	–	61.8
GPT-OSS (120B)	61.8	54.5	61.8	–

Table 5: Omni-MATH error overlap (Jaccard, %). Moderate overlap (55-63%) suggests partial complementarity on mathematical reasoning.

Overlap values differ across tasks, for e.g., IFEval shows consistently high overlap (≈68-83%), suggesting convergent failure modes on instruction-following constraints; Omni-MATH shows moderate overlap (≈55-63%), suggesting partial complementarity on mathematical reasoning. These patterns suggest that complementarity is task-dependent, i.e., model pairs that are redundant on one task type may be complementary on another.

Jaccard similarity is most reliable when both models have sufficient error samples (e.g., $\geq$10 failures each). When high-accuracy models make only 1-3 errors, overlap estimates become unstable: perfect overlap (100%) or zero overlap (0%) can occur by chance. IFEval and Omni-MATH typically have larger error sets and thus more stable estimates. Interpreting overlap for high-accuracy pairs requires caution.

5 Discussion

We have presented a methodology that combines task-aligned correctness criteria with per-scenario error overlap analysis. Our initial application suggests that: (i) task-specific metrics can reveal ranking differences not visible in generic scores, (ii) error patterns vary across model pairs and scenarios, and (iii) structured checks can serve as audit tools for LLM-as-judge pipelines.

Several important validation steps remain. First, we have not established that compliance checks or numeric equivalence correlate with human judgments of response quality, whether they are proxy measures that capture specific facets of correctness. Second, we have not tested whether low-overlap model pairs actually yield gains when combined in ensembles or routing systems. Third, our analysis is descriptive; we cannot make causal claims about why errors are shared. Fourth, our coverage is limited to three task types and 17 systems; generalization to other domains would require further study.

For practitioners, our methodology offers a complementary lens for model evaluation: task-aligned metrics may highlight capabilities that aggregate scores obscure, and error-overlap analysis may identify where models offer redundant versus complementary coverage. However, we emphasize that these tools should inform rather than dictate deployment decisions, which must account for numerous factors including cost, latency, safety requirements, and operational constraints.

Key next steps include: validating metrics against human judgments and task outcomes, testing ensemble and routing strategies informed by overlap analysis, extending coverage to additional task types and model families, and conducting deployment studies to assess operational impact. We will release our analysis code to support these efforts.

6 Conclusion

In this work, we have demonstrated some of the important limitations of evaluating large language models using aggregate scores and generic lexical metrics. We have argued that such an approach can obscure critical differences in model behaviour and fail to capture true task-specific capabilities. Our proposed methodology, which combines task-aware correctness checks with a detailed analysis of error overlap, provides a more granular and operationally relevant view of model performance. The evidence presented indicates that this approach not only re-ranks models according to criteria better aligned with task semantics but also identifies pairs of models with complementary strengths.

Limitations

Our compliance and numeric equivalence metrics are rule-based proxies for correctness. We have not validated them against human judgments or demonstrated that they predict downstream task success. They capture certain aspects of response quality (constraint adherence, mathematical accuracy) but not others (coherence, helpfulness, safety). Our evaluation covers 17 systems and three task types. Findings may not generalize to other model families, task domains, or evaluation setups. We have not performed statistical significance testing; observed differences could reflect sampling variation.

Moreover, we have not tested whether our methods improve real-world outcomes. Claims about routing, ensembling, or judge calibration are based on analysis of evaluation data, not deployment experience. Implementing such strategies introduces engineering challenges we do not address. Our error-overlap analysis is descriptive. We cannot determine whether shared errors result from common training data, architectural similarities, or task difficulty. Low overlap does not guarantee ensemble gains; high overlap does not prove causal dependence.

Some implementation details (tolerance values, parsing heuristics) were tuned based on observed data characteristics. Results may be sensitive to these choices. We will provide code and per-instance outputs to support investigation of robustness.

References

Rishi Bommasani, Kevin Klyman, Shayne Longpre, Sayash Kapoor, Nestor Maslej, Betty Xiong, Daniel Zhang, and Percy Liang. 2023a. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*.

Rishi Bommasani, Dilara Soylu, Thomas I Liao, Kathleen A Creel, and Percy Liang. 2023b. Ecosystem graphs: The social footprint of foundation models. *arXiv preprint arXiv:2303.15772*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Ming Li, Likang Xiao, Dingqi Yang, and 1 others. 2025. Harnessing multiple large language models: A survey on llm ensemble. *arXiv preprint arXiv:2502.18036*.

Kathleen A Creel, Deborah Hellman, and Deirdre K Mulligan. 2022. Algorithmic monoculture and systemic exclusion. In *FAccT*, pages 308–318.

Dan Hendrycks, Collin Burns, Steven Basart, and et al. 2021. Measuring massive multitask language understanding. *International Conference on Learning Representations (ICLR)*.

Hugging Face. 2023. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Shomik Jain, Jack Lanchantin, Maximilian Nickel, Karen Ullrich, Ashia Wilson, and Jamelle Watson-Daniels. 2025. Llm output homogenization is task dependent. *arXiv preprint arXiv:2509.21267*.

Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. 2025. Correlated errors in large language models. *International Conference on Machine Learning (ICML)*.

Jon Kleinberg and Manish Raghavan. 2021. Algorithmic monoculture and social welfare. *Proceedings of the National Academy of Sciences*, 118(22):e2018340118.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification, Outstanding Certification.

Zihao Lin, Samyadeep Basu, Mohammad Beigi, Varun Manjunatha, Ryan A Rossi, Zichao Wang, Yufan Zhou, Sriram Balasubramanian, Arman Zarei, Keivan Rezaei, and 1 others. 2025. A survey on mechanistic interpretability for multi-modal foundation models. *arXiv preprint arXiv:2502.17516*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Arjun Panickssery, Samuel Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kenny Peng and Nikhil Garg. 2024a. Monoculture in matching markets. *Advances in Neural Information Processing Systems*, 37:81959–81991.

Kenny Peng and Nikhil Garg. 2024b. Wisdom and foolishness of noisy matching markets. *arXiv preprint arXiv:2402.16771*.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*.

Jiannan Xu, Gujie Li, and Jane Yi Jiang. 2025. Ai self-preferencing in algorithmic hiring: Empirical evidence and insights. *arXiv preprint arXiv:2509.00462*.

Lianmin Zheng, Wei-Lin Chiang, Yingqi Sheng, and et al. 2023. Judging llm-as-a-judge. In *NeurIPS*.