STRESSEVAL: Failure-Driven Dynamic Benchmarking for Knowledge-Intensive Reasoning in Large Language Models

Yongrui Chen[1], **Yangyang Ma**[1], **Xiaoying Huang**[4], **Shenyu Zhang**[1], **Huajun Chen**[2], **Haofen Wang**[3] and **Guilin Qi**[1,*]
[1]Southeast University
[2]Zhejiang University
[3]Tongji University
[4]Hangzhou Dianzi University
{yongruichen, gqi}@seu.edu.cn.com

Abstract

Static benchmarks for LLMs are increasingly compromised by contamination and overfitting, especially on knowledge-intensive reasoning tasks. While recent dynamic benchmarks can alleviate staleness, they often increase difficulty at the expense of answerability and controllability. In this paper, we propose STRESSEVAL, a failure-driven data synthesis framework that turns observed model failures into dynamic, challenging, and controlable test instances. STRESSEVAL consists of three stages: (i) it constructs a semi-structured difficulty card that identifies the failed reasoning step and its root cause; (ii) it applies a dual-perspective instance-synthesis method that targets both knowledge gaps and reasoning breakdowns while preserving the underlying difficulty factors; and (iii) it applies a gating mechanism to retain only grounded, unambiguous instances. Seeding from multiple knowledge-intensive reasoning datasets, we employ STRESSEVAL to build DYNAMIC-ONEEVAL, a focused suite of challenging dynamic benchmark. Across several state-of-the-art LLMs, DYNAMIC-ONEEVAL yields substantially larger performance drops than the original benchmarks while retaining explicit difficulty factors, enabling more actionable iteration.

1 Introduction

Large language models (LLMs) have advanced rapidly, delivering strong performance across a broad range of language understanding tasks and increasingly sophisticated forms of reasoning [Brown *et al.*, 2020; OpenAI, 2023]. However, as model architectures and training pipelines evolve at pace, conventional static evaluation paradigms [Chen, 2021; Hendrycks *et al.*, 2021; Yue *et al.*, 2024] are becoming less informative. Leaderboard gains can obscure brittle behaviors, while widespread benchmark reuse and dataset contamination make it difficult to disentangle genuine generalization from memorization or overfitting [Lin *et al.*, 2024b; Wang *et al.*, 2025; Kandpal *et al.*, 2024; Anwar *et al.*, 2024]. These limitations are especially pronounced for knowledge-intensive reasoning like ONEEVAL [Chen *et al.*, 2025b], where success hinges on up-to-date factual knowledge, careful grounding, and reliable multi-step inference [Jiang *et al.*, 2024; Zhou *et al.*, 2025].

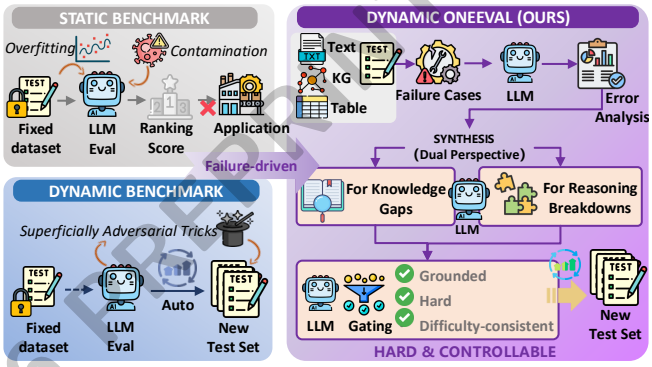

Figure 1: Upper left: static benchmarks degrade via overfitting and contamination. Bottom left: dynamic benchmarks refresh via generation but drift toward superficial trickiness. Right: STRESSEVAL turns observed failures into hard and controllable test instances.

Recent work [White *et al.*, 2024; Zhao *et al.*, 2025] has begun to explore *dynamic benchmarks* that leverage automatic data synthesis to keep evaluations continually refreshed and more resistant to benchmark contamination [White *et al.*, 2024; Li *et al.*, 2024]. While this direction can mitigate staleness, automatically generated instances often drift toward ungrounded content or superficially adversarial tricks, rather than isolating a well-defined capability or failure mode [White *et al.*, 2024; Li *et al.*, 2024; Wang *et al.*, 2025] (see the bottom-left of Figure 1). Consequently, the revealed errors are difficult to interpret and even harder to translate into actionable mitigations for model designers. This raises an open problem: how to generate evaluation instances that are both (i) *challenging*: they reliably stress specific knowledge points or reasoning patterns where models are weak, and (ii) *controllable difficulty*: their construction is governed by explicit factors so that failures can be traced back to concrete causes and directly guide model or system improvements.

*Corresponding author

To address this gap, we propose STRESSEVAL, a *failure-driven* data synthesis framework. Rather than generating difficulty from scratch, STRESSEVAL uses a three-stage pipeline that turns model's failure cases into follow-up instances that remain answerable and controlable, while systematically stressing the same underlying weakness (see the right of Figure 1). First, **Structured Error Analysis** takes instances where a model produces incorrect responses as input and employs an LLM-based analyzer to reconstruct the model's reasoning trajectory, pinpoint the bottleneck step, and diagnose the underlying root cause. It outputs a semi-structured *difficulty card* that records both the failing reasoning step and the triggering input property. Second, guided by the difficulty card, **Dual-Perspective Instance Synthesis** leverages LLMs to generate new challenges from two complementary perspectives: for *knowledge stress*, it freezes the original knowledge source, canonicalizes the missing fact as an atomic *knowledge black box*, and synthesizes dependency-aware questions and answers that still hinge on that missing fact; for *reasoning stress*, it synthesizes a fictitious but internally consistent virtual knowledge source with synthetic entities and then generates new question-answer pairs via a reasoning-skeleton procedure that inherits the same bottleneck and trigger while keeping the answer uniquely grounded. Third, a **Multi-criteria Gating** mechanism applies two LLM-based screening schemes, retaining only grounded instances that preserve the targeted difficulty card. We seed STRESSEVAL with multiple knowledge-intensive reasoning datasets and use it to produce a focused suite of high-difficulty evaluation items, STRESSEVAL-BENCH. Compared to the seed benchmarks, STRESSEVAL-BENCH yields substantially larger performance degradations across several state-of-the-art LLMs, while preserving explicit grounding and controllable difficulty factors. In summary, our contributions include:

- We propose STRESSEVAL, the first failure-driven framework for benchmarking knowledge-intensive reasoning that systematically converts observed model failures into new, difficulty-controllable test instances.
- We propose a complementary dual-perspective instance-synthesis method that systematically targets LLMs' knowledge gaps and reasoning breakdowns, enabling faithful reconstruction of specific difficulty factors.
- Comprehensive experiments on our released dynamic DYNAMIC-ONEEVAL demonstrate its difficulty and quality, and reveal that even state-of-the-art LLMs exhibit pronounced weaknesses in knowledge-intensive reasoning as well as fine-grained failure modes.

2 Problem Formulation

Assume a seed dataset $\Omega = \{(\mathcal{Q}_i, \mathcal{S}_i, \mathcal{A}_i)\}_{i=1}^{N}$, where $\mathcal{Q}_i$ is a question, $\mathcal{S}_i$ is its associated knowledge source (e.g., unstructured documents, semi-structured tables, or structured knowledge graphs), and $\mathcal{A}_i$ is the gold answer. We evaluate an LLM $\mathcal{M}$ on Ω and collect the instances on which it produces incorrect predictions. This yields a set of failure cases $\mathcal{E} = \{(\mathcal{Q}_i, \mathcal{S}_i, \mathcal{A}_i, \mathcal{Y}_i)\}_{i=1}^{M}$, where $\mathcal{Y}_i$ denotes the model output and M is the number of errors. Crucially, the source context $\mathcal{S}$ may be incomplete, i.e., it may not contain all information required to answer $\mathcal{Q}$, reflecting realistic retrieval-augmented generation settings in which models must reason under partial evidence.

Our objective is to leverage $\mathcal{E}$ to automatically construct a new dataset $\Omega^* = \{(\mathcal{Q}_k^*, \mathcal{S}_k^*, \mathcal{A}_k^*)\}_{k=1}^{K}$, where each synthesized instance is derived from one failure cases and explicitly preserves the key difficulty factors that caused $\mathcal{M}$ to fail (e.g., missing or partial fact in $\mathcal{S}$, multi-hop reasoning, compositional constraints, or distractors). The resulting dataset Ω' is designed to be challenging for the model $\mathcal{M}$.

3 STRESSEVAL

Figure 2 shows the overview of our proposed STRESSEVAL framework, which consists of: (i) **Structured Error Analysis** localizes the broken reasoning step and attributes the failure to a root cause; (ii) **Dual-Perspective Instance Synthesis** generates new challenging instances from the perspectives of *knowledge stress* and *reasoning stress*; and (iii) **Multi-criterion Gating** uses two LLM reviewers to enforce quality controls, retaining only instances that are well-grounded, and that instantiate the intended difficulty.

3.1 Structured Error Analysis

For each failure case $e_i = (\mathcal{Q}_i, \mathcal{S}_i, \mathcal{A}_i, \mathcal{Y}_i) \in \mathcal{E}$, we use an LLM as an analyzer to generate a structured error report. Concretely, let f_{ana} denote a reasoning LLM and P_{ana} be an analysis prompt template. We obtain

$$(\gamma_i, \Psi_i) \sim p_{f_{\text{ana}}}(\cdot \mid P_{\text{ana}}(\mathcal{Q}_i, \mathcal{S}_i, \mathcal{A}_i, \mathcal{Y}_i)), \tag{1}$$

where $\gamma_i \in \Gamma$ is a root-cause label that characterizes the error type with a concise, precise phrase (e.g., `Entity Linking Confusion`), and Ψ_i is our *difficulty card* that summarizes the primary challenge underlying the failure.

Difficulty card. Our defined Ψ_i aims to make each $e_i \in \mathcal{E}$ both interpretable (clarifying *why* it is difficult) and controllable (specifying *how* to preserve the same difficulty). It comprises the following fixed slots: (i) *Bottleneck Step:* the reasoning stage where the model fails (e.g., entity/relation recognition or comparison). For example, in the lower-left corner of Figure 2, for the question *Which plan is cheaper per month?*, the model breaks at *unit-time normalization* due to misinterpreting the *term* field in the table. (ii) *Trigger:* the input property, described in natural language, that triggers the failure. In the running example, *mixed billing terms* (monthly vs. yearly) create a unit mismatch that the model does not reconcile, causing it to compare prices without converting them to a common time basis (e.g., per month).

3.2 Dual-Perspective Instance Synthesis

Unlike existing methods [Lin *et al.*, 2024a; Zhao *et al.*, 2025; Wang *et al.*, 2025], we observe that failure-driven instance synthesis is prone to *root-cause entanglement*: without an explicit separation of failure sources, a synthesized instance can inadvertently mix missing-fact and reasoning breakdowns, causing the stress type to be mislabeled or the difficulty to drift (e.g., becoming trivial or unanswerable). This reduces controllability and weakens coverage of failure modes.

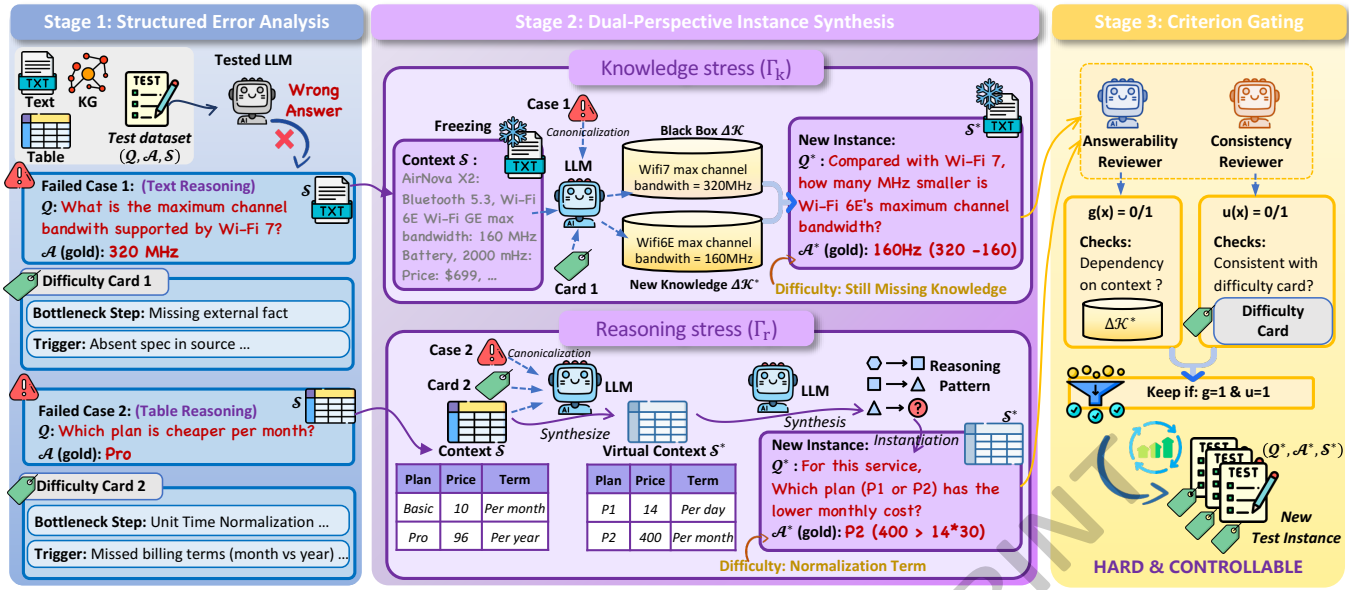

Figure 2: StressEval framwork. Stage 1 performs structured error analysis and produces per-case difficulty cards; Stage 2 synthesizes new instances via knowledge black-box for Γ_k and reasoning-skeleton for Γ_r, showing original and synthesized question-answer pairs; Stage 3 applies a LLM gating mechanism to keep answerable, unambiguous, yet unsolved hard instances.

To address this, we partition the observed failure cases $\mathcal{E}$ into two complementary perspectives according to the root-cause label γ_i and difficulty card:

- **Knowledge-stress** (Γ_k) failures: instances in which answering correctly requires information that is absent from the provided $\mathcal{S}_i$ and is also not recoverable from the model's parametric memory, resulting in errors driven by missing fact (i.e., Case 1 in Figure 2).
- **Reasoning-stress** (Γ_r) failures: instances in which the provided knowledge source $\mathcal{S}_i$ contains sufficient information to answer correctly, yet the model still produces an incorrect response due to misreading the evidence or flawed reasoning (i.e., Case 2 in Figure 2).

Below, we will introduce how to synthesize new, high-quality, and challenging test instances from each perspective.

Knowledge-stress Instance Synthesis

This branch aims to preserve the knowledge stress by *freezing the original knowledge context* $\mathcal{S}_i$. Intuitively, since the model answered question $\mathcal{Q}_i$ incorrectly due to missing fact, it is reasonable to expect that it will also answer incorrectly any questions derived from $\mathcal{Q}_i$. Specifically, for any failure case e_i with $\gamma_i \in \Gamma_k$, we keep: $\mathcal{S}_i^* := \mathcal{S}_i$.

Knowledge black box. For knowledge-stress failures, the missing fact is external to the provided source $\mathcal{S}_i$, so a faithful step-by-step decomposition is often underspecified: the model fails not because of an explicit reasoning error over $\mathcal{S}_i$, but because a crucial fact is unavailable. To preserve this *knowledge gap* while enabling controlled synthesis, we collapse the original question–answer pair into a self-contained atomic knowledge statement (like *black box*), denoted as K_0, and treat it as an indivisible unit that can be re-queried or rephrased without altering the underlying missing fact:

$$K_0 \sim p_{\mathcal{L}_{\text{can}}}(\cdot \mid P_{\text{can}}(\mathcal{Q}_i, \mathcal{A}_i)), \tag{2}$$

where $\mathcal{L}_{\text{can}}$ is an reasoning LLM used for canonicalization (with prompt template P_{can}), and P_{can} is a prompt template that converts the pair $(\mathcal{Q}_i, \mathcal{A}_i)$ into a declarative, self-contained knowledge statement. For instance, from $\mathcal{Q}_i$: *What is the maximum channel bandwidth supported by Wi-Fi 7?* and $\mathcal{A}_i$: *320 MHz*, we obtain canonical K_i: *The maximum channel bandwidth supported by Wi-Fi 7 is 320 MHz.*

Dependency-aware QA synthesis. Then, we extract a new, definitive piece of knowledge ΔK from $\mathcal{S}_i$ (i.e., directly supported by $\mathcal{S}_i$ and thus reliable), and combine it with the knowledge black box K_0 to synthesize a new question–answer pair $(\mathcal{Q}_i^*, \mathcal{A}_i^*)$. The resulting instance remains faithful to the fixed source $\mathcal{S}_i^*$, while ensuring that correctly solving it still hinges on first resolving the original $\mathcal{Q}_i$.

$$\Delta K \sim p_{f_{\text{ext}}}(\cdot \mid P_{\text{ext}}(\mathcal{S}_i^*, \Psi_i)), \tag{3}$$

$$(\mathcal{Q}_i^*, \mathcal{A}_i^*) \sim p_{f_{\text{know}}}(\cdot \mid P_{\text{know}}(\mathcal{S}_i^*, \Psi_i, K_0, \Delta K)), \tag{4}$$

where f_{ext} is an LLM-based knowledge extractor (with prompt template P_{ext}) that identifies a context-grounded fact ΔK from $\mathcal{S}_i^*$, and f_{know} (with template P_{know}) generates a new pair $(\mathcal{Q}_i^*, \mathcal{A}_i^*)$ conditioned on $\mathcal{S}_i$, the difficulty card Ψ_i, the missing-fact K_0, and the extracted fact ΔK.

Reasoning-stress Instance Synthesis

This branch aims to reproduce the same *reasoning trap* in new instances that are fully answerable from the provided context. For any failure case e_i with $\gamma_i \in \Gamma_r$, we create a *virtual* knowledge source with fictitious entities to prevent the model from exploiting parametric memory.

Virtual knowledge source synthesis. We first employ an LLM to sample a synthetic universe

$$\mathcal{U}_i = (\mathbf{E}, \mathbf{R}, \mathbf{V}) \sim p_{f_U}(\cdot \mid P_U(\mathcal{Q}_i, \mathcal{S}_i, \mathcal{A}_i, \Psi_i)), \quad (5)$$

where $\mathbf{E}$ is a set of fictitious entity identifiers, $\mathbf{R}$ defines a relation/field schema, and $\mathbf{V}$ assigns attribute values (e.g., numbers, timestamps, categories) along with unit and formatting conventions. We then render $\mathcal{U}$ into a concrete but virtual knowledge source $\mathcal{S}_i^*$ based on the original $\mathcal{S}_i$:

$$\mathcal{S}_i^* \sim p_{f_S}(\cdot \mid P_S(\mathcal{U}_i, \mathcal{Q}_i, \mathcal{S}_i, \mathcal{A}_i, \Psi_i)). \quad (6)$$

We enforce a *synthetic-entity policy*: all named entities are sampled from a reserved namespace (e.g., `P1` and `P2` in Figure 2) to minimize overlap with real-world facts.

QA pair synthesis with pattern inheritance. Given the virtual $\mathcal{S}_i^r$ and the difficulty card Ψ_i, we synthesize a new question-answer pair while explicitly *inheriting* the original reasoning bottleneck and trigger. Here, we use a two-step LLM procedure: (i) generate a *reasoning skeleton* that operationalizes Ψ_i into a minimal sequence of required reasoning operations and a concrete *trap injection* (e.g., an ambiguity/distractor that targets the bottleneck step), and (ii) render the skeleton into a natural-language question whose gold answer is uniquely derivable from $\mathcal{S}_i^*$.

Rather than asking an LLM to directly generate a new question, we first generate an explicit *reasoning skeleton* $\mathcal{G}_i$ that operationalizes Ψ_i into minimal, checkable components: required evidence fields/records in $\mathcal{S}_i^*$, the bottleneck operation to be executed (as stated in Ψ_i), a *trap injection* consistent with the trigger (a distractor that yields a plausible wrong answer if the bottleneck is skipped), and a short gold reasoning trace that points to concrete evidence. Formally,

$$\mathcal{G}_i \sim p_{f_{\text{skel}}}(\cdot \mid P_{\text{skel}}(\mathcal{S}_i^*, \Psi_i)), \quad (7)$$

$$(\mathcal{Q}_i^*, \mathcal{A}_i^*) \sim p_{f_{\text{reason}}}(\cdot \mid P_{\text{reason}}(\mathcal{S}_i^*, \Psi_i, \mathcal{G}_i)). \quad (8)$$

We constrain the LLM to (a) mention only entities/fields appearing in $\mathcal{S}_i^*$, (b) include at least one distractor consistent with the trigger in Ψ_i such that skipping the bottleneck step leads to a plausible but incorrect answer, and (c) ensure $\mathcal{A}_i^*$ is supported by an explicit reasoning trace grounded in $\mathcal{S}_i^*$.

3.3 Multi-criterion Gating

In this stage, we perform *multi-criterion gating* to control the quality of synthesized instances. For each candidate instance $x_i = (\mathcal{S}_i, \mathcal{Q}_i, \mathcal{A}_i, \Psi_i)$, we query two LLM-based reviewers, each prompted to return a structured verdict together with a calibrated confidence score. Specifically, we use: (1) an *Answerability* reviewer, which verifies that the question is answerable under the intended stress type and that the gold answer is properly grounded—i.e., supported by explicit spans/records in $\mathcal{S}_i$ for reasoning-stress, or demonstrably requiring a missing external fact beyond $\mathcal{S}_i$ for knowledge-stress; and (2) a *Consistency* reviewer, which attempts to solve the instance, checks internal consistency among $\mathcal{S}_i, \mathcal{Q}_i, \mathcal{A}_i$, and verifies that the difficulty card Ψ_i is actually instantiated rather than merely stated.

Each reviewer outputs a binary decision with confidence. We then aggregate their outputs into two binary gating signals: $g_i \in \{0, 1\}$ indicating answerability, and $u_i \in \{0, 1\}$ indicating consistency, $\text{keep}(x_i) = \mathbb{I}[g_i = 1] \cdot \mathbb{I}[u_i = 1]$, Notably, for knowledge-stress instances, grounding instead verifies the *dependency* on the missing-fact black box (i.e., solving $\mathcal{Q}_i$ necessarily requires resolving ΔK in addition to any context-grounded fact), and this dependency check is incorporated into g_i.

4 DYNAMIC-ONEEVAL

4.1 Seed Data & Implementation Details

To initialize DYNAMIC-ONEEVAL, we draw seed instances from widely used benchmarks. **a) Text Reasoning:** *HotpotQA* [Yang *et al.*, 2018] is a Wikipedia-based multi-hop QA benchmark that requires combining evidence across multiple documents to answer. **b) KG Reasoning:** *ComplexWebQ* [Talmor and Berant, 2018] is a compositional KBQA benchmark designed to answer complex questions by executing multi-relation queries over a knowledge graph. **c) Table Reasoning:** *WikiTableQuestion* (WTQ) [Pasupat and Liang, 2015] *TabFact* is a table-based fact verification dataset that asks whether a natural-language statement is entailed or contradicted by a given Wikipedia table.

We generated seed failure cases $\mathcal{E}$ by prompting GPT-5.2 [OpenAI, 2025], a strong LLM , to produce *chain-of-thought* [Wei *et al.*, 2022] answers and then verifying its outputs to collect instances where the model erred. For end-to-end quality control, the strong GPT-5.2 model was used throughout the StressEval pipeline for f_{ana}, f_{can}, f_{ext}, f_{know}, f_{U}, f_{U}. All API calls used deterministic decoding with temperature was set to 0.0, top-p was set to 0.95, and a maximum token limit was set to 2048. For each bad case $e \in \mathcal{E}$, StressEval will synthesize $5 \sim 10$ instances. In addition, we do not synthesize knowledge-stress instances for table reasoning, since the case of the WTQ dataset already contain all evidence required to answer the questions.

4.2 Benchmark Statistic

Table 1 reports statistics of the constructed dataset. For each split, we provide the number of synthesized instances $|\Omega^*|$, the size of the seed dataset $|\Omega|$, the size of the failure-case set $|\mathcal{E}|$, the average context and question lengths, the mean token number of $\mathcal{S}^*$ and $\mathcal{Q}^*$ (reported as $\text{avg}\,|\mathcal{S}^*|$ and $\text{avg}\,|\mathcal{Q}^*|$), the number of root causes $|\Gamma|$, and the token-level similarity between the synthesized and original questions, $\text{Sim}(\mathcal{Q}, \mathcal{Q}^*)$. Overall, STRESSEVAL generates a large number of challenging instances $|\Omega|$ from a relatively small collection of failure cases $|\mathcal{E}|$, demonstrating strong leverage of limited supervision. Moreover, in principle, with a continual stream of new failure cases, the construction process can keep producing arbitrarily many difficult instances, i.e., the dataset size is unbounded as the failure-case pool grows.

4.3 Diversity of Root Cases

Figure 3 shows the distribution of root-cause labels $\gamma \in \Gamma$ for both the original failure set $\mathcal{E}$ and the synthesized DYNAMIC-ONEEVAL. The inner ring indicates coarse-grained root-cause groups, while the outer ring breaks them down into the corresponding fine-grained categories. Overall, the synthesized benchmark largely preserves the diversity of error types

Type	Split	$\lvert\Omega^*\rvert$	$\lvert\Omega\rvert$	$\lvert\mathcal{E}\rvert$	Avg $\lvert\mathcal{S}^*\rvert$	Avg $\lvert\mathcal{Q}^*\rvert$	$\lvert\Gamma\rvert$	Sim($\mathcal{Q}$, $\mathcal{Q}^*$)	A(%)	U(%)	F(%)
Text	K-stress Γ_k	468	1000	78	1047.6	30.4	2	52.6	100.0	100.0	96.0
	R-stress Γ_r	388	500	146	407.3	30.1	10	53.5	98.0	100.0	90.0
KG	K-stress Γ_k	432	400	114	919.8	23.1	3	50.3	100.0	100.0	98.0
	R-stress Γ_r	347	1200	64	708.5	13.8	7	48.5	94.0	100.0	88.0
Table	R-stress Γ_r	294	1000	47	236.6	20.9	8	51.0	100.0	98.0	96.0
DYNAMIC-ONEEVAL		1929	4100	449	705.6	24.3	30	51.3	98.5	99.7	93.8

Table 1: Overall statistics of DYNAMIC-ONEEVAL and the results of human evaluation of different splits.

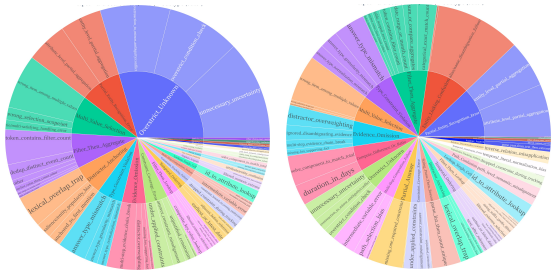

Figure 3: Left: root-cause distribution among the seed failure cases $\mathcal{E}$. Right: root-cause distribution of our DYNAMIC-ONEEVAL, which is comparatively more balanced.

present in the collected failures, and it also introduces modest expansion into additional categories. Notably, the label distribution in DYNAMIC-ONEEVAL is more balanced than in the seed failure set, reducing skew toward a few dominant root causes and yielding broader coverage of fine-grained model weaknesses.

4.4 Human Evaluation

We conducted a lightweight human evaluation to assess the quality of DYNAMIC-ONEEVAL. We randomly sampled 50 instances spanning both knowledge-stress Γ_k and reasoning-stress Γ_r, covering diverse root-cause labels $\gamma \in \Gamma$, and asked three annotators to provide judgments along three dimensions. First, *answerability* (**A**) evaluates whether the instance can be resolved from the provided context: for Γ_r, whether the gold answer $\mathcal{A}$ is supported by explicit context in $\mathcal{S}$; for Γ_k, whether solving the question $\mathcal{Q}$ necessarily requires an external fact missing from $\mathcal{S}$. Second, *unambiguity* (**U**) assesses whether a unique gold answer exists. Third, *faithfulness to difficulty cards* (**F**) measures whether the difficulty card Ψ is correctly instantiated, meaning that the specified bottleneck step and trigger are present in the instance. The results are reported in the right panel of Table 1. Overall, human judgments show consistently high **A** and **U** across splits, indicating that instances are generally solvable from the provided context and admit a unique gold answer. In contrast, **F** exhibits larger variance, with reasoning-stress splits showing notably lower faithfulness (e.g., $88.0\% \sim 96.0\%$) than knowledge-stress splits (up to 98.0%), suggesting that mismatches more often arise from imperfect instantiation of the intended bottleneck and trigger rather than from missing fact or ambiguity.

5 Experiments

5.1 Experimental Setup

We evaluate a set of representative open-source and proprietary LLMs on DYNAMIC-ONEEVAL, covering the Llama3.1 series [Grattafiori *et al.*, 2024] (Llama3.1-8B and Llama3.1-70B), the Qwen family [Yang *et al.*, 2024; Yang *et al.*, 2025] (Qwen2.5-72B and Qwen3-235B) and QWQ-32B [Qwen-Team, 2025], as well as proprietary models including GPT-5.2 [OpenAI, 2025], Gemini3-pro [Google, 2025], DeepSeek-V3.2 [Liu *et al.*, 2025], Claude-Sonnet-4.5 [Anthropic, 2025], and Doubao-Seed-1.6 [ByteDance, 2025]. To ensure a fair comparison, we apply a unified prompting template across all models. We use exact-match accuracy as the primary metric, reporting results by knowledge type (Text, KG, and Table) and stress type (K-Stress/R-Stress), together with per-type averages and an overall aggregated score.

5.2 Main Results

Table 2 reports the performance of different LLMs on DYNAMIC-ONEEVAL. The results show that DYNAMIC-ONEEVAL is genuinely hard and far from saturated: even the strongest proprietary model (Gemini3-pro) reaches only 48.2% overall, and the best open-source model (QWQ-32B) achieves 39.8%, with substantial error persisting across all knowledge types. This indicates that our failure-driven generation goes beyond superficial adversarial trickery, producing instances that are both challenging and unambiguous, and that reliably surface brittleness even in frontier LLMs, particularly on knowledge-intensive failure modes.

A consistent diagnostic pattern is that K-Stress is the dominant bottleneck, most severely in text reasoning: open-source models remain near-floor on Text K-Stress despite noticeably higher Text R-Stress, and although proprietary models improve (Gemini3-pro achieves 29.8%), most still hover around 10 on text K-Stress. In KG reasoning, R-Stress is comparatively high and clustered, while K-Stress drops sharply, indicating that graphs can skeleton compositional reasoning but cannot compensate when crucial knowledge is stressed or withheld. Finally, table reasoning yields high R-Stress for several models (e.g., Claude-Sonnet-4.5 achieves 80.9%, Gemini3-pro achieves 75.5%), yet overall scores remain limited because knowledge-stress failures persist, reinforcing DYNAMIC-ONEEVAL as a targeted, controlable stress test rather than a knowledge type-specific formatting challenge.

Type	Model	Text			KG			Table	Overall
		K-Stress	R-Stress	Avg.	K-Stress	R-Stress	Avg.	R-Stress	
Open-sourced LLM	Llama3.1-8B [Grattafiori *et al.*, 2024]	15.6	20.1	17.6	11.2	44.1	25.9	42.2	24.7
	Llama3.1-70B [Grattafiori *et al.*, 2024]	**21.4**	21.1	21.3	27.8	51.3	38.3	44.9	31.7
	Qwen2.5-72B [Yang *et al.*, 2024]	14.7	21.6	17.8	**33.9**	**53.0**	**42.4**	58.5	33.9
	QWQ-32B [QwenTeam, 2025]	17.9	**23.7**	**20.5**	22.4	50.7	35.0	**74.8**	**34.6**
	Qwen3-235B [Yang *et al.*, 2025]	16.7	21.6	18.9	13.0	50.7	29.8	54.1	28.7
Proprietary LLM	GPT-5.2 [OpenAI, 2025]	26.9	26.8	26.9	10.0	49.6	27.6	73.8	34.3
	Gemini3-pro [Google, 2025]	**54.3**	**27.6**	**42.2**	**28.6**	54.2	**40.0**	75.5	**46.4**
	DeepSeek-V3.2 [Liu *et al.*, 2025]	15.6	13.9	14.8	15.7	**55.0**	33.2	50.3	27.7
	Claude-Sonnet-4.5 [Anthropic, 2025]	31.2	23.7	27.8	13.9	53.4	31.5	**80.9**	37.4
	Doubao-Seed-1.6 [ByteDance, 2025]	32.9	15.7	25.1	1.9	51.8	24.1	47.6	28.1

Table 2: Performance (%) of open-source and proprietary LLMs on DYNAMIC-ONEEVAL. All test instances were generated using GPT-5.2.

5.3 Ablation Studies

We fix GPT-5.2 as the synthesis backbone, generate instances under each ablated setting, and evaluate the resulting datasets using DeepSeek V3.2, together with human judgments on the corresponding instances. Here, we consider three ablation groups. **Pipeline-level** ablations remove either (a) *error analysis*, to test whether bottleneck identification and trigger localization are necessary for controllable synthesis, or (b) *criterion gating*, to quantify its role in enforcing grounding, answerability, unambiguity, and difficulty preservation. **Knowledge-stress** ablations remove either (c) the *missing-fact black box* ΔK, to test whether explicitly modeling atomic missing facts stabilizes reliance on external knowledge, or (d) *context freezing*, allowing $\mathcal{S}$ to be rewritten and potentially filling the intended knowledge gap. **Reasoning-stress** ablations remove either (e) *virtual context* $\mathcal{S}^*$ construction, to assess whether entity virtualization prevents parametric-memory shortcuts, or (f) the *reasoning skeleton*, to test whether explicitly instantiating the bottleneck and injecting traps improves faithfulness and grounding.

Table 3 reports the ablation results for STRESSEVAL, showing that its components must operate in concert to preserve both difficulty and validity. At the pipeline level, removing error analysis substantially degrades performance (e.g., DeepSeek V3.2 drops from 27.7% overall in Table 2) and sharply reduces human-rated faithfulness. Removing criterion gating also consistently lowers human faithfulness, confirming its role in enforcing grounding, unambiguity, and difficulty preservation. For K-stress, disabling the missing-fact black box K produces the most severe failures, while unfreezing $\mathcal{S}$ weakens the intended knowledge gap, consistent with context leakage. For R-stress, removing the virtual context $\mathcal{S}^*$ or the two-step reasoning-skeleton procedure yields instances that are noticeably easier and less faithful, supporting the claim that virtualization mitigates parametric-memory shortcuts and that an explicit skeleton enforces context-grounded reasoning.

5.4 Performance on Different Root Causes

Figure 4 compares models across fine-grained root error causes; due to space constraints, we report representative categories. Performance is highly error-dependent: frontier proprietary models excel on structured temporal/compositional skills. For instance, Claude-Sonnet-4.5 achieves 95% on `Temporal_Ordering`, whereas gaps widen for failures requiring robust entity grounding and multi-step control. For instance, `Partial_Entity_Recognition_Then_Reason` falls from 85% (Claude-4.5) to 15% (DeepSeek-V3.2). Across models, `Evidence_Omission` and `Distractor_Anchoring` remain difficult, suggesting that brittleness stems less from surface retrieval than from evidence use under adversarial distractions.

Variant	Text↓	KG↓	Table↓	A	F
DYNAMIC-ONEEVAL	14.8	**33.2**	50.3	**98.4**	**93.6**
w/o error analysis	68.9	40.3	61.7	90.0	48.0
w/o criterion gating	**11.7**	43.1	**39.2**	92.0	84.0
Knowledge-stress	**15.6**	**15.7**	–	**100.0**	**97.0**
w/o black box	66.7	83.6	–	98.0	96.0
w/o freezing $\mathcal{S}$	32.1	23.8	–	84.0	92.0
Reasoning-stress	**13.9**	**55.0**	50.3	97.3	**91.3**
w/o virtual $\mathcal{S}^*$	59.7	61.8	54.5	96.0	88.0
w/o reasoning skeleton	56.8	66.5	**34.1**	**98.0**	80.0

Table 3: Ablation study on DYNAMIC-ONEEVAL (using GPT-5.2 for data synthesis). We report performance (%) of DeepSeek V3.2.

5.5 Impact of Different Backbones

To assess STRESSEVAL's robustness to the backbone, we reran the pipeline with several lightweight/low-cost LLMs as generators and evaluated the resulting instances using representative LLMs. Figure 5 summarizes the results. Model rankings are largely stable across backbones, suggesting limited generator-induced bias, while the backbone mainly affects absolute difficulty: Llama-4-scout generally yields harder cases (lower scores) and Gemini2.5 tends to yield easier ones (higher scores) across all three settings. Notably, Gemini3 performs particularly well on Gemini2.5-generated cases, consistent with potential within-family transfer effects.

6 Related Work

Knowledge-Intensive Reasoning Benchmarks Early success of LLMs, exemplified by models like GPT-4 [OpenAI,

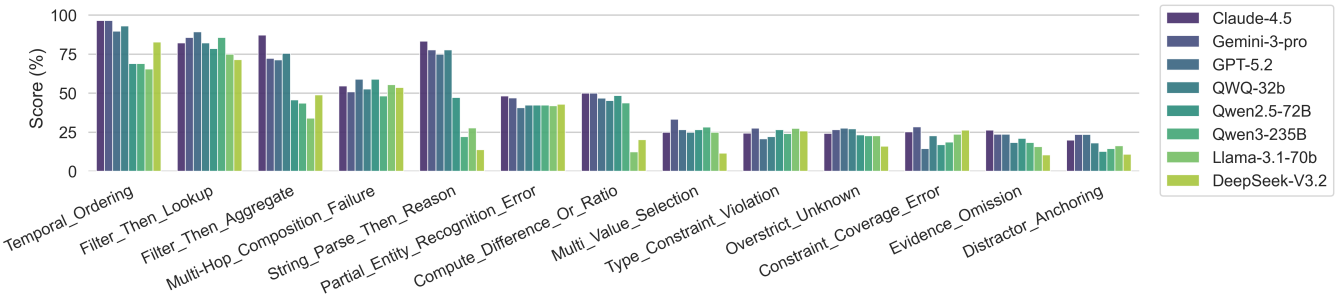

Figure 4: Performance of different LLMs on each root cause. Due to space limitations, only representative root causes are included here.

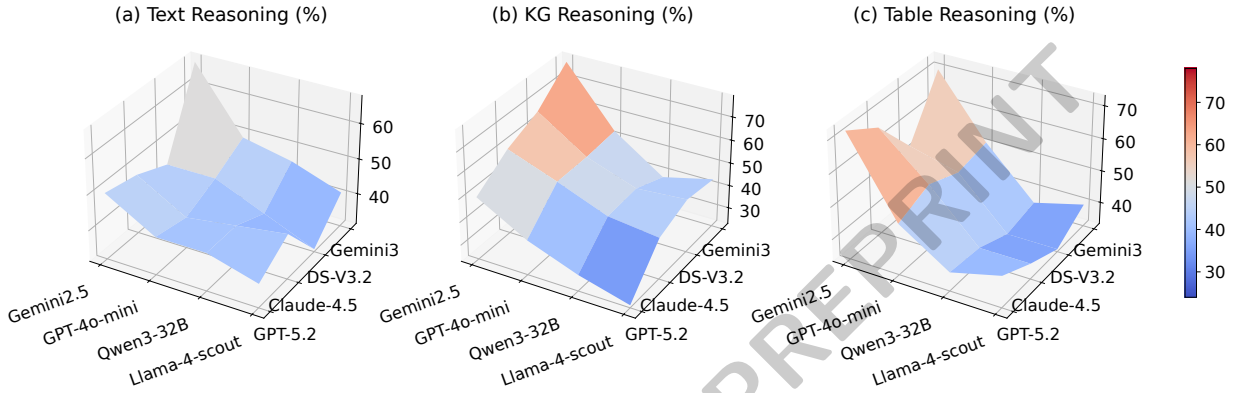

Figure 5: Performance of StressEval using different backbones. We use different backbone LLMs (x-axis) to run StressEval and generate evaluation instances, and then benchmark a set of target LLMs (y-axis) on the resulting datasets.

2023] and DeepSeek [Liu *et al.*, 2024], demonstrated impressive capabilities across a spectrum of language understanding and reasoning tasks. However, conventional static evaluation practices have struggled to keep pace with the rapid evolution of these models. Issues such as benchmark contamination [Deng *et al.*, 2024; Dong *et al.*, 2025], where models may memorize test data present in their vast training corpora, lead to an inflated sense of performance and obscure genuine generalization abilities [Lin *et al.*, 2024b]. This problem is particularly acute in knowledge-intensive settings [Chen *et al.*, 2025a; Chen *et al.*, 2022], where correctness relies on up-to-date facts, strict grounding, and robust multi-step inference, rather than superficial pattern matching [Jiang *et al.*, 2024]. The need for benchmarks that can genuinely assess an LLM's ability to reason with and apply external knowledge, rather than merely recall memorized patterns, remains a significant challenge.

Dynamic LLM Benchmarks Automatic benchmarking approaches [Wang *et al.*, 2025] broadly span four categories: static human-curated suites, LLM-as-judge frameworks [Zheng *et al.*, 2023], agent-driven automated arenas [Zhao *et al.*, 2025], and *in-the-wild* benchmarks mined from real interactions [Lin *et al.*, 2024a]. While each improves scalability or realism, they share complementary weaknesses: static benchmarks saturate and risk contamination; mined sets lack controllable latent factors for causal stress tests [Valiulla, 2025]; and fully automated pipelines may generate unreliable or causally unfaithful cases [Shukla, 2025]. Model-in-the-loop and adversarial generation frameworks [White *et al.*, 2024; Li *et al.*, 2024] help surface brittle heuristics but, without strict answerability constraints, risk drifting toward underspecified or spurious examples.

7 Conclusion

We present StressEval, a failure-driven pipeline that converts observed LLM failures into dynamic, answerable, and high-difficulty evaluation instances through structured error analysis, dual-perspective instance synthesis, and multi-criterion gating. Instantiated from knowledge-intensive seed datasets, StressEval produces grounded and controlable tests that induce substantially larger, and more diagnostic, performance drops across frontier models than standard static benchmarks. Future work will investigate how different combinations of LLMs within the same StressEval framework affect data generation, e.g., using heterogeneous generator–critic–verifier ensembles, to improve diversity, robustness, and controllability.

Acknowledgements

This work is supported by National Nature Science Foundation of China under No.62506071, No.6250072425, and No.62476058. We thank the Big Data Computing Center of Southeast University for providing the facility support on the numerical calculations in this paper.

References

[Anthropic, 2025] Anthropic. Claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025. Accessed: 2026-01-12.

[Anwar *et al.*, 2024] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, Benjamin L. Edelman, Zhaowei Zhang, Mario Günther, Anton Korinek, José Hernández-Orallo, Lewis Hammond, Eric J. Bigelow, Alexander Pan, Lauro Langosco, Tomasz Korbak, Heidi Chenyu Zhang, Ruiqi Zhong, Seán Ó hÉigeartaigh, Gabriel Recchia, Giulio Corsi, Alan Chan, Markus Anderljung, Lilian Edwards, Aleksandar Petrov, Christian Schröder de Witt, Sumeet Ramesh Motwani, Yoshua Bengio, Danqi Chen, Philip Torr, Samuel Albanie, Tegan Maharaj, Jakob Nicolaus Foerster, Florian Tramèr, He He, Atoosa Kasirzadeh, Yejin Choi, and David Krueger. Foundational challenges in assuring alignment and safety of large language models. *Trans. Mach. Learn. Res.*, 2024, 2024.

[Brown *et al.*, 2020] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[ByteDance, 2025] ByteDance. Doubao seed 1.6. https://seed.bytedance.com/en/seed1_6, 2025. Accessed: 2026-01-12.

[Chen *et al.*, 2022] Yongrui Chen, Huiying Li, Guilin Qi, Tianxing Wu, and Tenggou Wang. Outlining and filling: Hierarchical query graph generation for answering complex questions over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8343–8357, 2022.

[Chen *et al.*, 2025a] Yongrui Chen, Yi Huang, Yunchang Liu, Shenyu Zhang, Junhao He, Tongtong Wu, Guilin Qi, and Tianxing Wu. K-decore: Facilitating knowledge transfer in continual structured knowledge reasoning via knowledge decoupling. *arXiv preprint arXiv:2509.16929*, 2025.

[Chen *et al.*, 2025b] Yongrui Chen, Zhiqiang Liu, Jing Yu, Lin Ren, Nan Hu, Xinbang Dai, Jiajun Liu, Jiazhen Kang, Shenyu Zhang, Xinda Wang, et al. Oneeval: Benchmarking llm knowledge-intensive reasoning over diverse knowledge bases. *arXiv preprint arXiv:2506.12577*, 2025.

[Chen, 2021] Mark Chen. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[Deng *et al.*, 2024] Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. 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 8706–8719, 2024.

[Dong *et al.*, 2025] Yihong Dong, Xue Jiang, Xuanming Zhang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Evaluating data contamination for large language models, 2025.

[Google, 2025] Google. Gemini api documentation — models (gemini 3 pro). https://ai.google.dev/gemini-api/docs/models#gemini-3-pro, 2025. Accessed: 2026-01-12.

[Grattafiori *et al.*, 2024] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[Hendrycks *et al.*, 2021] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[Jiang *et al.*, 2024] Weiming Jiang, Xin Sun, Jian Li, Jun Li, and Yang Liu. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2403.11187*, 2024.

[Kandpal *et al.*, 2024] Naman Kandpal, Kevin Li, Yi Zhang, Pengfei Wang, Jiatao Li, Yu Wang, Zhiheng Chen, and Percy Liang. Mind the gap: Data contamination in llm benchmarks. *arXiv preprint arXiv:2404.03222*, 2024.

[Li *et al.*, 2024] Qintong Li, Jiahui Gao, Sheng Wang, Renjie Pi, Xueliang Zhao, Chuan Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Forewarned is forearmed: Leveraging llms for data synthesis through failure-inducing exploration. *arXiv preprint arXiv:2410.16736*, 2024.

[Lin *et al.*, 2024a] Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*, 2024.

[Lin *et al.*, 2024b] Zhiting Lin, Zhiyong Xu, Guodong Zeng, Xipeng Li, Jiliang Tang, Qi Lv, and Jie Yang. Rethinking llm evaluation: A comprehensive survey. *arXiv preprint arXiv:2407.03713*, 2024.

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

[Liu *et al.*, 2025] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[OpenAI, 2023] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[OpenAI, 2025] OpenAI. Openai api documentation (model: Gpt-5.2). https://platform.openai.com/docs, 2025. Accessed: 2026-01-12.

[Pasupat and Liang, 2015] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, 2015.

[QwenTeam, 2025] QwenTeam. Qwq-32b: Embracing the power of reinforcement learning. https://qwen.ai/blog?id=qwq-32b, March 2025. Accessed: 2025/03/06.

[Shukla, 2025] Anil Kumar Shukla. Large language model evaluation in 2025: Smarter metrics that separate hype from trust. 2025.

[Talmor and Berant, 2018] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, 2018.

[Valiulla, 2025] Adeeb Valiulla. From benchmarks to real-world performance: A data-driven assessment of large language models in 2025. *Available at SSRN 5991894*, 2025.

[Wang *et al.*, 2025] Zongqi Wang, Tianle Gu, Chen Gong, Xin Tian, Siqi Bao, and Yujiu Yang. From rankings to insights: Evaluation should shift focus from leaderboard to feedback. *arXiv preprint arXiv:2505.06698*, 2025.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[White *et al.*, 2024] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, et al. Livebench: A challenging, contamination-limited llm benchmark. *arXiv preprint arXiv:2406.19314*, 2024.

[Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380, 2018.

[Yang *et al.*, 2024] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

[Yue *et al.*, 2024] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[Zhao *et al.*, 2025] Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Weiwen Xu, Deli Zhao, and Lidong Bing. Auto-arena: Automating llm evaluations with agent peer battles and committee discussions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4440–4463, 2025.

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[Zhou *et al.*, 2025] Yujia Zhou, Zheng Liu, and Zhicheng Dou. How credible is an answer from retrieval-augmented llms? investigation and evaluation with multi-hop qa. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4232–4242, 2025.