VIRALBENCH: DO LLMS LEARN OR MEMORIZE? A BILINGUAL BENCHMARK OF VIRAL TRICK QUESTIONS AND THEIR PERTURBED VARIANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Every few months, a single "trick question" goes viral on social media by making state-of-the-art large language models (LLMs) fail in embarrassingly simple ways: counting the letter *r* in *strawberry*, deciding whether 9.9 or 9.11 is larger, or asking whether one should walk or drive to a car wash 50 meters away. Within weeks, the newest model checkpoints "fix" these questions—but is such fixing genuine capability improvement, or targeted memorization of the specific viral instance? We introduce **ViralBench**, a bilingual (Chinese–English) benchmark of **64 provenance-annotated seed questions** that genuinely went viral—each annotated with its source, the period it circulated, and the language in which it went viral—plus **450 programmatically generated variants** organized in a **three-tier design**: the original seed, a *trivial control* with the trap removed, and *perturbed* instances that preserve the failure mechanism while changing surface details. All 514 items are organized by a **two-axis taxonomy** crossing surface form (7 categories) with failure mechanism (6 modes), and answers are computed by construction. Evaluating 15 models—including *longitudinal version series* (six dated Qwen-Plus snapshots spanning 2025; DeepSeek V3 $\rightarrow$ V3.1 $\rightarrow$ V3.2 $\rightarrow$ V4-Pro)—under a deterministic 975-query protocol, we measure a **memorization gap** between seed and perturbed accuracy. The gap is not one number but three signatures: early Qwen-Plus snapshots *fail the exact viral instance* ("9.9 vs. 9.11" in English) while passing novel perturbations of the same mechanism—the famous error itself is absorbed, then patched between the April and July 2025 snapshots; aggregate positive gaps concentrate in two families (strawberry, 9.9-vs-9.11) and shrink toward zero without them; and on 2026-era items that postdate every training cutoff the gap turns positive for 14 of 15 models (up to $+0.40$), showing that the trap mechanisms behind viral failures generalize to instances no model could have memorized. The version-series design lets us watch the gap evolve across releases of the same model family, turning "learn or memorize" from a philosophical question into a measurable trajectory. We release the dataset, the deterministic variant generator, and the evaluation harness.

1 INTRODUCTION

Public perception of LLM competence is disproportionately shaped by a stream of *viral trick questions*. The timeline is by now familiar: the Reversal Curse discussions of 2023; "which is bigger, 9.9 or 9.11?" in July 2024; "how many *r*'s in *strawberry*?" in August 2024; a "three b's in blueberry" incident in August 2025; the "car wash" pragmatic trap in early 2026; and the "seahorse emoji" hallucination. Each follows the same lifecycle: a question surfaces, most deployed models fail it simultaneously, screenshots circulate, and the *next* model release answers it correctly.

This lifecycle raises a question that is scientifically important but empirically unaddressed: **when a new model "passes" a formerly viral question, has it acquired a transferable capability, or has it simply memorized the specific instance** that appeared thousands of times in its training data and post-training feedback? The distinction matters for how we interpret benchmark progress: a model

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

that patches individual viral instances will look competent on exactly the probes the public uses, while remaining brittle on trivial perturbations.

Answering this question requires three things the community currently lacks: (i) a *curated collection* of genuinely viral questions with provenance and dates, (ii) *controlled perturbations* of each question whose correct answers are known by construction—together with *trivial controls* that rule out the confound "perturbed items are simply harder"—and (iii) a *reproducible evaluation protocol* with deterministic scoring. ViralBench provides all three.

Beyond collection, we contribute two conceptual tools. First, a **two-axis taxonomy**: the community discusses these questions by surface form ("the strawberry question"), but surface form is not the right unit of analysis—visually distinct questions can share a mechanism, and visually similar questions can differ in mechanism (e.g., "how many apples did I have *yesterday*?" looks like a physical-commonsense question but fails through pragmatic presupposition, not physics). We therefore annotate every item along two axes—surface `category` (7 values) and abstract `failure_mode` (6 values)—and argue that the mechanism axis is the generalizable one. Second, a **longitudinal design**: hosting platforms now serve *dated snapshots* of the same model family side by side, letting us evaluate a whole version series under one identical protocol and observe whether the memorization gap shrinks (learning) or the seed accuracy alone rises (patching).

Contributions.

1. **ViralBench dataset**: 64 provenance-annotated viral seed questions (source URL, viral period, *viral language*) + 450 deterministic variants in a three-tier design (seed / trivial control / perturbed), bilingual (zh/en), 514 items, released with a byte-reproducible generator (fixed seed 20260727).
2. **Two-axis taxonomy**: surface `category` × `failure_mode`, with a principled argument for why the two axes are not one-to-one.
3. **Memorization-gap methodology**: a three-tier accuracy decomposition that operationalizes "learned vs. memorized," robust to the "variants are harder" confound and to family-size concentration.
4. **Longitudinal & temporal analysis**: version-series trajectories (Qwen-Plus snapshots; DeepSeek V3→V4-Pro) and a natural contamination experiment aligning each seed's viral period with model training cutoffs.
5. **Empirical study**: 15 models evaluated with 975 queries each under a deterministic protocol, including a reasoning model, along the mechanism axis, the memorization gap, the viral-language fingerprint, and cross-lingual consistency.

2 Related Work

Adversarial and brittleness benchmarks. A growing line of work shows that LLMs fail on inputs that are trivial for humans. The AIW ("Alice in Wonderland") study (Nezhurina et al., 2024) reports dramatic accuracy collapse on simple relational counting problems under minor rephrasing. The Heuristic Override Benchmark (HOB) (Li et al., 2026) formalizes commonsense "obvious" questions—of which the car-wash question is the canonical instance—showing that models follow surface heuristics (distance) over unstated constraints (the car must be present). The Reversal Curse (Berglund et al., 2023) shows that a model trained on "A is B" often fails "B is A." ViralBench differs by treating *virality itself* as the sampling criterion: rather than constructing adversarial inputs, we collect inputs that empirically broke deployed models in the wild, then add controlled perturbations and trivial controls.

Memorization and contamination. Concerns that benchmark scores reflect training contamination have motivated re-authored test sets—e.g., GSM1k (Zhang et al., 2024), a freshly written analogue of a popular grade-school math set—which measure overfitting via the accuracy gap between original and re-authored items. Our memorization gap adopts the same logic but is tailored to viral questions: the "original" is a specific instance with a *datable* viral period, enabling a temporal contamination analysis that re-authoring alone cannot provide. The version-series design further distinguishes two mechanisms that aggregate contamination checks conflate: a family can improve

because new snapshots *learned the skill* (gap shrinks) or because they *absorbed the instance* (seed accuracy rises, gap persists).

Chinese-community data. The "Ruozhiba" (弱智吧) subset of the COIG-CQIA corpus (Bai et al., 2025) showed that absurd or trap-like questions are unusually informative training signal. ViralBench is complementary: we use such questions as *evaluation* probes and pair each Chinese item, where meaningful, with an English counterpart.

Positioning. Prior work typically isolates a single mechanism (tokenization (Sennrich et al., 2016), relational reasoning, or reversal). ViralBench is the first to assemble the *viral trick question* as a phenomenon in its own right and to impose a cross-mechanism, bilingual, provenance-annotated, longitudinal structure on it.

3 THE VIRALBENCH DATASET

3.1 COLLECTION METHODOLOGY

We assembled seed questions through (a) social-media archaeology across Weibo, Zhihu, X, Reddit, and Mastodon, and (b) back-tracing academic and blog write-ups to their original viral instance. A candidate is admitted iff it (i) demonstrably went viral (we record a `viral_period` and, where available, a `source_url`), (ii) has a well-defined, machine-checkable canonical answer, and (iii) can be assigned to a failure mechanism. Each item stores its bilingual phrasings, provenance, a viral period, a *viral language* (`viral_language` $\in$ {zh, en, both}: the language community in which the question primarily circulated; 33/20/11 seeds respectively), and a machine-readable scoring rule (Section 3.4).

The resulting 64 seeds span 2023–2026 (plus 3 "classic" items of uncertain age treated as pre-2023). Where the original was monolingual (e.g., a Chinese homophone pun or an English lipogram) we keep it monolingual and set `language_specific = true` (53 of 514 items); otherwise we provide aligned zh/en phrasings, yielding 975 queries per model under the bilingual protocol ($(514 - 53) \times 2 + 53$).

3.2 THE TWO-AXIS TAXONOMY

category		items	example
numeric_comparison	数值比较	126	is 9.9 or 9.11 bigger
char_perception	字符感知	123	how many *r* in *strawberry*
overfitted_riddle	过拟合谜题	101	trivialized river-crossing; captain's age
physical_commonsense	物理常识	53	the car-wash / drying-clothes question
self_reference	自指约束	51	answer without using the character 的
knowledge_consistency	知识一致性	33	seahorse emoji; reversal curse
pragmatic_trap	语用陷阱	27	is a cooked oyster still a "raw oyster"?
total		**514**	

Table 1: Surface-form axis with final item counts (seeds + variants).

Axis 1 — `category` (surface form; how the question circulates).

Axis 2 — `failure_mode` (abstract mechanism; why the model errs).

The axes are not one-to-one. A given surface category can draw from multiple mechanisms, and this is the crux of our argument that mechanism is the generalizable unit. For example, item `pc-008` ("I bought 3 apples yesterday and ate 1 this morning; how many did I have *yesterday*?") is surface-classified as physical_commonsense but is mechanistically a pragmatic_presupposition_failure—the temporal reference, not physics, is the trap. We encode such cases via per-item and per-family mechanism overrides on top of a category-default mapping.

failure_mode	items	hypothesis
memorized_pattern_override	227	a high-frequency template/solution overrides the actual prompt
sub_token_perception	123	BPE tokenization (Sennrich et al., 2016) hides characters from the model
self_output_monitoring	51	the model cannot count/constrain its own ongoing output
pragmatic_presupposition_failure	42	a false presupposition / ambiguity is not detected
implicit_constraint_neglect	38	salient surface cues override an unstated physical constraint
knowledge_consistency_failure	33	uni-directional memory, entity confusion, fabricated memory

Table 2: Failure-mechanism axis. Every variant inherits its seed's mechanism, enforced by an automated test suite (43 tests).

3.3 Three-tier variant design

For 40 of the 64 seed families we implement deterministic generators (fixed seed `20260727`) producing structurally identical but novel instances. All variants carry `variant_of` linking them to their seed and inherit the seed's `failure_mode`. Rule-scored variants compute their canonical answer by construction, guaranteeing zero annotation error. The generator is *byte-stable*: regenerating the dataset reproduces every item exactly.

The 450 variants form two tiers, and together with the seeds the dataset has a **three-tier design**:

1. **Seed** (64 items): the exact viral instance that circulated online. High accuracy here is consistent with either capability or memorization.
2. **Trivial control** (20 items across 18 families, `perturbation=trivial`): the family's question with the *trap removed*—e.g., in the hunter-and-birds riddle no shot is fired and one bird simply flies away on its own (pure arithmetic, $10-1=9$); in the car-wash question the shop is 50 km away, so driving is overdetermined. These are answerable by plain reading comprehension. They guard against the confound "perturbed items are just harder": if a model scores high on trivial controls but low on perturbations of the same family, the deficit is the trap, not difficulty.
3. **Perturbed variant** (430 items): the trap is preserved while surface details change—new words/letters for character counting, new decimal pairs, new entities for reversal-curse and fabricated-plot items, new quantities for race/trade/arithmetic puzzles, de-trapped-then-retrapped riddles.

Perturbation strategies include **substitution** (new target word/letter, new numeric pairs), **entity swap** (celebrity parent–child pairs; characters placed into the wrong novel), **quantity re-derivation** (profit of a chicken trade, siblings counting, race times with a sleeping hare—answers computed programmatically), and **constraint transformation** (version-string vs. decimal comparison with reversed semantics, unit-equality traps, trailing-zero stripping).

3.4 Scoring rules

Items carry one of seven machine-readable scoring types. Distribution over the 514 items: numeric 212, choice 135, llm_judge 88, contains 45, regex 15, not_contains 13, word_count 6.

4 Evaluation Framework

Prompting protocol. Each item is asked zero-shot, single-turn, with no system prompt, at `temperature = 0`. Bilingual items are asked once in each language and scored independently, yielding 975 queries per model.

Deterministic scoring.

- `numeric` — a three-level conclusion extractor: (1) answer declarations ("答案是 X" / "the answer is X"); (2) other conclusion phrasings ("一共 X," "有 X 个," "出现 X 次," "X times," arithmetic "= X"), taking the *last* match in the response with a negation-context filter ("not 2 times," "误答成 2 个" are skipped); (3) fallback to the last number. This handles the common failure of naive last-number extraction when responses end with a distractor ("... 3 r's. Many people mistakenly answer 2.").
- `choice` — parse explicit comparison assertions ("X 比 Y 大" / "X is bigger than Y") and take the subject of the last assertion; otherwise a last-occurrence position heuristic.
- `contains` / `not_contains` — membership / prohibition checks (the latter for self-constraint tasks such as avoiding a character).
- `regex` — language-specific patterns for exact-format tasks.
- `word_count` — exact length plus required content.

LLM-as-judge. The 88 open-ended items (e.g., anachronism detection, fabricated-plot questions, pen-name knowledge) use an LLM judge (Zheng et al., 2023) with an item-specific rubric returning PASS/FAIL. The judge (Qwen-Max) is disjoint from all evaluated models to avoid self-preference. In a stratified manual audit (Section A), a human annotator confirmed 96.6% (28/29) of sampled judge verdicts on Chinese items across 14 models; all 26 audited PASS verdicts were confirmed—no leniency bias—and the single error was an overly strict FAIL.

5 Experimental Setup

5.1 Models

All models are evaluated through a single OpenAI-compatible endpoint (Alibaba Cloud Bailian), guaranteeing identical serving conditions across vendors. Table 3 lists the lineup; Section 5.2 explains the version-series design.

model	family / role	type	queries
qwen-plus-2025-01-25	Qwen-Plus snapshot	standard	975
qwen-plus-2025-04-28	Qwen-Plus snapshot	standard	975
qwen-plus-2025-07-14	Qwen-Plus snapshot	standard	975
qwen-plus-2025-09-11	Qwen-Plus snapshot	standard	975
qwen-plus-2025-11-05	Qwen-Plus snapshot	standard	975
qwen-plus-2025-12-01	Qwen-Plus snapshot	standard	975
qwen-plus	Qwen-Plus current alias	standard	975
qwen-flash	small-model contrast	standard	975
deepseek-v3	DeepSeek generation g	standard	975
deepseek-v3.1	DeepSeek generation $g{+}1$	hybrid	975
deepseek-v3.2	DeepSeek generation $g{+}2$	hybrid	975
deepseek-v4-pro	DeepSeek generation $g{+}3$	hybrid	975
deepseek-r1	reasoning contrast (Guo et al., 2025)	reasoning	975
kimi-k2.5	cross-vendor contrast	standard	975
glm-5.3	cross-vendor contrast	standard	975

Table 3: Evaluated models. The Qwen-Plus dated snapshots and the DeepSeek V3→V4 series enable the longitudinal analysis; DeepSeek-R1 tests whether explicit re-derivation substitutes for recall.

5.2 Longitudinal version-series protocol

Hosting platforms serve dated snapshots of a model family side by side. A snapshot released at date d could only have memorized viral instances that circulated before its training cutoff (which upper-bounds by d). Evaluating an entire series under one protocol yields a *trajectory* of the memorization gap:

- **Patching signature**: seed accuracy rises across snapshots while the seed-vs-perturbed gap persists — the family is absorbing instances.

- **Learning signature**: the gap shrinks across snapshots as perturbed accuracy rises toward the trivial-control ceiling — the family is acquiring the underlying skill.

This is the paper's central methodological upgrade over one-shot evaluations: the question "do models learn or memorize viral questions?" becomes observable as a trend within a controlled model family.

5.3 METRICS

Per model m (and per snapshot within a series):

$$\text{gap}_{\text{seed}\to\text{perturb}}(m) = \text{acc}_{\text{seed}}(m) - \text{acc}_{\text{perturb}}(m) \quad (1)$$

$$\text{gap}_{\text{seed}\to\text{trivial}}(m) = \text{acc}_{\text{seed}}(m) - \text{acc}_{\text{trivial}}(m) \quad (2)$$

$$\text{gap}_{\text{trivial}\to\text{perturb}}(m) = \text{acc}_{\text{trivial}}(m) - \text{acc}_{\text{perturb}}(m) \quad (3)$$

Eq. equation 1 is the headline memorization gap; Eq. equation 3 is the pure "trap cost" net of item difficulty. We additionally report per-family gaps (with a robustness check excluding the two largest families, which contribute 31.8% of variants), the *viral-language fingerprint* $\Delta_{\text{fp}} = \text{gap}(\text{lang} = \text{viral_language}) - \text{gap}(\text{lang} \neq \text{viral_language})$ over bilingual items, and the era-conditioned gap (seeds bucketed by viral period: pre-2023 "classic," 2023, 2024, 2025, 2026).

6 RESULTS

6.1 ACCURACY BY FAILURE MECHANISM

model	sub-tok	pattern	impl.con.	pragm.	self-mon.	knowl.	overall
Qwen-Plus 01-25	86.5	82.8	81.6	83.6	68.8	84.5	82.7
Qwen-Plus 04-28	95.4	86.1	77.6	78.1	70.1	94.8	86.4
Qwen-Plus 07-14	**99.6**	**95.6**	85.5	83.6	81.8	96.5	**93.8**
Qwen-Plus 09-11	98.3	90.1	84.2	89.0	77.9	94.8	90.9
Qwen-Plus 11-05	96.2	89.0	86.8	83.6	80.5	**98.3**	90.0
Qwen-Plus 12-01	95.8	87.0	86.8	84.9	79.2	96.5	88.9
Qwen-Plus (current)	95.8	87.0	**88.2**	84.9	79.2	96.5	89.0
Qwen-Flash	94.9	91.2	82.9	86.3	67.5	86.2	88.9
DeepSeek-V3	97.0	76.4	53.9	80.8	83.1	91.4	81.4
DeepSeek-V3.1	97.0	85.7	63.2	89.0	81.8	93.1	87.1
DeepSeek-V3.2	97.9	86.6	80.3	**90.4**	87.0	94.8	89.6
DeepSeek-V4-Pro	82.7	76.6	85.5	86.3	**92.2**	91.4	81.6
DeepSeek-R1	60.3	87.4	80.3	86.3	79.2	89.7	79.7
Kimi-K2.5	95.4	64.3	81.6	78.1	74.0	89.7	76.5
GLM-5.3	84.8	53.1	71.0	72.6	87.0	94.8	68.8

Table 4: Accuracy (%) by failure mechanism over all 975 queries per model. Abbreviations: sub-tok = sub_token_perception, pattern = memorized_pattern_override, impl.con. = implicit_constraint_neglect, pragm. = pragmatic_presupposition_failure, self-mon. = self_output_monitoring, knowl. = knowledge_consistency_failure. Best per column in bold.

Table 4 reports the mechanism-axis main result. Overall accuracy spans 68.8% (GLM-5.3) to 93.8% (Qwen-Plus 07-14), and three observations structure everything that follows. **(1) The hardest mechanism is model-specific, not universal.** GLM-5.3 and Kimi-K2.5 collapse on *memorized_pattern_override* (53.1% / 64.3%) while scoring $\geq$ 89.7% on knowledge consistency; DeepSeek-V3/V3.1 instead collapse on *implicit_constraint_neglect* (53.9% / 63.2%). No single difficulty ordering explains all models. **(2) Explicit reasoning does not repair perception.** DeepSeek-R1 posts the worst *sub_token_perception* in the entire lineup (60.3%, vs. $\geq$ 82.7% for every other model) and scores 0.00 on the exact *strawberry* seed while reaching 44.4% on its novel variants—long chains of thought can talk a model *out of* a count it would otherwise get right. **(3) Version beats scale.** Small Qwen-Flash (88.9% overall) matches the current Qwen-Plus alias (89.0%); differences across releases within a family dominate differences in model size across families.

6.2 THREE-TIER DECOMPOSITION OF THE MEMORIZATION GAP

model	seed	trivial	perturb	$\Delta_{s\to p}$	$\Delta_{s\to t}$	$\Delta_{t\to p}$
Qwen-Plus 01-25	0.776	0.919	0.830	-0.054	-0.143	0.089
Qwen-Plus 04-28	0.819	0.892	0.869	-0.050	-0.073	0.023
Qwen-Plus 07-14	0.922	0.973	0.939	-0.017	-0.051	0.034
Qwen-Plus 09-11	0.897	1.000	0.906	-0.010	-0.103	0.094
Qwen-Plus 11-05	0.871	0.892	0.905	-0.034	-0.021	-0.013
Qwen-Plus 12-01	0.828	0.892	0.898	-0.070	-0.064	-0.006
Qwen-Plus (current)	0.836	0.892	0.898	-0.062	-0.056	-0.006
Qwen-Flash	0.836	0.919	0.895	-0.059	-0.083	0.024
DeepSeek-V3	0.793	0.757	0.820	-0.027	0.036	-0.063
DeepSeek-V3.1	0.853	0.865	0.874	-0.020	-0.011	-0.009
DeepSeek-V3.2	0.853	0.919	0.901	-0.048	-0.066	0.018
DeepSeek-V4-Pro	0.862	0.811	0.810	0.052	0.051	0.001
DeepSeek-R1	0.776	0.838	0.798	-0.022	-0.062	0.040
Kimi-K2.5	0.836	0.946	0.747	0.089	-0.110	0.199
GLM-5.3	0.776	0.865	0.668	0.108	-0.089	0.197

Table 5: Three-tier accuracy decomposition and the three gaps (Eqs. 1–3). The trivial-control tier separates "memorized the instance" ($\Delta_{s\to t}$) from "defeated by the trap" ($\Delta_{t\to p}$).

The decomposition separates two very different failure patterns. **Trap cost dominates for Kimi and GLM**: both handle the trivial controls well (0.946 / 0.865) yet lose ≈ 0.20 between de-trapped and re-trapped readings of the same families ($\Delta_{t\to p} = 0.199$ / 0.197, the two largest in the study)—they are defeated by the trap itself on novel instances, not by item difficulty. **Instance absorption dominates for the Qwen series**: trivial accuracy is near ceiling (0.89–1.00) and $\Delta_{t\to p} \leq 0.094$, yet $\Delta_{s\to t}$ is strongly negative (down to -0.143 for 01-25): the exact viral instances are *harder* for these models than de-trapped readings of the same families—the signature of having absorbed the famous instance together with its famous wrong answer. DeepSeek-V4-Pro stands apart: its seed accuracy exceeds even its own trivial controls ($\Delta_{s\to t} = +0.051$) while its perturbed tier tracks the trivial tier almost exactly ($\Delta_{t\to p} = +0.001$)—seed accuracy rising without the perturbed tier following, the pure *patching* signature predicted in Section 5.2.

Robustness. The two largest families (`cp-001` strawberry, `nc-001` 9.9-vs-9.11) contribute 31.8% of all variants—and, it turns out, most of the aggregate gap signal. Excluding them moves every positive gap toward zero: GLM-5.3 $+0.108 \to +0.048$, Kimi-K2.5 $+0.089 \to +0.028$, DeepSeek-V4-Pro $+0.052 \to -0.015$; the remaining twelve models all lie within ± 0.07 both before and after exclusion. The aggregate memorization gap is therefore *family-concentrated* and must be read together with per-family gaps (`results/family_gap.csv`), where the effects are dramatic in both directions: GLM-5.3 scores 1.00 on the exact 9.9-vs-9.11 seed but 0.37 on its 142 novel variants (family gap $+0.634$), while DeepSeek-R1 scores 0.00 on the exact *strawberry* seed but 0.44 on its variants (family gap -0.444).

6.3 LONGITUDINAL: DO VERSION SERIES LEARN OR PATCH?

Figure 1 plots the trajectories.

Qwen-Plus: patch, then drift. Seed accuracy rises from 0.776 (01-25) to a peak of 0.922 (07-14). The July snapshot is also where the famous English 9.9-vs-9.11 failure is fixed: 01-25 and 04-28 answer the Chinese phrasing correctly but fail the English one, while 07-14 passes both—dating that patch to the April–July 2025 window. The aggregate gap shrinks in step ($-0.054 \to -0.017$), driven from the seed side. But instance-level competence is *not monotone*: seed accuracy regresses to 0.871 (11-05) and 0.828 (12-01) while perturbed accuracy holds at ≈ 0.90, pushing the gap back down to -0.070. Snapshots can *lose* previously absorbed instances.

DeepSeek: learning, then a patch. V3 $\to$ V3.2 lift all three tiers together (perturbed $0.820 \to 0.901$) with the gap pinned near $-0.02 \ldots -0.05$: a *learning* signature. V4-Pro breaks the pattern—

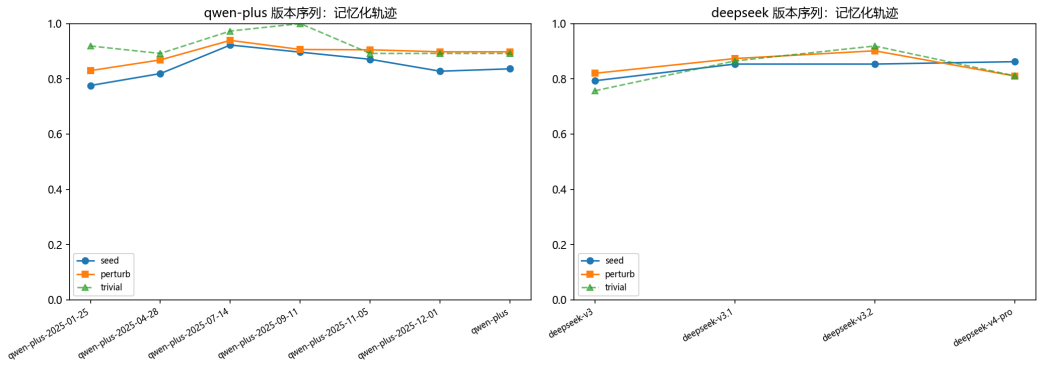

Figure 1: Seed / trivial / perturbed accuracy across the Qwen-Plus snapshot series (left) and the DeepSeek V3→V4-Pro series (right). A shrinking gap with rising perturbed accuracy indicates learning; rising seed accuracy with a persistent gap indicates patching.

seed accuracy rises to 0.862 while the perturbed tier *drops* to 0.810, flipping the gap positive ($+0.052$, the only sign flip in either series) with $\Delta_{t\to p} \approx 0$: a *patching* signature, though the family-exclusion check (Section 6.2) shows it too is concentrated in the two largest families.

DeepSeek-R1: reasoning against itself. R1's aggregate gap is small (-0.022) but hides the most extreme family-level dissociation in the study: 0.00 on the exact *strawberry* seed (both languages) vs. 0.44 on its 144 variants (family gap -0.444), alongside the lineup-worst 60.3% on sub_token_perception. Explicit re-derivation does not substitute for perception on sub-token tasks; on the canonical phrasing it appears to give the model more room to talk itself out of the correct count.

6.4 Temporal contamination: the pollution calendar

Every seed carries a dated viral period; every snapshot carries a release date that upper-bounds its training cutoff. Aligning the two yields a *natural contamination experiment*: seeds that went viral before a model's cutoff had a chance to enter its training data; seeds that went viral after did not.

Two structures in Figure 2 carry the paper's main temporal claims. **(1) The viral error itself gets absorbed (2024 era).** For every Qwen-Plus snapshot and for DeepSeek V3.x/R1 the 2024-era gap is *negative* (-0.047 to -0.147): the exact instances that went viral in 2024 are *harder* for these models than the 515 perturbed queries of the same era—the 9.9-vs-9.11 family gap alone is -0.282 for 01-25 and -0.380 for 04-28. The early-snapshot failures attach to specific viral surface forms—Chinese phrasing correct, English phrasing wrong on 9.9-vs-9.11—consistent with models having absorbed the famous *wrong* answer alongside the famous question. GLM-5.3 ($+0.132$) and Kimi-K2.5 ($+0.068$) show the opposite, memorization-like pattern on the same era. **(2) Traps generalize to un-memorizable instances (2026 era).** The three 2026-era seed families (car wash 2026-02, safe-moving 2026-03, phone repair) went viral after every dated snapshot's release; no dated snapshot could have memorized them. Yet this era carries the *largest positive gaps* in the calendar: $+0.125$ to $+0.396$ across Qwen-Plus snapshots, $+0.29$ to $+0.35$ for DeepSeek V3.x, $+0.33$ for R1—14 of 15 models positive (Kimi ≈ 0). On fresh instances the seed-vs-variant difference cannot reflect memorization; it measures the trap mechanism itself, and that mechanism is very much alive. Caveats: the 2026 era holds only 3 seed families (16 variant queries per model), and current-alias endpoints could in principle absorb 2026 items via rolling updates—which is why this claim rests on the dated snapshots. The 2025 era covers two seed families (4 queries) but has no generated variants and is omitted from gap comparisons.

6.5 The viral-language fingerprint

If memorization of a viral instance is mediated by the training data of the language community where it went viral, the gap should be *larger* when the question is asked in its viral language. Each

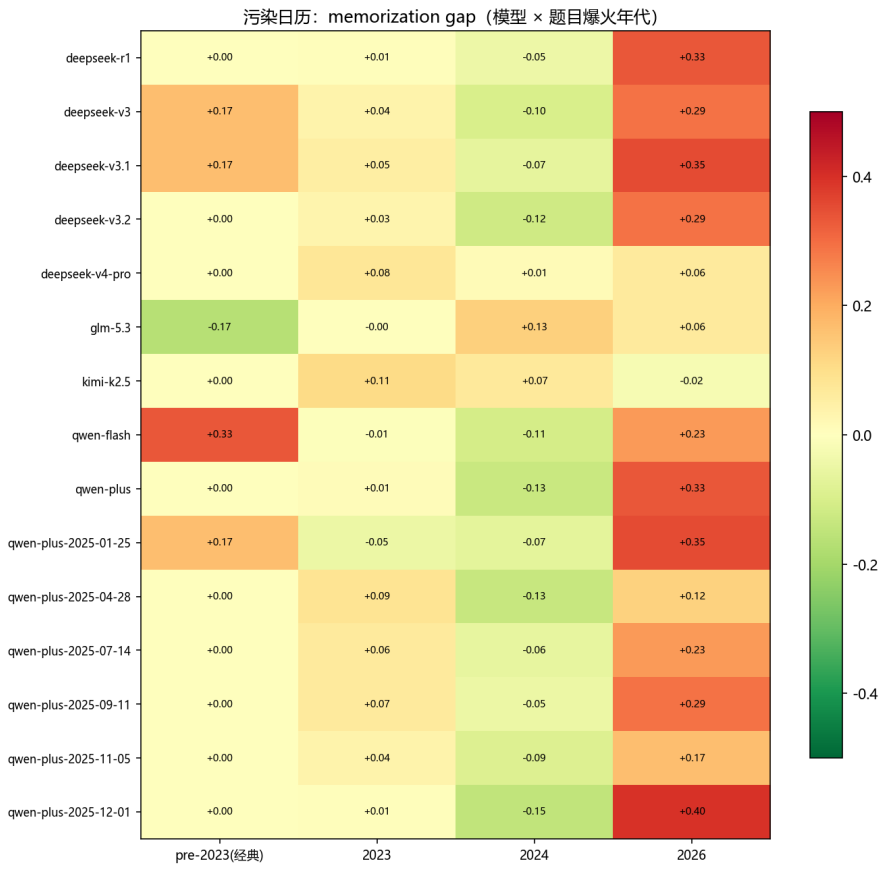

Figure 2: The memorization gap by viral era and model snapshot. The anticipated diagonal (large gaps only for eras preceding each snapshot's release) does *not* appear: 2024-era gaps are negative for every Qwen-Plus snapshot, while the post-cutoff 2026 era carries the largest positive gaps—on instances no dated snapshot could have memorized, the trap mechanism itself dominates.

seed's `viral_language` annotation (zh: 33, en: 20, both: 11) lets us test this on bilingual items: we compare the gap when query language matches the viral language against when it does not.

Result: null. Across the 15 models Δ_{fp} ranges from -0.143 (DeepSeek-V3.1) to $+0.074$ (Qwen-Plus 04-28), with no consistent sign within or across families: Qwen-Plus snapshots are mostly small-negative, the DeepSeek series mixed, Kimi $+0.056$, GLM -0.001; a hash-based mock model gives ≈ 0, confirming the estimator is unbiased under a memorization-free scorer. We find *no evidence* that instance absorption is mediated primarily by the language community in which a question went viral. The one language-linked effect we observe is item-specific rather than community-wide: the early-snapshot failure on 9.9-vs-9.11 (viral in *both* languages) attaches to its English phrasing—Qwen-Plus 01-25 and 04-28 answer the Chinese version correctly and the English one wrongly, while passing 78–88% of the family's 142 novel decimal variants. Whatever models absorb from a viral episode appears to bind to exact surface forms, not to a language community's overall gap profile.

6.6 Cross-lingual consistency

Per-item verdict consistency ranges from 0.668 (DeepSeek-V4-Pro) to 0.928 (Qwen-Plus 07-14). Pooled over all models and bilingual items, consistency is *lowest* for memorized_pattern_override (0.781) and pragmatic_presupposition_failure (0.800), and *highest* for knowledge_consistency_failure (0.907)—with sub_token_perception at 0.896, the opposite of what a tok-

model	zh acc	en acc	per-item consistency
Qwen-Plus 01-25	0.824	0.837	0.818
Qwen-Plus 04-28	0.866	0.874	0.831
Qwen-Plus 07-14	0.941	0.952	**0.928**
Qwen-Plus 09-11	0.913	0.918	0.883
Qwen-Plus 11-05	0.887	0.931	0.870
Qwen-Plus 12-01	0.889	0.900	0.859
Qwen-Plus (current)	0.883	0.909	0.866
Qwen-Flash	0.894	0.913	0.907
DeepSeek-V3	0.848	0.787	0.761
DeepSeek-V3.1	0.881	0.874	0.833
DeepSeek-V3.2	0.874	0.926	0.844
DeepSeek-V4-Pro	0.783	0.842	0.668
DeepSeek-R1	0.813	0.785	0.781
Kimi-K2.5	0.748	0.783	0.779
GLM-5.3	0.675	0.677	0.738

Table 6: zh/en accuracy and per-item verdict consistency on the 461 bilingual items.

enization story would predict. Cross-language inconsistency concentrates exactly where instance-level exposure lives: a famous phrasing absorbed (with its famous error) in one language produces language-split verdicts, whereas character-counting failures, though severe for R1, are language-independent in *which items* they hit.

7 Discussion

Is fixing viral questions real progress? The version series answer: *sometimes, and not monotonically*. The Qwen-Plus family genuinely improved between January and July 2025—all three tiers rose and the 9.9-vs-9.11 patch is datable to the April–July window (Section 6.3)—but later snapshots lost seed accuracy again while variant accuracy held, and DeepSeek-V4-Pro's positive gap is a family-concentrated patching signature rather than broad competence (Section 6.4). Meanwhile the 2026-era holdout shows the underlying trap mechanisms remain effective on instances no dated snapshot has seen. The prescription follows directly: report the seed-vs-perturb gap *per family and per era* alongside headline accuracy. A model that "fixes" a viral question without lifting its perturbed tier has moved a datum, not a capability frontier.

Why the mechanism axis matters. Grouping by mechanism (Table 4) reveals structure that surface grouping obscures—e.g., pragmatic-presupposition items scattered across "physical commonsense" and "pragmatic trap" categories behave alike. Mitigations should target mechanisms (e.g., tool-augmented character enumeration for sub_token_perception) rather than individual viral questions.

Trivial controls as difficulty calibration. The $\Delta_{t\to p}$ component isolates the trap itself from item difficulty, and it cleanly stratifies the lineup: Kimi-K2.5 and GLM-5.3 lose ≈ 0.20 between de-trapped and re-trapped readings of the *same* families—heuristic override in the sense of HOB (Li et al., 2026)—while every one of the other thirteen checkpoints stays within ± 0.094 (ten of thirteen within ± 0.05). A high trivial tier with a low perturbed tier means the model can read but not resist. The Qwen series' *negative* $\Delta_{s\to t}$ adds a third pattern that no difficulty-based account can explain: the canonical instance is harder than its own de-trapped control, which only instance-level exposure—absorbing the famous question together with the famous wrong answer—can produce.

8 Limitations

- **Model access**: all models are served through one Chinese hosting platform (Alibaba Cloud Bailian); GPT-4o/o3-mini-class Western models were not available to us under this access. The uniform endpoint removes cross-vendor serving confounds but limits vendor coverage; whether Western frontier models exhibit the same three signatures is an open question.

- **Cutoff proxy**: snapshot release dates upper-bound training cutoffs; exact cutoffs are undisclosed, so era buckets near a release date are fuzzy.
- **Judge dependence**: 88 items rely on an LLM judge. Our manual audit (Section A) covers the Chinese verdicts only (29 items, single annotator): 96.6% confirmed, every sampled PASS verdict confirmed, no leniency detected—but English judge verdicts and full double annotation remain future work.
- **Virality is a biased sampler**: ViralBench covers failures the public happened to discover, not the full space of model errors.
- **Generator artifacts**: rule-scored variants are structurally faithful but may introduce distributional artifacts (e.g., rare words) a fully manual set would avoid; the trivial-control tier partially mitigates this.
- **Seed scarcity per family**: only 64 seeds by design (each must be genuinely viral); per-family statistics on seeds alone are noisy, which is why gaps are reported pooled and with family-level robustness checks.

9 Conclusion

ViralBench turns a recurring internet phenomenon—the viral LLM trick question—into a controlled instrument. By pairing provenance- and date-annotated seeds with a three-tier variant design (seed / trivial control / perturbed), organizing everything along a surface $\times$ mechanism taxonomy, and evaluating entire model version series under one protocol, it makes "learn or memorize" observable: aggregate memorization gaps are family-concentrated and sign-mixed—early snapshots can fail the exact viral instance while passing novel perturbations, and later ones patch and regress non-monotonically—yet on post-cutoff 2026 items the gap turns positive for 14 of 15 models, showing that the trap mechanisms behind viral failures generalize to instances no model could have memorized. The memorization gap, tracked over releases and aligned against viral dates, is a more honest progress metric than accuracy on the original viral prompt.

Ethics Statement

All items are publicly circulated test questions containing no private information; sources are attributed where available. The benchmark is intended for diagnostic evaluation, not for training-time gaming. The dataset, generator, and harness will be released under CC BY 4.0 (dataset) and MIT (code).

Reproducibility Statement

The anonymized supplementary material contains everything needed to reproduce this paper: the 514-item dataset with provenance and date annotations (`data/`), the byte-reproducible rule-based generator (fixed seed 20260727, `scripts/`), the evaluation harness and scorers (`src/`), the integrity and scorer test suite (`tests/`), the model configuration (`configs/`), the raw responses of all 15 checkpoints (`results/raw/`), the judge–human audit sheet (`results/audit/`), and the analysis scripts that emit every table and figure. Concretely: `python scripts/generate_variants.py` rebuilds the dataset byte-identically; `pytest tests -q` runs 43 integrity and scorer tests; `python -m viralbench.runner` evaluates a model (resume-safe); `python -m viralbench.analysis` regenerates every table and figure in this paper as CSV/PNG.

AI Use Statement

In this work, we used a generative AI tool (Claude Code, Anthropic) for two tasks. *(i) Writing assistance*: polishing and revising manuscript text. *(ii) Code assistance*: helping implement and debug the dataset generator, evaluation harness, scorers, and analysis scripts. Beyond these, we have not used generative AI tools for other tasks with required disclosure; in particular, the benchmark

contains no AI-generated items—the 64 seeds are human-curated from publicly circulated questions and the 450 variants are produced by a deterministic rule-based generator—and no AI tool was involved in collecting the experimental results. The LLM judges used for automated scoring are a described component of the evaluation methodology itself and are audited against human judgment in Appendix A. We have reviewed all AI-assisted work: AI-assisted code was read and validated by the authors against a suite of 43 integrity and scorer tests, and all reported results were regenerated and checked by the authors from the raw evaluation logs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Yuelin Bai, Xinrun Du, Yiming Liang, Yonggang Jin, et al. COIG-CQIA: Quality is all you need for chinese instruction fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. arXiv:2403.18058.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "A is B" fail to learn "B is A". *arXiv preprint arXiv:2309.12288*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yubo Li, Lu Zhang, Tianchong Jiang, Ramayya Krishnan, and Rema Padman. The model says walk: How surface heuristics override implicit constraints in LLM reasoning. *arXiv preprint arXiv:2603.29025*, 2026.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models. *arXiv preprint arXiv:2406.02061*, 2024.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, 2016. arXiv:1508.07909.

Hugh Zhang, Jeff Da, Dean Lee, et al. A careful examination of large language model performance on grade school arithmetic. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. arXiv:2405.00332.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. arXiv:2306.05685.

A JUDGE–HUMAN AGREEMENT AUDIT

Protocol. We drew a stratified sample of 100 llm_judge verdicts (round-robin over models, seed 42) from the 15 models' raw logs—6 to 7 verdicts per model; the sample contained 87 PASS / 13 FAIL. A human annotator reviewed each sampled verdict against the question, canonical answer, item-specific rubric, and the full model response, and marked whether the judge verdict was correct. The annotator audited the 29 Chinese items; English items are left to future work. Because this produces a *meta-judgment* on the judge rather than an independent verdict on the same scale, we report confirmation rate rather than Cohen's κ.

Result. 28/29 judge verdicts (96.6%) were confirmed. All 26 sampled PASS verdicts were confirmed—we detect *no leniency bias*, which matters because llm_judge items are over-represented among variants and systematic leniency would deflate the memorization gap. Of 3 FAIL verdicts, 2 were confirmed; the single error was *over-strict*: on a pun-disambiguation item ("my monitor is broken—should I go to the eye clinic or to after-sales service?"), DeepSeek-V4-Pro explicitly identified the wordplay and gave the correct resolution for both readings, but the judge applied the rubric's "any hospital/clinic recommendation $\Rightarrow$ fail" clause mechanically. The error direction implies the judge, if anything, slightly *understates* model accuracy.

B DATASET STATISTICS

Scoring-type distribution: numeric 212, choice 135, llm_judge 88, contains 45, regex 15, not_contains 13, word_count 6. Language-specific items: 53. Variant coverage: 40/64 seed families. Trivial controls: 20 items across 18 families. Queries per model: 975.

C EXAMPLE ITEMS

Four worked families illustrating the three-tier design and the two taxonomy axes. Variant answers are computed by construction; original Chinese phrasings are shown via `\zh{}` (full bilingual text in the released dataset).

1. Car wash — `pc-001` (2026-02, viral in both languages; physical_commonsense × implicit_constraint_neglect; llm_judge). Seed: 我想洗车，我家距离洗车店只有 50 米，请问你推荐我走路去还是开车去呢？ ("I want to wash my car; the shop is only 50 m away—should I walk or drive?"). Canonical: drive—the car itself must be brought to the shop. Trivial control (`v-triv-016`): distance changed to 50 km, so driving is overdetermined and the trap cannot bite. Perturbed variants (`v-tr-001/002` and four more): 20 m / 100 m / . . . —trap preserved, surface quantity novel. The three 2026-era families (car wash, safe-moving, phone repair) drive the post-cutoff gaps of Section 6.4.

2. Hunter and birds — `pc-004` (2023, viral in zh; physical_commonsense × pragmatic_presupposition_failure; llm_judge). Seed: 树上有 10 只鸟，猎人开枪打死了 1 只，树上还剩几只鸟？ ("10 birds in a tree; a hunter shoots one dead—how many remain?"). Canonical: 0 (the gunshot scares the rest away). Trivial controls (`v-bird-001` and two more): no shot is fired—有 1 只没有受伤、自己飞走了 ("one uninjured bird flies away on its own")—canonical 9, pure arithmetic. The seed/control pair separates presupposition failure from subtraction failure.

3. Chicken trade — `or-013` (classic; overfitted_riddle × memorized_pattern_override; numeric). Seed: 我 8 元买了一只鸡，9 元卖掉，10 元又买回来，11 元再卖出去。请问我一共赚了多少钱？ (buy at 8, sell at 9, re-buy at 10, sell at 11—total profit?). Canonical: 2 yuan. Perturbed (`v-chick-002` and seven more): quantities re-derived to 10/12/15/18—canonical 5, computed programmatically. A model that recites "2" on the seed while miscomputing re-derived quantities exhibits memorized-pattern override.

4. Pen names — `kc-003` (2023, viral in zh; knowledge_consistency × knowledge_consistency_failure; llm_judge). Seed: 鲁迅为什么要暴打周树人？ ("Why did Lu Xun beat up Zhou Shuren?"). Canonical: they are the same person (Zhou Shuren is Lu Xun's real name); the question's presupposition is false. Entity-swap variants (`v-aka-001/002` and four more): 金庸为什么要暴打查良镛？ (Jin Yong = Cha Liangyong) and 茅盾为什么要暴打沈雁冰？ (Mao Dun = Shen Yanbing)—the same mechanism on different pen-name pairs, answers known by construction.